

**WYDZIAŁ
ELEKTROTECHNIKI
I INFORMATYKI
POLITECHNIKI RZESZOWSKIEJ**

mgr inż. Paweł Kuraś

Redukcja niespójności w macierzach porównań parami
i nowy algorytm generowania losowych macierzy
porównań parami o zadanym zakresie współczynników
niespójności - FAST-PCM

Praca doktorska

Promotor:

dr hab. inż. Dominik Strzałka, prof. PRz

Politechnika Rzeszowska im. Ignacego Łukasiewicza

Promotor pomocniczy:

dr Edyta Łukasik

Politechnika Lubelska

Rzeszów, 2023

Podziękowania chcę przekazać:

- *mojej żonie Dominice i synowi Piotrusiowi oraz całej mojej rodzinie za wyrozumiałość przy pisaniu i zapewnieniu odpowiedniego czasu i przestrzeni na finalizację pracy*
- *promotorom za wsparcie, mentoring i pomoc na wszelkich płaszczyznach*
- *prof. Jackowi Klusce za ostateczne sugestie i pomoc w złożeniu tej pracy*
- *wszystkim koleżankom i kolegom z Zakładu Systemów Złożonych, Centrum Transferu Technologii – w szczególności Bartkowi Kowalowi i Patrykowi Organiściakowi*
- *studentom z Koła Naukowego Machine Learning, inżynierii i analizy danych, informatyki (S i H), a także z inf. i cyberbezpieczeństwa na WSiIZ – za to że po prostu jesteście i wciąż pytaliście jak mi idzie, mimo że nie lubiałem odpowiadać na to pytanie*

Życiorys autora

mgr inż. Paweł Kuraś



Urodzony 1 lipca 1995 roku w Leżajsku, gdzie w 2014 roku ukończył Liceum Ogólnokształcące im. Bolesława Chrobrego w klasie matematyczno-fizycznej. Następnie rozpoczął studia dzienne inżynierskie na kierunku informatyka na Wydziale Elektrotechniki i Informatyki Politechniki Rzeszowskiej, które ukończył w roku 2018, a także – na tej samej uczelni – studia magisterskie, ukończone w roku 2019. Absolwent studiów Executive Master of Business Administration na Akademii

Ekonomiczno - Humanistycznej w Warszawie w 2022 roku. Zawodowo związany najpierw z branżą gier komputerowych i mobilnych, poprzez pracę dla firmy Simplicity Games (od 2016 do 2022 roku) na stanowisku projektanta gier komputerowych oraz programisty gier Unity i iOS, gdzie pracował przy projektowaniu oraz realizacji ponad 30 projektów gier o zasięgu międzynarodowym, pobranych łącznie ponad 40 milionów razy. W tej branży w tym okresie współpracował również z firmami Creativity AR, Riddle Games, gray2rgb, Rebel Twins, MobilWay, Azur Games, Superera, Pirxy Games, zaś w 2021 i 2022 roku współpracował również ze startupem Playstrict, platformą wzrostu dla gier (playstrict.com), gdzie zajmował się pozyskiwaniem deweloperów gier do wczesnego dostępu do narzędzia Playstrict, rozwijaniem sieci afiliacyjnej do UA, opracowywaniem strategii marketingowej dla narzędzia Playstrict oraz reprezentowaniem firmy w kontaktach biznesowych. Od 2021 roku pracuje na stanowisku asystenta naukowo-dydaktycznego w Zakładzie Systemów Złożonych Politechniki Rzeszowskiej, a także współpracuje dydaktycznie jako gościnnie wykładowca na Wyższej Szkole Informatyki i Zarządzania w Rzeszowie, Wyższej Szkole Bankowej w Warszawie oraz Wrocławskiej Wyższej Szkole Informatyki Stosowanej, gdzie od strony dydaktycznej prowadzi laboratoria, ćwiczenia oraz wykłady i szkolenia z zakresu baz danych, sieci komputerowych, uczenia maszynowego, projektowania oprogramowania oraz cyberbezpieczeństwa i innych zagadnień informatycznych, a także z dziedziny zarządzania projektami. Podczas pracy na Politechnice Rzeszowskiej, uczestnik wielu projektów badawczych w tym trwających wciąż Politechnicznej Sieci Via Carpatia, CriNet

- systemu bezpieczeństwa sieci SDN dla infrastruktury krytycznej (w konsorcjum z firmami Exatel, Gaz-System pod nadzorem Narodowego Centrum Badań i Rozwoju), a także w zakończonych już projektach wykorzystujących metody uczenia maszynowego w modelowaniu i analizie wpływu sztywności nóg i stawów na urazy związane z aktywnością fizyczną (we współpracy z Podkarpackim Centrum Innowacji), badającym metody uczenia maszynowego w prognozowaniu anomalii w metrykach oraz korelacji krytycznych zdarzeń w maszynach mainframe (we współpracy z firmą Z-RAYS) oraz realizacji systemu zbierania i analizy danych wspierającym ocenę skuteczności zabiegów kosmetycznych w terapiach spa i uzdrowiskowych. Ostatni z wymienionych projektów poskutkowało założeniem w 2023 roku pierwszej spółki spin-off Politechniki Rzeszowskiej – Suntrail sp. z o.o. – której został prezesem zarządu oraz menadżerem projektu. Lider projektu „Design Factory: Rzeszów” na Politechnice Rzeszowskiej, menadżer projektu MoVA (laureata Carpathian Startup Fest 2022) oraz opiekun Koła Naukowego Machine Learning Politechniki Rzeszowskiej. Audytor wiodący zarządzania bezpieczeństwem informacji zg. z normą ISO 27001, ekspert Urzędu Marszałkowskiego Województwa Podkarpackiego ds. oceny projektów w ramach regionalnego programu "Fundusze Europejskie dla Podkarpacia 2021-2027" w Departamencie Realizacji Projektów Infrastrukturalnych Regionalnego Programu Operacyjnego, w sekcji wyboru projektów z zakresu mobilności, łączności i cyfryzacji, członek Polskiego Towarzystwa Informatycznego oraz Google for Startups. Posiadacz licznych certyfikatów uznanych firm takich jak NVIDIA, Microsoft, TÜV Nord, IBM oraz Niebezpiecznik.pl, a także prelegent i moderator na wielu wydarzeniach (m.in. targi pracy Network Power w G2A Arena Rzeszów, konferencje Via Carpatia oraz „Wschodnie innowacje” w Lublinie, czy PUKN na Politechnice Gdańskiej). W czasie wolnym, kompozytor muzyczny w ramach własnej działalności gospodarczej.

Spis treści

Życiorys autora	3
Spis treści	5
Wykaz skrótów używanych w pracy doktorskiej	11
Wprowadzenie do pracy doktorskiej	20
1. Przegląd literatury – macierze porównań parami, ich zastosowania, właściwości oraz problem redukcji niespójności	22
1.1. Wprowadzenie.....	20
1.2. Porównania parami i macierze porównań parami	25
1.2.1. Spójność macierzy porównań parami	27
1.2.2. Metody ustalania priorytetów w macierzach porównań parami.....	27
1.2.2.1. Metoda średniej geometrycznej	28
1.2.2.2. Metoda wartości własnej (wektora własnego).....	28
1.2.2.3. Metoda średniej arytmetycznej.....	29
1.2.2.4. Metoda najmniejszych kwadratów	29
1.2.2.5. Metoda Siraja i in.....	30
1.2.2.6. Inne interesujące podejścia	31
1.2.3. Skale dla macierzy porównań parami	32
1.3. Wielokryterialne metody podejmowania decyzji (MCDM).....	33
1.3.1. Analityczny proces hierarchiczny (AHP).....	33
1.3.1.1. Kroki stosowania metody AHP	34
1.3.1.2. Podejścia do grupowego podejmowania decyzji	35
1.3.1.3. Analityczny proces sieciowy	36
1.3.2. Best-Worst Method.....	36
1.3.2.1. Kroki stosowania BMW	36
1.3.3. Inne metody MCDM wykorzystujące macierze porównań parami	37
1.3.3.1. VIKOR.....	37
1.3.3.2. PROMETHEE	38
1.3.3.3. ELECTRE.....	38
1.4. Problem niespójności w macierzach porównań parami.....	39
1.4.1. Inne indeksy niespójności.....	39
1.4.1.1. Indeks Koczkodaja.....	39
1.4.1.2. Golden-Wang Index.....	40
1.4.1.3. Palaez-Lamata Index.....	40
1.4.1.4. (Triads) Geometric Consistency Index	41
1.4.1.5. Wskaźnik błędu względnego (REI)	41
1.4.1.6. Wskaźnik spójności harmonicznej (HCI)	41
1.5. Algorytmy redukcji niespójności.....	42

1.5.1 Algorytm Xu i Wei	42
1.5.2 Algorytm Szybowskiego.....	43
1.5.3 Algorytm Cao.....	43
1.6. Wnioski	44
2. Numeryczne porównanie algorytmów iteracyjnych redukcji niespójności w porównaniach parami	25
2.1. Wstęp	46
2.2. Miary zachowania preferencji macierzy porównań parami	47
2.3. Iteracyjne algorytmy redukcji niespójności	49
2.4. Symulacje Monte Carlo	49
2.5. Dyskusja na temat wyników	51
2.6. Narzędzie PCM_CR.....	57
2.7. Wnioski	57
3. REDUCE.py: moduł Pythona do redukcji niespójności w macierzach porównań parami.....	59
3.1. Wstęp.....	60
3.1.1. Uzasadnienie wyboru algorytmów do konstrukcji modułu	60
3.2. Architektura i struktura oprogramowania	61
3.2.1. Repozytorium w serwisie GitHub.....	62
3.2.2. Konteneryzacja biblioteki Reduce.py za pomocą narzędzia Docker	63
3.2.3. Licencja MIT dla projektu Reduce	63
3.3. Funkcjonalności modułu	64
3.3.1. Importowanie macierzy porównań parami z pliku CSV - import_pc_matrix_from_csv()	64
3.3.2. Generowanie losowej macierzy porównań parami - create_pc_matrix()	65
3.3.3. Funkcje pomocnicze	66
3.3.4. Funkcje implementujące algorytmy redukcji niespójności.....	66
3.3.4.1. Funkcja implementująca algorytm Xu & Wei	67
3.3.4.2. Funkcja implementująca algorytm Cao.....	67
3.3.4.3. Funkcja implementująca algorytm Szybowskiego.....	68
3.3.5. Funkcje implementujące indeksy spójności macierzy porównań parami	68
3.3.5.1. Funkcja implementująca Koczkodaj Index (KI)	69
3.3.5.2. Funkcja implementująca Golden-Wang Index (GWI)	69
3.3.5.3. Funkcja implementująca Palaez-Lamata Index (PLI).....	70
3.3.5.4. Funkcja implementująca Geometric Consistency Index (GCI)	70
3.3.5.5. Funkcja implementująca Triads Geometric Consistency Index (TGCI)....	71
3.3.5.6. Funkcja implementująca Relative Error Index (REI).....	71
3.3.5.7. Funkcja implementująca Harmonic Consistency Index (HCI)	72
3.4. Testy jednostkowe.....	72

3.4.1 Testy pomocnicze oraz testy indeksów – test_auxillary_and_indexes.py	73
3.4.2 Test algorytmu Cao – test_cao.py	73
3.4.3 Test algorytmu Szybowskiego – test_szybowski.py	74
3.4.4 Test algorytmu Xu and Wei – test_xu_and_wei.py	74
3.4.5 Proces testowania	74
3.5. Złożoność obliczeniowa funkcji modułu	75
3.5.1. Analiza zbiorcza złożoności obliczeniowej modułu Reduce.py	75
3.5.2. Złożoność obliczeniowa funkcji pomocniczych	77
3.5.3. Złożoność obliczeniowa funkcji redukcji niespójności	78
3.5.4. Złożoność obliczeniowa funkcji z indeksami spójności	79
3.5.5. Potencjał poprawy złożoności obliczeniowej do wdrożenia w przyszłych wersjach modułu	80
3.6. Testowanie wydajności algorytmów redukcji niespójności zawartych w bibliotece	81
3.6.1. Testy czasu wykonania	82
3.6.2. Testy zużycia pamięci	84
3.6.3. Skalowalność algorytmów	86
3.6.4. Zbieżność algorytmów	86
3.7. Przykład ilustrujący zastosowanie modułu	87
3.8. Wpływ modułu REDUCE na rozwój dziedzinowy	88
3.8.1. Potencjalny wpływ na rozwój małych i średnich przedsiębiorstw, w badaniach i zastosowania przemysłowe modułu	89
3.9. Wnioski i kierunek dalszego rozwoju	89
4. REDUCE (reduce.prz.edu.pl): internetowe narzędzie wspomagania decyzji oraz redukcji niespójności w multiplikatywnych porównaniach parami	90
4.1. Wstęp	91
4.1.1 Ogólna struktura i działanie aplikacji	91
4.1.2 Uzasadnienie wybranych technologii	95
4.2. Analiza budowy aplikacji	99
4.2.1. Warstwa front-end	99
4.2.1.1 Struktura sekcji body kodu HTML	99
4.2.1.2 Atrybuty stylu graficznych CSS	103
4.2.1.3 Skrypty JavaScript oraz jQuery	105
4.2.1.3.1 Funkcja setMatrixVisibility()	105
4.2.1.3.2 Funkcja setMatrixSize()	106
4.2.1.3.3 Funkcja getParam()	107
4.2.1.3.4 Funkcja setMatrixValue()	108
4.2.1.3.5 Funkcja generateRandomMatrix()	108
4.2.1.3.6 Funkcja resetMatrix()	109
4.2.1.3.7 Funkcja setReduceButton()	109

4.2.1.3.8	Funkcja getCICR()	110
4.2.1.3.9	Funkcja sendMatrix()	111
4.2.1.3.10	Funkcja setNewMatrix	111
4.2.1.3.11	Funkcja checkMatrix()	112
4.2.1.3.12.	Funkcja checkMatrixItem().....	113
4.2.1.3.13.	Funkcja \$(document).ready().....	114
4.2.2.	Warstwa back-end.....	115
4.2.2.1.	Funkcja computeCI_CR()	115
4.2.2.2.	Funkcja reduceButton().....	116
4.2.3.	Rozwiązania serwerowe i administracyjne	117
4.2.3.1	Konfiguracja Gunicorn oraz NGINX.....	118
4.2.3.2	Konfiguracja narzędzia Certbot (Let's Encrypt).....	119
4.2.3.3	Konfiguracja firewalli i zabezpieczeń.....	119
4.2.3.4	Konfiguracja dostępu administracyjnego.....	121
4.3.	Testy zgodności, bezpieczeństwa oraz wydajnościowe aplikacji	122
4.3.1.	Testy zgodności na różnych urządzeniach i platformach.....	122
4.3.2.	Testy bezpieczeństwa i firewalla	129
4.3.2.1	Skanowanie portów z użyciem nmap.....	130
4.3.2.2	Symulacja ataku XSS i reakcji systemu Snort	130
4.3.2.3	Symulacja ataku SQL Injection i reakcji systemu Snort.....	131
4.3.2.4	Symulacja ataku DDoS	132
4.3.3	Testy TLS/SSL.....	133
4.3.4	Testy wydajności ładowania aplikacji	143
4.4.	Walidacja poprawności budowy aplikacji w zg. z obowiązującymi standardami ..	148
4.4.1	Walidacja zgodności ze standardem W3C HTML	149
4.4.2.	Walidacja zgodności ze standardem W3C CSS.....	152
4.4.3.	Audyt dostępności wg. standardu WCAG 2.1	153
4.5.	Ścieżki dalszego rozwoju oprogramowania i jego komercjalizacji.....	160
4.5.1	Identyfikacja potrzeby rynkowej	161
4.5.2	Opis rynku docelowego	162
4.5.3	Opis przygotowanej ścieżki komercjalizacji.....	164
4.6.	Wnioski końcowe.....	166
5.	Zbadanie potencjału użycia wybranych technik uczenia maszynowego do redukcji niespójności macierzy porównań parami	167
5.1.	Wstęp	168
5.1.1.	Uzasadnienie wyboru algorytmu będącego podstawą badań.....	168
5.1.2.	Miary zachowania preferencji jako ostateczna metoda ewaluacji skuteczności nowego modelu	170

5.1.3. Zasadność użycia technik machine learning do zagadnienia redukcji niespójności w macierzach porównania parami	171
5.2. ML w kontekście redukcji niespójności macierzy porównań parami.....	172
5.2.1. Podstawowe rodzaje uczenia maszynowego	173
5.2.1.1. Uczenie nadzorowane	173
5.2.1.2. Uczenie nienadzorowane	174
5.2.1.3. Uczenie przez wzmacnianie.....	175
5.2.1.4. Wybór odpowiedniej grupy algorytmów do badań	176
5.2.2. Wybrane do badań techniki regresji	181
5.2.3. Wskaźniki dot. skuteczności wybranych technik regresji	183
5.3. Metodyka badań.....	186
5.3.1. Zbieranie i przetwarzanie danych	187
5.3.2. Zastosowanie technik uczenia maszynowego.....	188
5.3.2.1. Użyte biblioteki.....	189
5.3.2.2. Analiza kodu	189
5.3.3. Kryteria oceny.....	193
5.4. Wyniki i ich omówienie.....	194
5.4.1. Wartości MSE, MAE oraz R^2	194
5.4.2. Zgodność obliczeń z prawdziwą wartością CR oraz dalsze wskaźniki	196
5.4.3. Porównanie proponowanego podejścia z istniejącymi metodami redukcji niespójności w macierzach porównań parami	197
5.4.4. Dyskusja na temat wyników	198
5.5. Wnioski.....	198
6. Nowy algorytm generowania losowych macierzy porównań parami o zadanym zakresie współczynników niespójności - FAST-PCM oraz narzędzie online PC MATRICES GENERATOR do wydajnego generowania macierzy porównań parami	200
6.1. Wstęp	201
6.2. Opis problemu i opracowanego algorytmu.....	201
6.2.1. Standardowa metoda generowania macierzy porównań parami z żądanym współczynnikiem spójności	202
6.2.2. Ulepszona metoda generowania macierzy porównań parami z żądanym współczynnikiem spójności	202
6.3. Wyniki i ich omówienie.....	204
6.3.1. Metodyka badań.....	204
6.3.2. Wyniki dla metody standardowej losowego generowania macierzy pp.....	206
6.3.3. Wyniki dla ulepszonej metody losowego gen. macierzy pp.....	206
6.4. "PC MATRICES GENERATOR" – impl. proponowanej nowej metody.....	208
6.5. Wnioski.....	209
Podsumowanie rozprawy doktorskiej.....	210

Bibliografia	212
Spis rysunków	212
Spis listingów	258
Spis tabel	260
Streszczenie	261
Abstract	262
Appendix A	262

Wykaz skrótów używanych w pracy doktorskiej

- **0-RTT** (*Zero Round Trip Time, pol. czas bez ponownego nawiązywania połączenia*)
- mechanizm umożliwiający szybsze nawiązywanie bezpiecznych połączeń w internecie, pomijając niektóre etapy ręki podczas wznawiania sesji.
- **ACL** (*Access Control List, pol. lista kontroli dostępu*) - lista definiująca uprawnienia dostępu dla użytkowników lub grup do określonych zasobów w systemie komputerowym.
- **AHP** (*Analytic Hierarchy Process, pol. analityczny proces hierarchiczny*) - metoda wspomaganie decyzji, która organizuje i analizuje złożone decyzje, opierając się na matematyce i psychologii.
- **AIA** (*Authority Information Access, pol. dostęp do informacji autorytatywnych*)
- część certyfikatu cyfrowego, która zawiera informacje o tym, jak uzyskać informacje o certyfikacji.
- **AIR** (*Algorithms of Inconsistency Reduction, pol. algorytmy redukcji niespójności*)
- metody lub procedury matematyczne stosowane do poprawy spójności w systemach decyzyjnych.
- **AJAX** (*Asynchronous JavaScript and XML, pol. asynchroniczny JavaScript i XML*) - technika tworzenia interaktywnych aplikacji internetowych, która pozwala na aktualizację strony bez konieczności jej przeładowywania.
- **AMM** (*Arithmetic Mean Method, pol. metoda średniej arytmetycznej*)
- znana również jako metoda znormalizowanych wartości średnich.
- **ANOVA** (*Analysis of Variance, pol. analiza wariancji*) - zbiór modeli statystycznych oraz ich procedur analitycznych, które są używane do analizy różnic między średnimi grupowymi.
- **ANP** (*Analytic Network Process, pol. analityczny proces sieciowy*) - metoda oceny wielokryterialnej, która umożliwia uwzględnienie zależności i sprzężeń zwrotnych między elementami decyzji.
- **ARIMA** (*AutoRegressive Integrated Moving Average, pol. autoregresyjny model ze średnią ruchomą*) - rodzaj modelu statystycznego używanego do analizy i prognozowania danych czasowych.
- **ART** (*Adaptive Resonance Theory, pol. teoria rezonansu adaptacyjnego*) - teoria i metoda uczenia się sieci neuronowych w czasie rzeczywistym.

- **BSD** (*Berkeley Software Distribution, pol. dystrybucja oprogramowania Berkeley*) - rodzina systemów operacyjnych UNIX, które zostały rozwinięte i rozpowszechnione przez Uniwersytet Kalifornijski w Berkeley.
- **BWM** (*Best Worst Method, pol. metoda najlepszego i najgorszego*) - metoda stosowana do określenia wagi kryteriów w wielokryterialnej analizie decyzyjnej.
- **CI** (*Consistency Index, pol. indeks spójności*) - miara używana w metodzie AHP do oceny, jak bardzo oceny w macierzy porównań parami są spójne.
- **CLI** (*Command Line Interface, pol. interfejs wiersza poleceń*) - interfejs użytkownika, który pozwala użytkownikom wprowadzać polecenia tekstowe w celu wykonania określonych zadań.
- **CLS** (*Cumulative Layout Shift, pol. kumulatywne przesunięcie układu*) - metryka używana do oceny stabilności wizualnej strony internetowej, mierzy częstotliwość i znaczenie niespodziewanych zmian układu strony.
- **CR** (*Consistency Ratio, pol. współczynnik spójności*) - stosunek między indeksem spójności a odpowiednim indeksem przypadkowym, używany w AHP.
- **CSM** (*Chi-Square Method, pol. metoda chi-kwadrat*) - metoda lub procedura wyboru najbardziej spójnych wyników z zestawu danych.
- **CSS** (*Cascading Style Sheets, pol. kaskadowe arkusze stylów*) - język służący do opisu prezentacji strony internetowej napisanej w HTML lub XML.
- **CSV** (*Comma-Separated Values, pol. wartości oddzielone przecinkami*) - format pliku służący do przechowywania tabelarycznych danych, gdzie każda linia pliku jest rekordem danych, a każdy rekord składa się z pól oddzielonych przecinkami.
- **DDoS** (*Distributed Denial of Service, pol. rozproszona odmowa usługi*) - rodzaj ataku na system komputerowy, który polega na przeciążeniu zasobów serwera, co uniemożliwia obsługę prawidłowych żądań użytkowników.
- **DNS CAA** (*Domain Name System Certificate Authority Authorization, pol. autoryzacja organu certyfikującego w systemie nazw domen*) - rekord w DNS, który pozwala właścicielom domen określić, które jednostki certyfikujące mogą wydawać certyfikaty dla ich domen.
- **DQN** (*Deep Q-Network, pol. głęboka sieć Q*) - typ sztucznej sieci neuronowej stosowany w uczeniu ze wzmocnieniem.
- **DoS** (*Denial of Service, pol. odmowa usługi*) - atak na system komputerowy lub sieć, który uniemożliwia użytkownikom korzystanie z usług.

- **ELECTRE** (*ELimination Et Choix Traduisant la REalité*, pol. eliminacja i wybór odwzorowujący rzeczywistość) - rodzina metod w teorii decyzji wielokryterialnej.
- **EV** (*Eigenvalue*, pol. wartość własna) - liczba charakteryzująca przekształcenia liniowe w algebrze liniowej.
- **FCP** (*First Contentful Paint*, pol. pierwsze rysowanie z treścią) - metryka wydajności strony internetowej mierząca czas od rozpoczęcia ładowania do momentu, gdy pojawi się pierwszy element tekstu lub obrazu.
- **GAIA** (*Geometric Analysis for Interactive Aid*, pol. geometria analizy dla interaktywnej pomocy) - technika analizy danych wielowymiarowych wykorzystująca geometrię do wsparcia podejmowania decyzji.
- **GAN** (*Generative Adversarial Network*, pol. generatywne sieci antagonistyczne) - klasa algorytmów uczenia maszynowego, w której dwa modele sieci neuronowych rywalizują ze sobą w celu generowania nowych, syntetycznych instancji danych.
- **GCI** (*Geometry Consistency Index*, pol. geometria indeksu spójności) - miara spójności w metodach decyzyjnych, szczególnie w kontekście porównań parami.
- **GDSS** (*Group Decision Support System*, pol. system wspomagania decyzji grupowych) - klasy systemów informacyjnych wspomagających procesy podejmowania decyzji w grupach.
- **GEM** (*Generalized Eigenvalue Method*, pol. uogólniona metoda wartości własnej) - metoda matematyczna stosowana w różnych dziedzinach, w tym w analizie decyzji.
- **GLSM** (*Generalized Least Squares Method*, pol. uogólniona metoda najmniejszych kwadratów) - metoda statystyczna służąca do estymacji nieznanymi parametrów w modelach regresji.
- **GMM** (*Geometric Mean Method*, pol. metoda średniej geometrycznej) - metoda statystyczna wykorzystywana do oszacowania parametrów modeli, w której do wyznaczenia centralnej tendencji zestawu liczb używa się średniej geometrycznej.
- **GPL** (*General Public License*, pol. ogólna licencja publiczna) - szeroko stosowana darmowa licencja oprogramowania, która gwarantuje użytkownikom końcowym wolność korzystania, studiowania, dzielenia się i modyfikowania oprogramowania.
- **GPM** (*Geometric Priority Method*, pol. geometryczna metoda priorytetowa) - metoda stosowana do określania priorytetów w analizie decyzyjnej.
- **GWI** (*Golden-Wang Index*, pol. indeks Golden-Wang) - wskaźnik używany do pomiaru spójności w procesie decyzyjnym.

- **HCI** (*Harmonic Consistency Index*, pol. *harmoniczny indeks spójności*) - wskaźnik używany w metodach decyzyjnych do mierzenia spójności porównań parami.
- **HM** (*Harmonic Model*, pol. *model harmoniczny*) - model stosowany do w systemie lub procesie decyzyjnym.
- **HPKP** (*HTTP Public Key Pinning*, pol. *przywiązanie klucza publicznego HTTP*) - mechanizm bezpieczeństwa, który pozwala witrynom internetowym na wskazanie kluczy publicznych, które powinny być akceptowane przez przeglądarki.
- **HSTS** (*HTTP Strict Transport Security*, pol. *ściśle zabezpieczenie transportu HTTP*) - polityka bezpieczeństwa, która zmusza przeglądarki do korzystania z połączeń HTTPS.
- **HTML** (*HyperText Markup Language*, pol. *język znaczników hipertekstowych*) - standardowy język znaczników używany do tworzenia stron internetowych.
- **HTTP(S)** (*HyperText Transfer Protocol (Secure)*, pol. *protokół przesyłania hipertekstu (bezpieczny)*) - protokół komunikacyjny używany w Internecie, wersja zabezpieczona (HTTPS) używa szyfrowania.
- **ICMP** (*Internet Control Message Protocol*, pol. *internetowy protokół kontroli komunikatów*) - protokół wykorzystywany do przekazywania informacji o problemach z komunikacją w sieciach IP.
- **IMAP** (*Internet Message Access Protocol*, pol. *protokół dostępu do wiadomości w Internecie*) - protokół umożliwiający dostęp do wiadomości e-mail przechowywanych na serwerze.
- **INSITE** (*Interactive Sensitivity Investigation, Testing, and Evaluation*, pol. *interaktywne testowanie, badanie i ocena wrażliwości*) - zestaw narzędzi lub metodyka przeznaczona do oceny wrażliwości systemu lub modelu.
- **IP** (*Internet Protocol*, pol. *protokół internetowy*) - główny protokół komunikacyjny używany do przesyłania danych przez sieci komputerowe.
- **JS** (*JavaScript*) - skryptowy język programowania używany do tworzenia interaktywnych efektów w przeglądarkach internetowych.
- **KI** (*Koczkodaj Index*, pol. *indeks Koczkodaja*) - miara niespójności w porównaniach parami w metodach decyzyjnych, wprowadzona przez Waldemara Koczkodaja.
- **KNN** (*K-Nearest Neighbors*, pol. *k-najbliższych sąsiadów*) - algorytm używany w statystyce do klasyfikacji i regresji.

- **LCP** (*Largest Contentful Paint*, pol. *największe malowanie z treścią*) - metryka wydajności strony internetowej, która mierzy czas od rozpoczęcia ładowania strony do momentu, gdy największy element treści został wyrenderowany na ekranie.
- **LDM** (*Least Distance Method*, pol. *metoda najmniejszej odległości*) – jedna z metod priorytetyzacji.
- **LOIC** (*Low Orbit Ion Cannon*) - oprogramowanie używane do przeprowadzania prostych ataków typu DoS lub DDoS.
- **LP** (*Linear Programming*, pol. *programowanie liniowe*) - metoda matematyczna służąca do znalezienia najlepszego wyniku (na przykład maksymalizacji zysku lub minimalizacji kosztów) w modelu, którego wymagania są reprezentowane przez liniowe relacje.
- **LSM** (*Least Squares Method*, pol. *metoda najmniejszych kwadratów*) - metoda w statystyce służąca do znajdowania najlepszego dopasowania krzywej do zestawu punktów danych poprzez minimalizację sumy kwadratów odchyleń.
- **MAE** (*Mean Absolute Error*, pol. *błąd średni absolutny*) - metryka używana w statystyce do mierzenia różnicy między wartościami przewidywanymi a rzeczywistymi.
- **MARSAN** (*Multi-Attribute Range Sensitivity Analysis*, pol. *wieloatrybutowa analiza wrażliwości zakresu*) - technika analizy wrażliwości w badaniach operacyjnych i podejmowaniu decyzji wielokryterialnych.
- **MCDM** (*Multi-Criteria Decision Making*, pol. *wielokryterialne podejmowanie decyzji*) - dyscyplina zajmująca się podejmowaniem decyzji z uwzględnieniem wielu kryteriów.
- **MIT** (**Massachusetts Institute of Technology**) - prywatna uczelnia badawcza w Cambridge, Massachusetts
- **ML** (*Machine Learning*, pol. *uczenie maszynowe*) - dział sztucznej inteligencji, który zajmuje się tworzeniem systemów zdolnych do uczenia się i dokonywania wniosków na podstawie danych.
- **MLP** (*Multi-Layer Perceptron*, pol. *wielowarstwowy perceptron*) - typ sieci neuronowej używany w uczeniu maszynowym.
- **MLR** (*Multiple Linear Regression*, pol. *wielokrotna regresja liniowa*) - technika statystyczna służąca do modelowania relacji między jedną zmienną zależną a dwiema lub więcej zmiennymi niezależnymi.

- **MSE** (*Mean Squared Error, pol. średni błąd kwadratowy*) - metryka używana do mierzenia jakości estymatora; jest to średnia wartość kwadratów błędów, czyli różnic między estymowanymi a rzeczywistymi wartościami.
- **MSRPC** (*Microsoft Remote Procedure Call*) - protokół sieciowy opracowany przez Microsoft, który umożliwia komunikację między procesami.
- **MŚP** (*Male i średnie przedsiębiorstwa*) - klasyfikacja przedsiębiorstw w zależności od ich wielkości, która obejmuje liczby pracowników, obroty czy bilans ogólny.
- **NAT** (*Network Address Translation, pol. translacja adresów sieciowych*) - metoda mapowania adresów IP stosowana w sieciach komputerowych, która umożliwia jeden adres IP dostawcy usług internetowych (ISP) na wiele komputerów w sieci lokalnej.
- **NJR** (*Normalized Judgment Ratio, pol. znormalizowany stosunek oceny*) - metryka używana w analizie decyzji, szczególnie w kontekście metod porównań parami.
- **NJV** (*Normalized Judgment Vector, pol. znormalizowany wektor oceny*) - wektor stosowany w metodach decyzyjnych, który reprezentuje znormalizowane oceny w procesie porównań parami.
- **OCSP** (*Online Certificate Status Protocol, pol. protokół statusu certyfikatu online*) - protokół internetowy używany do uzyskania statusu certyfikatu cyfrowego.
- **PAPRIKA** (*Potentially All Pairwise RanKings of all possible Alternatives, pol. metoda potencjalnie wszystkich rankingów par wszystkich możliwych alternatyw*) - metoda używana w podejmowaniu decyzji wielokryterialnych do generowania rankingów alternatyw.
- **PC** (*Pairwise Comparison, pol. porównanie parami*) - metoda oceny, w której elementy są porównywane parami w celu ustalenia ich priorytetów.
- **PCM** (*Pairwise Comparison Matrix, pol. macierz porównań parami*) - macierz używana w metodach decyzyjnych do reprezentowania i analizy porównań parami.
- **PLI** (*Palaez-Lamata Index, pol. indeks Palaez-Lamata*) - wskaźnik używany do mierzenia spójności w metodach podejmowania decyzji opartych na porównaniach parami.
- **POP3** (*Post Office Protocol 3, pol. protokół pocztowy 3*) - standardowy protokół komunikacyjny używany przez klientów poczty e-mail do odbierania wiadomości z serwera poczty elektronicznej.

- **PPO** (*Proximal Policy Optimization*, pol. *optymalizacja polityki proksymalnej*) - algorytm uczenia ze wzmocnieniem stosowany w sztucznej inteligencji do optymalizacji strategii decyzyjnych.
- **PROMETHEE** (*Preference Ranking Organization METHod for Enrichment Evaluations*, pol. *metoda organizacji rankingu preferencji dla oceny wzbogacenia*) - metoda podejmowania decyzji wielokryterialnych, która pomaga w ustaleniu preferencji między różnymi opcjami.
- **REI** (*Relative Error Index*, pol. *indeks błędu względnego*) - wskaźnik używany do mierzenia błędu względnego w analizie decyzji.
- **RI** (*Random Index*, pol. *indeks losowy*) - wartość używana w metodzie AHP do oceny spójności porównań parami.
- **RL** (*Reinforcement Learning*, pol. *uczenie ze wzmocnieniem*) - obszar sztucznej inteligencji skoncentrowany na tym, jak agenci powinni podejmować działania w środowisku, aby maksymalizować jakąś koncepcję nagrody kumulatywnej.
- **RSA** (*Rivest–Shamir–Adleman*) - jeden z pierwszych publicznych algorytmów kryptograficznych z kluczem publicznym, szeroko stosowany do bezpiecznej wymiany danych.
- **SHA256** (*Secure Hash Algorithm 256-bit*, pol. *256-bitowy bezpieczny algorytm skrótu*) - jedna z funkcji skrótu używana w kryptografii, która generuje unikalny, stałodługościowy skrót dla danych.
- **SMTP** (*Simple Mail Transfer Protocol*, pol. *prosty protokół przesyłania poczty*) - internetowy standard komunikacyjny używany do przesyłania wiadomości e-mail między serwerami.
- **SOM** (*Self-Organizing Map*, pol. *samoorganizująca się mapa*) - typ sieci neuronowych stosowany do wizualizacji i analizy danych wielowymiarowych.
- **SQL** (*Structured Query Language*, pol. *strukturalny język zapytań*) - język programowania używany do tworzenia, modyfikowania, zarządzania i pobierania danych z relacyjnych baz danych.
- **SSH** (*Secure Shell*, pol. *bezpieczna powłoka*) - protokół sieciowy umożliwiający bezpieczną wymianę danych między komputerami w sieci.
- **STJD** (*Standard Total Judgment Deviation*, pol. *standardowa całkowite odchylenie oceny*) - termin stosowany w niektórych metodach podejmowania decyzji dla określenia odchylenia między ocenami a standardem.

- **SVM** (*Support Vector Machine*, pol. *maszyna wektorów nośnych*) - nadzorowany model uczenia maszynowego stosowany w analizie danych do klasyfikacji i regresji.
- **TBT** (*Total Blocking Time*, pol. *całkowity czas blokowania*) - metryka wydajności stron internetowych, która mierzy całkowity czas, w którym strona jest zablokowana przed interakcją użytkownika.
- **TGCI** (*Triads Geometric Consistency Index*, pol. *geometryczny indeks spójności triad*) - metoda oceny spójności w procesach podejmowania decyzji, gdzie wykorzystuje się porównania parami.
- **TJD** (*Total Judgment Deviation*, pol. *całkowite odchylenie oceny*) - metryka używana do mierzenia odchylenia w ocenach w metodach podejmowania decyzji.
- **TLS/SSL** (*Transport Layer Security/Secure Sockets Layer*, pol. *bezpieczeństwo warstwy transportowej/protokół warstwy gniazd bezpiecznych*) - protokoły kryptograficzne zapewniające bezpieczeństwo komunikacji w sieci Internet.
- **TND** (*Total Normalized Deviation*, pol. *całkowite znormalizowane odchylenie*) - termin używany w analizie decyzji dla mierzenia znormalizowanego odstępstwa od spójności.
- **TRPO** (*Trust Region Policy Optimization*, pol. *optymalizacja polityki regionu zaufania*) - algorytm uczenia ze wzmocnieniem, który ogranicza krok aktualizacji polityki, aby zapewnić monotonne poprawy.
- **TTFB** (*Time to First Byte*, pol. *czas do pierwszego bajta*) - czas, jaki upływa od złożenia przez użytkownika żądania do serwera do momentu otrzymania pierwszego bajta odpowiedzi.
- **TTI** (*Time to Interactive*, pol. *czas do interaktywności*) - metryka wydajności strony internetowej, określająca jak długo trwa, zanim strona stanie się w pełni interaktywna dla użytkownika.
- **UFW** (*Uncomplicated Firewall*, pol. *nieskomplikowany firewall*) - narzędzie konfiguracyjne firewalla przeznaczone dla systemów operacyjnych Linux, mające na celu ułatwienie konfiguracji iptables.
- **VIKOR** (*VlseKriterijumska Optimizacija i Kompromisno Resenje*, pol. *wielokryterialna optymalizacja i rozstrzygnięcie kompromisowe*) - metoda pomocna w podejmowaniu decyzji wielokryterialnych, szczególnie w sytuacjach, gdzie kompromis jest pożądanym.
- **W3C** (*World Wide Web Consortium*, pol. *konsorcjum World Wide Web*) - międzynarodowa organizacja, która zajmuje się tworzeniem standardów dla World Wide Web.

- **WCAG** (*Web Content Accessibility Guidelines*, pol. wytyczne dostępności treści sieciowych) - zestaw wytycznych dotyczących uczynienia treści internetowych bardziej dostępnymi dla osób z różnymi niepełnosprawnościami.
- **WSGI** (*Web Server Gateway Interface*, pol. interfejs bramy serwera sieciowego) - specyfikacja, która opisuje jak serwer sieci web komunikuje się z aplikacjami webowymi i jak te aplikacje mogą być łańcuchowane do przetwarzania żądań HTTP.
- **XHTML** (*eXtensible HyperText Markup Language*, pol. rozszerzalny język znaczników hipertekstu) - rodzaj HTML, który stosuje bardziej rygorystyczne zasady XML.
- **XSS** (*Cross-Site Scripting*, pol. skryptowanie międzystronicowe) - typ ataku na strony internetowe, który umożliwia wstrzyknięcie obcego skryptu do strony widzianej przez użytkownika.

Wprowadzenie do pracy doktorskiej

W ciągu ostatnich dziesięcioleci dziedzina wielokryterialnego podejmowania decyzji (MCDM) przeżywała dynamiczny rozwój, co zostało zaakcentowane w tej rozprawie doktorskiej. W dzisiejszym dynamicznie rozwijającym się świecie informatyki technicznej i telekomunikacji, coraz większe znaczenie zyskują metody pozwalające na efektywną analizę i porównanie rozmaitych opcji. Metoda porównań parami stanowi bardzo solidne narzędzie w tym procesie, umożliwiając precyzyjną ocenę i selekcję najlepszych rozwiązań z wielu dostępnych alternatyw.

Dlaczego porównania parami są tak ważne w naszej dziedzinie? Odpowiedź tkwi w zdolności tej metody do przetwarzania złożonych decyzji w sposób, który jest intuicyjny i dostępny nawet dla tych, którzy nie są ekspertami w dziedzinie technicznej. W kontekście informatyki technicznej i telekomunikacji, porównania parami oraz oparte na niej wielokryterialne metody podejmowania decyzji umożliwiają m.in. efektywne projektowanie sieci, alokację zasobów, ocenę jakości usług (QoS), a także zarządzanie ryzykiem.

Chcąc już na wstępie podać przykłady rzeczywistych aplikacji takich metod w zastosowaniach dyscypliny informatyki i telekomunikacji, byłoby to między innymi projektowanie efektywnych sieci telekomunikacyjnych, gdzie porównania parami pomagają w ocenie różnych konfiguracji pod kątem kosztu, niezawodności, wydajności i skalowalności^{1 2}. W alokacji zasobów, taka metoda pozwala na optymalne przydzielanie szerokości pasma, mocy i zasobów obliczeniowych, co jest kluczowe dla utrzymania wydajności sieci³. W kontekście QoS, porównania parami umożliwiają holistyczną ocenę jakości usług, uwzględniając przepustowość, opóźnienie, utratę pakietów i inne kryteria⁴. Zarządzanie ryzykiem również korzysta na tej metodzie, umożliwiając identyfikację i uszeregowanie strategii ograniczania ryzyka na podstawie wielu czynników⁵.

Centralnym aspektem tych podejść jest wykorzystanie macierzy porównań parami jako głównego narzędzia na którym operuje większość wielokryterialnych metod wspomagania decyzji. Macierze te nie tylko odgrywają kluczową rolę w powyższych metodach, ale są również stosowane w wielu innych obszarach. Niemniej jednak, badacze

¹ Pu, C. (2021). Pro^NDN: MCDM-Based Interest Forwarding and Cooperative Data Caching for Named Data Networking. *Journal of Computer Networks and Communications*, 2021, 1-16.

² Najeeb, Z. A. (2017). Evaluation of Combined Sewer Network Design Using GIs and Multi Criteria Decision Making (MCDM). *Al-Nahrain Journal for Engineering Sciences*, 20(5), 1143-1153.

³ Mohamed H, Al-Masri E, Kotevska O, Soury A. A Multi-Objective Approach for Optimizing Edge-Based Resource Allocation Using TOPSIS. *Electronics*. 2022; 11(18):2888.

⁴ Park, J. J. H., & Jeong, H. Y. (2013). The QoS-based MCDM system for SaaS ERP applications with Social Network. *The Journal of Supercomputing*, 66, 614-632.

⁵ Kuraś, P., Smalara, K., Organiściak, P., Ćmil, M., & Bączek, E. A COMPARISON OF EXISTING RISK ANALYSIS METHODS USED IN INFORMATION SECURITY SYSTEM AUDITS AND THE POTENTIAL FOR THEIR DEVELOPMENT.

napotykali trudności związane z efektywnym tworzeniem losowych macierzy porównań parami, szczególnie w kontekście symulacji Monte Carlo, które potrzebują dużej ilości danych, co jest niezwykle istotne dla badaczy chcących opracowywać nowe skuteczniejsze metody w dziedzinie MCDM.

Problem ten staje się bardziej skomplikowany, gdy mowa o macierzach "dużych" (rozmiarów od 6x6 do 10x10) i jednoczesnym dążeniu do zachowania niskiego współczynnika *CR* - spójności, preferencyjnie poniżej 0,1. Proces tworzenia tysięcy takich macierzy mógł zająć wiele dni, co stanowiło barierę w prowadzeniu badań.

Aby rozwiązać ten problem, wykorzystano algorytmy redukcji niespójności, aby opracować metodę, która znacząco usprawnia ten proces, pomagając badaczom oszczędzać zarówno czas, jak i zasoby obliczeniowe. Niniejsza rozprawa, zatytułowana "Redukcja niespójności w macierzach porównań parami i nowy algorytm generowania losowych macierzy porównań parami o zadanym zakresie współczynników niespójności - FAST-PCM", zagłębia się w złożoność redukcji niespójności w macierzach porównań parami, podstawowym elemencie wielokryterialnych metod podejmowania decyzji. Głównym celem tych badań jest wprowadzenie i analiza algorytmu FAST-PCM. Ten algorytm, oparty na istniejących metodach redukcji niespójności, ma na celu znaczącą optymalizację procesu generowania losowych macierzy porównań z określonym zakresem współczynników niespójności.

Rozpoczynając od dogłębnego przeglądu literatury, autor analizuje koncepcję macierzy porównań parami, ich zastosowania oraz wyzwania związane z niespójnością. Ten teoretyczny fundament stanowi bazę dla empirycznych badań przedstawionych w kolejnych rozdziałach. Drugi rozdział oferuje szczegółową analizę różnych iteracyjnych algorytmów redukcji niespójności, korzystając z metodologii symulacji Monte Carlo. Trzeci rozdział przedstawia bibliotekę REDUCE.py, narzędzie programistyczne oparte na języku Python, dedykowane do minimalizacji niespójności w porównaniach parami. Czwarty rozdział skupia się na aplikacji internetowej REDUCE, przeznaczonej do automatycznej redukcji niespójności w macierzach porównań parami. W piątym rozdziale autor bada potencjał technik uczenia maszynowego w kontekście redukcji niespójności, łącząc tradycyjne podejścia z nowoczesnymi technologiami analizy danych. Kluczowy szósty rozdział omawia algorytm FAST-PCM oraz narzędzie online PC MATRICES GENERATOR. Te innowacje nie tylko potwierdzają tezę rozprawy, ale także wskazują na nowe możliwości w dziedzinie wielokryterialnego podejmowania decyzji.

Teza i cele pracy

W niniejszej pracy doktorskiej przedstawiono algorytm FAST-PCM. Bazując na ugruntowanych metodach redukcji niespójności w macierzach porównań parami, zaproponowano algorytm mający na celu zwiększenie wydajności w generowaniu losowych macierzy porównań o określonym zakresie współczynników niespójności.

Teza rozprawy brzmi następująco:

- **Algorytm FAST-PCM bazujący na znanych metodach redukcji niespójności w macierzach porównań parami umożliwia poprawę wydajności generowania losowych macierzy porównań o zadanym zakresie współczynników niespójności.**

Głównymi celami pracy są:

- Numeryczne porównanie skuteczności algorytmów iteracyjnych w redukcji niespójności w porównaniach parami
- Badanie skuteczności algorytmów uczenia maszynowego w kontekście redukcji niespójności macierzy porównań parami oraz ich porównanie z algorytmami iteracyjnymi
- Przedstawienie założeń algorytmu FAST-PCM oraz badanie jego skuteczności

Dodatkowo, w trakcie realizacji pracy, przyjęto następujące cele dodatkowe:

- Stworzenie biblioteki programistycznej REDUCE.py
- Opracowanie ogólnodostępnej aplikacji REDUCE
- Wprowadzenie ogólnodostępnej aplikacji PC MATRICES GENERATOR, która wspiera proces generowania macierzy porównań z wykorzystaniem algorytmu FAST-PCM

Każdy z tych rozdziałów wnosi wkład w dziedzinę wielokryterialnego podejmowania decyzji, a całość pracy ma na celu nie tylko przedstawienie nowych metod i technik, ale także ich praktyczne zastosowanie w rzeczywistych problemach badawczych.

Publikacje i wystąpienia towarzyszące rozprawie doktorskiej

Przygotowując niniejszą rozprawę doktorską, autor pracy wraz z zespołem badawczym, opublikował następujące artykuły w czasopismach naukowych:

- P. Kuraś - **Pairwise comparison matrices, their applications, properties and the problem of inconsistency reduction** – w *Advances in IT and Electrical Engineering* (dopuszczone do druku) - 2023
- J. Mazurek, R. Perzina, D. Strzałka, B. Kowal, P. Kuraś - **A Numerical Comparison of Iterative Algorithms for Inconsistency Reduction in Pairwise Comparisons** – w *IEEE Access* – 2021
- P. Kuraś, B. Kowal, D. Strzałka, J. Mazurek - **REDUCE: A Python module for reducing inconsistency in Pairwise Comparison Matrices** – w *Advances in Science and Technology Research Journal* – 2023
- P.Kuraś, A. Gerka - **Using Inconsistency Reduction Algorithms in Comparison Matrices to Improve the Performance of Generating Random Comparison Matrices with a Given Inconsistency Coefficient Range** – w *Advances in Science and Technology Research Journal* – 2023

Towarzyszyły jej również rozdziały w monografiach naukowych:

- B. Kowal, P. Kuraś, D. Strzałka, J. Mazurek, R. Perzina - **REDUCE – an online decision support tool for reduction of inconsistency in multiplicative pairwise comparisons** – w publikacji *3rd International conference on Decision making for Small and Medium-Sized Enterprises DEMSME 2021. Conference Proceedings*
- P. Kuraś, P. Organiściak, B. Kowal, E. Łukasik - **USING MACHINE LEARNING TECHNIQUES TO REDUCE PAIRWISE COMPARISON MATRIX INCONSISTENCIES.** – w publikacji *4th International conference on Decision making for Small and Medium-Sized Enterprises DEMSME 2023. Conference Proceedings.*

A także wystąpienia konferencyjne:

- P. Kuraś - **REDUCE – an online decision support tool for reduction of inconsistency in multiplicative pairwise comparisons** – DEMSME 2021, Karvina, Czechy, 13.05.2021
- P. Kuraś - **Using Inconsistency Reduction Algorithms in Comparison Matrices to Improve the Performance of Generating Random Comparison Matrices with a Given Inconsistency Coefficient Range** – CMES 2022, Zamość, 25.11.2022
- P. Kuraś - **USING MACHINE LEARNING TECHNIQUES TO REDUCE PAIRWISE COMPARISON MATRIX INCONSISTENCIES** – DEMSME 2023, Karvina, Czechy, 18.05.2023
- P. Kuraś - **REDUCE: A Python module for reducing inconsistency in Pairwise Comparison Matrices** – CMES 2023, Puławy, 24.11.2023

1. Przegląd literatury – macierze porównań parami, ich zastosowania, właściwości oraz problem redukcji niespójności

Słowa kluczowe rozdziału:

AHP, porównania parami, redukcja niespójności

Abstrakt rozdziału:

Po wprowadzeniu oraz przedstawieniu tezy i celów pracy doktorskiej, w niniejszym rozdziale przedstawiona zostanie koncepcja macierzy porównań parami oraz ich zastosowania, a w szczególności prezentacja wielokryterialnych metod podejmowania decyzji (MCDM): PROMETHEE, PAPRIKA, VIKOR, ELECTRE, BWM (Best-Worst Method) oraz AHP (Analytic Hierarchy Process). Omówione zostaną również właściwości takich macierzy oraz problem niespójności ocen ekspertów/respondentów w takich macierzach, wskaźniki wykorzystywane do oceny niespójności macierzy (CI, CR, KI, GCI, GWI, PLI, TGCI, REI, HCI) oraz zagadnienia związane z redukcją niespójności macierzy.

1.1 Wprowadzenie

W pierwszym rozdziale rozprawy zostanie przedstawiony przegląd macierzy porównań parami, ich zastosowań oraz różnych metod MCDM wykorzystywanych do ich analizy. Omówione zostaną wady i zalety każdej z metod, a także ich przydatność w różnych scenariuszach decyzyjnych, właściwości takich macierzy, w tym ich spójność i przechodniość oraz problem niespójności w ocenach ekspertów lub respondentów.

Do oceny niespójności macierzy porównań parami można użyć kilka wskaźników, takich jak wskaźnik spójności (CI), współczynnik spójności (CR), wskaźnik Koczkodaja (KI) [80], geometryczny wskaźnik spójności (GCI) [81], Golden-Wang Index (GWI) [82], Palaez-Lamata Index (PLI) [83], Triads Geometric Consistency Index ($TGCI$) [84], Relative Error Index (REI) [85] oraz Harmonic Consistency (HCI) [86]. Każdy z tych indeksów zostanie szczegółowo przedstawiony i zilustrowane zostanie ich zastosowanie w ocenie spójności macierzy porównań parami.

Na koniec zostanie opisany problem redukcji niespójności macierzy porównań parami, w tym różne techniki i strategie, które można stosować do poprawy spójności macierzy.

1.2. Porównania parami i macierze porównań parami

Macierze porównań parami są cennym narzędziem w procesach decyzyjnych, które obejmują ocenę zbioru podmiotów na podstawie określonych kryteriów. W macierzy porównań parami preferencje jednego podmiotu względem drugiego są systematycznie porównywane i zapisywane w macierzy. Wynikowa macierz zapewnia sposób oceny względnego znaczenia każdego podmiotu w odniesieniu do innych.

Na początek tworzony jest zbiór danych X , reprezentujący zbiór n elementów do porównania. Preferencje i -tego elementu w stosunku do j -tego podmiotu są zapisywane we wpisie a_{ij} macierzy porównań parami. Należy zauważyć, że wartości a_{ij} muszą spełniać określone warunki, aby macierz miała znaczenie. W szczególności wartości $a_{ij} \in \mathbb{R}$ muszą być nieujemne i spełniać właściwość wzajemności (1.1), tak że dla $a_{ij} > 0; \forall i, j \in \{1, 2, \dots, n\}$,

$$a_{ij} = \frac{1}{a_{ji}}, \forall i, j \in \{1, 2, \dots, n\}. \quad (1.1)$$

Właściwość opisana w równaniu (1.1) jest kluczowa dla ustanowienia multiplikatywnych porównań parami. Aby skonstruować macierz porównań parami (PCM)

dla zbioru n podmiotów, wszystkie możliwe porównania parami można zapisać w macierzy kwadratowej $n \times n$ (1.2).

$$\begin{pmatrix} 1 & a_{12} & \dots & a_{1n} \\ a_{21} & 1 & \dots & a_{2n} \\ \dots & \dots & 1 & \dots \\ a_{n1} & a_{n2} & \dots & 1 \end{pmatrix}. \quad (1.2)$$

Należy zauważyć, że aby PCM była ważna i przydatna w procesie podejmowania decyzji, wartości macierzy muszą spełniać określone warunki. W szczególności, właściwość wyrażona w równaniu (1.1) jest wymagana do ustanowienia multiplikatywnych porównań parami. Właściwość ta zapewnia, że porównania parami są wzajemne, co oznacza, że preferencja jednego podmiotu w stosunku do drugiego jest odwrotnością preferencji drugiego podmiotu w stosunku do pierwszego. Spełniając tę właściwość, PCM można stosować do oceny względnego znaczenia każdego podmiotu i informowania o procesach decyzyjnych.

Zakładając, że wszystkie macierze porównań parami są wzajemne, możemy zdefiniować metodę porównań parami jako dowolną metodę podejmowania decyzji, która obejmuje systematyczne porównywanie i rejestrowanie preferencji jednego podmiotu w stosunku do drugiego. Z drugiej strony, metoda priorytetyzacji jest dowolną procedurą, która wyprowadza wektor priorytetów $w = (w_1, \dots, w_n)$ - wektor wag wszystkich n porównywanych obiektów - z macierzy porównań parami $n \times n$.

Można powiedzieć, że wektor priorytetów w jest powiązany z macierzą porównań parami A lub że wektor priorytetów w jest wyprowadzany metodą generowania priorytetów na podstawie macierzy porównań parami A . Powszechnie jest normalizowanie wektora priorytetów w - jak w przykładzie równania (1.3), co oznacza, że suma jego elementów jest równa jeden:

$$\sum_{i=1}^n w_i = 1. \quad (1.3)$$

Pozwala to na interpretację wektora priorytetów jako zestawu wag, które można wykorzystać do porównania i uszeregowania podmiotów na podstawie ich względnego znaczenia. Celem metod porównywania parami jest uszeregowanie porównywanych obiektów od najlepszego do najgorszego w oparciu o wyprowadzony wektor priorytetów $w = (w_1, \dots, w_n)$. Wartości wektora priorytetu w można wykorzystać do ustalenia rankingu podmiotów, przy czym wyższe wartości odpowiadają wyższemu priorytetowi lub

większemu względnemu znaczeniu. Możliwe są również remisy w rankingu, co wskazuje, że wiele podmiotów ma ten sam poziom ważności w odniesieniu do ocenianych kryteriów.

Wykorzystując macierze porównań parami i metody ustalania priorytetów, decydenci mogą systematycznie oceniać i porównywać podmioty w oparciu o różne kryteria oraz zastosować wynikowe rankingi do informowania o swoich procesach decyzyjnych.

1.2.1 Spójność macierzy porównań parami

Pojęcie (kardynalnej) spójności dla porównań parami można formalnie zdefiniować w następujący sposób - macierz $A_{n \times n} = [a_{ij}]$ jest uważana za spójną wtedy i tylko wtedy, gdy dla wszystkich indeksów i, j i k między 1 a n spełniony jest następujący warunek (1.4):

$$a_{ik} = a_{ij} \cdot a_{jk} \quad \forall i, j, k \in \{1, \dots, n\}. \quad (1.4)$$

Ważne jest, aby pamiętać, że jeśli macierz jest spójna, to jest również odwrotna, ale odwrotne stwierdzenie niekoniecznie jest prawdziwe. Co więcej, jeśli macierz porównań parami jest w pełni spójna, to wszystkie jej wiersze (lub kolumny) są współliniowe, co oznacza, że wszystkie są wielokrotnościami siebie nawzajem [9].

Innym sposobem wyrażenia idei spójności w kontekście macierzy porównań parami jest stwierdzenie, że wszystkie wiersze lub kolumny są do siebie proporcjonalne, aż do stałej mnożenia. Alternatywnie można powiedzieć, że wszystkie wiersze (lub kolumny) są liniowo zależne od siebie, aż do współczynnika skalowania. Mazurek w [9] podaje propozycję związku między wpisami spójnej macierzy PC a jej wektorem priorytetów, co można udowodnić zgodnie z pracą [10]. Niech $A_{n \times n} = [a_{ij}]$ będzie spójną macierzą porównań parami. Wtedy istnieje unikalny (aż do stałej mnożenia) wektor priorytetów $w = (w_1, \dots, w_n)$ związany z macierzą A spełniający równanie (1.4)

1.2.2 Metody ustalania priorytetów w macierzach porównań parami

Istnieją dwie podstawowe techniki uzyskiwania wektora priorytetów z macierzy porównań parami: metoda średniej geometrycznej (metoda najmniejszych kwadratów logarytmicznych) i metoda wartości własnej (wektora własnego). Saaty [10] wprowadził metodę wartości własnych, podczas gdy Crawford i Williams wprowadzili metodę średniej geometrycznej [12,17]. Oprócz tych dwóch podstawowych metod, w literaturze zaproponowano również inne podejścia, takie jak AMM [11], LSM [13, 14, 15] oraz Siraj i in. [16].

1.2.2.1 Metoda średniej geometrycznej

Wektor priorytetów w w metodzie średniej geometrycznej (GMM) dla macierzy $A_{n \times n} = [a_{ij}]$ uzyskuje się poprzez obliczenie średniej geometrycznej wszystkich wierszy A przy użyciu wzoru (1.6) [9]:

$$w_i = \frac{(\prod_{k=1}^n a_{ik})^{\frac{1}{n}}}{\sum_{j=1}^n (\prod_{k=1}^n a_{jk})^{\frac{1}{n}}}, \forall i. \quad (1.6)$$

Można to również wyrazić jako rozwiązanie problemu programowania nieliniowego, określonego przez równania (1.7, 1.8):

$$\min \sum_{i=1}^n \sum_{j=1}^n \left(\ln a_{ij} - \ln \frac{w_i}{w_j} \right)^2 \quad (1.7)$$

$$\text{z zastrzeżeniem } \sum_{i=1}^n w_i = 1, w_i \geq 0, \forall i. \quad (1.8)$$

Należy zauważyć, że wektor priorytetów w jest znormalizowany. Metoda GMM przyjmuje średnią geometryczną każdego wiersza w macierzy, a następnie normalizuje wynikowe wartości w celu utworzenia wektora priorytetów. Innymi słowy, znajduje średnią każdego wiersza w macierzy, a następnie skaluje wartości tak, aby sumowały się do 1. Ta metoda (jak oczywiście i inne metody) - jest wykorzystywana w procesach decyzyjnych, w których należy porównać i uszeregować wiele kryteriów, i pomaga określić względne znaczenie każdego kryterium.

1.2.2.2 Metoda wartości własnej (wektora własnego)

W metodzie wartości własnych (z ang. *eigenvalue* - EV) wektor priorytetów uzyskuje się przez zrównanie go z odpowiednim wektorem własnym odpowiadającym największej wartości własnej macierzy porównań parami $A_{n \times n} = [a_{ij}]$. Można to wyrazić następującym równaniem [62,63] (1.9):

$$A \cdot w = \lambda_{max} \cdot w, \lambda_{max} \geq n. \quad (1.9)$$

W (1.9) λ_{max} jest dodatnią wartością własną o wartości większej lub równej n , której istnienie gwarantuje twierdzenie Perrona-Frobeniusa. Wektor w jest odpowiednim prawym wektorem własnym A .

Mówiąc bardziej bezpośrednio, metoda znajduje największą wartość własną i odpowiadający jej wektor własny macierzy i zrównuje wektor priorytetowy z tym wektorem własnym. Wynikowy wektor priorytetów reprezentuje względne znaczenie każdego porównywanego kryterium. Zazwyczaj przyjmuje się [62], że w jest znormalizowany, co oznacza, że jego długość wynosi 1 lub $\|w\| = 1$. Metoda ta jest powszechnie stosowana [62-64] do ustalania priorytetów wielu kryteriów w oparciu o porównania parami i pomaga określić względne znaczenie każdego kryterium.

1.2.2.3 Metoda średniej arytmetycznej

Oprócz dwóch głównych metod wspomnianych wcześniej, istnieją inne metody, które zostały zaproponowane w literaturze. Jedną z takich metod jest metoda średniej arytmetycznej (AMM) [9], znana również jako metoda znormalizowanych wartości średnich.

AMM wyprowadza wektor priorytetów w z macierzy porównań parami $A = [a_{ij}]$, dzieląc każdą kolumnę macierzy przez jej sumę (normalizując kolumnę), a następnie obliczając wagę i -tego obiektu jako średnią arytmetyczną i -tego wiersza. Wzór (1.10) na wagę i -tego obiektu przy użyciu AMM jest następujący:

$$w_i = \frac{1}{n} \sum_{j=1}^n \frac{a_{ij}}{c_j}, \quad (1.10)$$

gdzie c_j jest sumą j -tej kolumny macierzy A . Normalizacja zapewnia, że każdy obiekt jest oceniany w stosunku do innych, a średnia arytmetyczna nadaje równą wagę każdemu obiektowi. Podobnie jak inne metody, AMM również spełnia ograniczenie normalizacji $\|w\| = 1$.

1.2.2.4 Metoda najmniejszych kwadratów

Metoda najmniejszych kwadratów (LSM) to metoda określania wektora priorytetów w na podstawie macierzy porównań parami $A = [a_{ij}]$. LSM została wprowadzona przez nieznanego autora, a jej wyprowadzenie można znaleźć w różnych źródłach, takich jak [13], [14] lub [15]. LSM określa wektor wag w poprzez rozwiązanie następującego (1.11, 1.12) problemu optymalizacyjnego:

$$\min \sum_{i=1}^n \sum_{j=1}^n \left(a_{ij} - \frac{w_i}{w_j} \right)^2 ; \forall i, j, \quad (1.11)$$

$$\text{z zastrzeżeniem } w_i > 0, \forall i \quad \|w\| = 1. \quad (1.12)$$

Funkcją celu jest suma kwadratów różnic pomiędzy elementami macierzy porównań parami A i stosunkiem wag odpowiednich obiektów. Problem optymalizacji ma na celu znalezienie wektora wag w , który minimalizuje tę funkcję celu, jednocześnie spełniając ograniczenia, że $w_i > 0$ dla wszystkich indeksów i oraz $\|w\| = 1$.

LSM można interpretować jako metodę znajdowania wag, które najlepiej pasują do macierzy porównań parami A , w sensie najmniejszych kwadratów. Jest to również popularna metoda w dziedzinie wielokryterialnej analizy decyzyjnej i jest szeroko stosowana w różnych metodach MCDM (na przykład w podjęciu do metody AHP zaproponowanym przez Bozókiego i Lewisa [137]).

1.2.2.5 Metoda Siraja i in.

Siraj i in. zaproponowali interesującą metodę wyznaczania wektora priorytetowego macierzy porównań parami w oparciu o drzewa rozpinające grafowej reprezentacji macierzy [16]. Ostateczny wektor priorytetów jest obliczany jako średnia arytmetyczna wektorów priorytetów wszystkich drzew rozpinających, gdzie liczba drzew rozpinających dla kompletnej macierzy porównań parami jest określona wzorem Cayleya, $\eta = n^{n-2}$. Całe równanie dla metody jest przedstawione we wzorze (1.13):

$$w = \frac{1}{\eta} \sum_{s=1}^{\eta} w(\tau_s), \quad (1.13)$$

gdzie $w(\tau_s)$ oznacza wektory priorytetów poszczególnych drzew rozpinających. Metoda ta ma oczywistą wadę, którą jest duża liczba drzew rozpinających dla kompletnej macierzy porównań parami. Aby temu zaradzić, autorzy sugerują zastosowanie odpowiednich technik próbkowania. Metoda priorytetyzacji drzew rozpinających była dalej rozwijana w kolejnych badaniach [18] i [19].

W [20] wykazano, że metoda drzewa rozpinającego jest równoważna metodzie średniej geometrycznej (wierszowej) dla kompletnej macierzy porównań parami, podczas gdy badanie [21] udowodniło równoważność również dla niekompletnych macierzy porównań parami. Oznacza to, że metoda drzewa rozpinającego jest ważną alternatywą dla metody średniej geometrycznej, szczególnie w przypadku dużych macierzy porównań parami.

Ogólnie rzecz biorąc, metoda drzewa rozpinającego zapewnia unikalne podejście do określania wektora priorytetów macierzy porównań parami, a jej równoważność z metodą średniej geometrycznej pokazuje jej ważność i przydatność w różnych zastosowaniach.

1.2.2.6 *Inne interesujące podejścia*

Wektory priorytetów uzyskane różnymi metodami dla macierzy porównań parami (PC) mogą się różnić, zwłaszcza gdy macierz jest niespójna. Jednak w przypadku spójnej macierzy PC wektory priorytetów uzyskane metodami wektora własnego (EV) i średniej geometrycznej (GMM) są identyczne. Nawet w przypadku niespójnych macierzy PC z $n = 3$, wektory priorytetów uzyskane za pomocą obu metod mogą być takie same. W przypadku niespójnych macierzy metody EV i GM dają bardzo podobne wyniki, a różnice są bliskie zeru, jak pokazano w [22].

W różnych badaniach porównywano wydajność numeryczną różnych metod ustalania priorytetów, np. w [23], [11], [24] i [25]. Ishizaka i Lusti [11] stwierdzili, że metoda lewej wartości własnej i metoda prawej wartości własnej dają najbardziej różne wyniki, a sprzeczności w rankingach między różnymi metodami (takimi jak metoda średniej wartości znormalizowanej, metoda lewej wartości własnej, metoda prawej wartości własnej i metoda średniej geometrycznej) rosną niemal liniowo wraz z rozmiarem macierzy i wartością współczynnika spójności (CR).

Oprócz powszechnie [9] stosowanych metod priorytetyzacji, takich jak metoda wektorów własnych (EV) i metoda średniej geometrycznej (GMM), w literaturze zaproponowano kilka innych metod. Metody sumy wierszy/średniej zostały wprowadzone przez Ra i Saaty'ego [138, 139], w których wektor priorytetów uzyskuje się poprzez sumowanie lub uśrednianie wierszy macierzy porównań parami (PC). Mazurek i Ramik zaproponowali nową metodę priorytetyzacji opartą na warunku zachowania kolejności [26]. W literaturze można również znaleźć inne rzadziej stosowane metody, takie jak metoda chi-kwadrat (CSM), metoda gradientowej wagi własnej (GEM), metoda najmniejszej odległości (LDM), geometryczna metoda najmniejszych kwadratów (GLSM) lub metoda programowania celów (GPM) [27-31]. Niektóre z tych metod, takie jak metoda Chi-kwadrat i metoda gradientowej wagi własnej (GEM), zostały zastosowane do niewzajemnych macierzy PC, jak omówiono w badaniach [29] i [32].

Ogólnie rzecz biorąc, wybór metody ustalania priorytetów zależy od charakterystyki macierzy PC i konkretnego zastosowania. Podczas gdy niektóre metody mogą dawać podobne wyniki, inne mogą mieć zalety pod względem wydajności obliczeniowej lub odporności na niespójności.

1.2.3 Skale dla macierzy porównań parami

W przypadku porównywania parami stosuje się różne skale. Najczęściej jest stosowana podstawowa skala Saaty'ego [10], która przyjmuje wartości (całkowite) od 1 do 9 i obejmuje odwrotności, jak widać w Tabeli 1.1. Istnieją jednak inne skale liniowe z 10, 5 lub 3 punktami, a nawet skale ciągłe, takie jak \mathbb{R}_0^+ . Zaproponowano również skale nieliniowe, takie jak skale logarytmiczne, wykładnicze i oparte na funkcji logistycznej [33-40].

Tabela 1.1. Skala Saaty'ego. Źródło [10].

Numer w skali	Definicja
1	Równe znaczenie
2 (1/2)	Słabe lub niewielkie znaczenie
3 (1/3)	Umiarkowane znaczenie
4 (1/4)	Umiarkowane + znaczenie
5 (1/5)	Duże znaczenie
6 (1/6)	Znaczenie silne +
7 (1/7)	Bardzo duże znaczenie
8 (1/8)	Bardzo, bardzo duże znaczenie
9 (1/9)	Skrajne znaczenie

Przeprowadzono badania w celu określenia najlepszej skali do porównań parami, ale wyniki nie są jednoznaczne. Dong i in. [41] stwierdzili, że skala geometryczna była najlepsza, podczas gdy Elliott stwierdził, że żadna skala nie odzwierciedlała dokładnie preferencji wszystkich osób [42]. Tri-antaphyllou i in. porównali 78 skal i stwierdzili, że żadna z nich nie była lepsza od pozostałych [43]. Franek i Kresta stwierdzili, że skala Saaty'ego jest nadal korzystna [44], ale wyższą spójność można osiągnąć za pomocą skali pierwiastkowej lub logarytmicznej. Cavallo i Ishizaka przeprowadzili badanie opinii i stwierdzili, że odwrotna skala liniowa była najdokładniejsza [45], podczas gdy skale potęgowe i geometryczne były najmniej dokładne.

W innym badaniu Ishizaka i in. [46] porównali siedem różnych skal i stwierdzili, że skala logarytmiczna stosowana z agregacją multiplikatywną najczęściej wybiera kompromis pożądaną przez teorię wyboru konsumenta. Siraj i in. przeprowadzili symulacje Monte Carlo w celu zbadania niespójności macierzy PC dla różnych skal i stwierdzili, że możliwość uzyskania akceptowalnej (tj. posiadającej niski współczynnik niespójności) macierzy PC jest bardzo wrażliwa na wybraną skalę [47]. Stwierdzili również, że najbardziej ogólną znaczącą skalą dla porównań parami jest skala dodatnich liczb rzeczywistych.

Podczas gdy niektóre badania zalecają pewne skale nad innymi, nie ma wyraźnego konsensusu co do najlepszej skali dla porównań parami. Ostatecznie wybór skali zależy od preferencji decydenta i konkretnego kontekstu podejmowanej decyzji.

1.3. Wielokryterialne metody podejmowania decyzji (MCDM)

Wielokryterialne metody podejmowania decyzji (MCDM) to zestaw podejść do podejmowania decyzji, który obejmuje ocenę i wybór alternatyw w oparciu o wiele kryteriów lub atrybutów. Podejście to jest stosowane w wielu dziedzinach, takich jak finanse, inżynieria, biznes i zarządzanie środowiskiem, gdzie decydenci muszą wybrać najlepszą alternatywę z zestawu opcji, które mają różne poziomy wydajności w różnych kryteriach [48,49].

Metody MCDM zapewniają systematyczne i ustrukturyzowane podejście do oceny i rankingu alternatyw w oparciu o wiele kryteriów. Metody te mają na celu uchwycenie preferencji osób podejmujących decyzje oraz zapewnienie przejrzystego i obiektywnego procesu decyzyjnego. Podejście to opiera się na założeniu, że decydenci mają jasne zrozumienie kryteriów i ich względnego znaczenia. Kryteria mogą być różnego rodzaju, takie jak ilościowe lub jakościowe, obiektywne lub subiektywne, i mogą mieć przypisane różne wagi lub priorytety [50-54]. Wielokryterialne metody podejmowania decyzji zapewniają ramy do rozważenia tych kryteriów w spójny i uporządkowany sposób. W kolejnych podrozdziałach omówione zostaną różne metody MCDM, które zostały opracowane w celu rozwiązania tych problemów decyzyjnych.

1.3.1. Analityczny proces hierarchiczny (AHP)

Analityczny proces hierarchiczny (AHP), jako skuteczne narzędzie decyzyjne, został opracowany przez Saaty'ego [62-64]. AHP umożliwia decydentom ilościowe określenie względnych priorytetów dla danego zestawu alternatyw w skali proporcji, w oparciu o ich ocenę. Podkreśla znaczenie intuicyjnych ocen decydentów, a także spójność porównania alternatyw w procesie decyzyjnym [62].

Podejście AHP dobrze pasuje do zachowania decydentów, którzy podejmują decyzje w oparciu o swoją wiedzę i doświadczenie. Technika ta zapewnia ustrukturyzowane, ale stosunkowo proste rozwiązanie problemów decyzyjnych poprzez systematyczne organizowanie zarówno czynników materialnych, jak i niematerialnych [64]. Sprawia to, że proces decyzyjny jest bardziej przejrzysty i pomaga uniknąć stroniczych decyzji.

Unikalną cechą podejścia AHP jest jego zdolność do rozdzielania złożonych problemów na mniejsze, łatwiejsze w zarządzaniu komponenty, pozwalając decydentom na

podejmowanie decyzji w oparciu o logiczny, krok po kroku proces. Poprzez proste porównywanie parami, AHP ułatwia połączenie małych elementów z większymi, co skutkuje kompleksowym i spójnym rozwiązaniem [62].

1.3.1.1 Kroki stosowania metody AHP

AHP zapewnia ustrukturyzowane i systematyczne ramy podejmowania decyzji w oparciu o osąd decydentów. Aby zastosować AHP, Saaty opracował sześćoetapowy proces [62-64] w następujący sposób:

Tabela 1.2. Średnia losowa niespójność (RI). Źródła [62],[63],[64].

Rozmiar macierzy	Losowa spójność
1×1	0
2×2	0
3×3	0.58
4×4	0.9
5×5	1.12
6×6	1.24
7×7	1.32
8×8	1.41
9×9	1.45
10×10	1.49

- Zdefiniowanie problemu i ustalenie jego celu.
- Ustrukturyzowanie hierarchii problemu, zaczynając od góry (cele z perspektywy decydenta) i przechodząc przez poziomy pośrednie (kryteria, od których zależą kolejne poziomy) do najniższego poziomu (lista alternatyw).
- Utworzenie zestawu macierzy porównań parami, po jednej dla każdego elementu na poziomie bezpośrednio powyżej, przy użyciu skali Saaty'ego przedstawionej w Tabeli 1.1, aby określić, który element dominuje nad drugim.
- Podanie oceny wymaganej do opracowania zestawu macierzy w poprzednim kroku. Wzajemne oceny są automatycznie przypisywane w każdym porównaniu parami.
- Użycie syntezy hierarchicznej, aby zważyć wektory własne za pomocą wag kryteriów i zsumować wszystkie ważone wpisy wektorów własnych odpowiadające tym na następnym niższym poziomie hierarchii.
- Określenie spójności ocen, obliczając wskaźnik spójności (CI) przy użyciu wartości własnej (λ_{\max}), gdzie n jest rozmiarem macierzy. Spójność oceny można sprawdzić, biorąc współczynnik spójności (CR) z CI wraz z odpowiednią wartością w Tabeli 1.2. Współczynnik CR poniżej 0,10 jest akceptowalny. Jeśli jest wyższy, macierz

ocen jest niespójna, a oceny powinny zostać zweryfikowane i poprawione w celu uzyskania spójnej macierzy.

- Działania opisane w ostatnich trzech krokach są powtarzane dla każdego poziomu w hierarchii.

1.3.1.2 Podejścia do grupowego podejmowania decyzji

Analityczny proces hierarchiczny pozwala na grupowe podejmowanie decyzji poprzez umożliwienie członkom grupy wykorzystania ich wiedzy, wartości i doświadczenia do dekonstrukcji problemów w hierarchię i rozwiązywania ich poprzez podążanie za krokami AHP. Burza mózgów i wymiana pomysłów i spostrzeżeń, która jest nieodłącznym elementem stosowania AHP w grupie, często prowadzi do bardziej kompleksowego zrozumienia i przedstawienia omawianych kwestii. Podczas podejmowania decyzji grupowych z uczestnikami, którzy mają wspólne interesy, typowe jest powstawanie konfliktów między poszczególnymi członkami grupy ze względu na ich unikalne motywacje. Ponieważ jednak członkowie grupy mają wspólny cel, najlepiej jest pracować razem jako grupa, aby osiągnąć konsensus, maksymalizując w ten sposób komunikację i udział każdego członka w decyzji [69].

Korzystanie z dedykowanego oprogramowania AHP, takiego np. jak Expert Choice [65], zmniejsza problem "myślenia grupowego" lub dominacji silnego członka grupy, ponieważ skupia uwagę na konkretnym aspekcie problemu podczas procesu podejmowania decyzji, eliminując dryfowanie od tematu do tematu, co często ma miejsce podczas dyskusji grupowych [67-68]. Struktura ta pozwala nawet nieśmiałym i niezdecydowanym członkom grupy czuć się bardziej komfortowo w wyrażaniu swoich poglądów, zwłaszcza w obszarach, w których posiadają oni wiedzę specjalistyczną. W rezultacie inne procesy decyzyjne, takie jak technika Delphi, które zostały zaprojektowane w celu złagodzenia problemów związanych z myśleniem grupowym i dominacją, mogą nie być już konieczne [66].

Podczas sesji grupowej, grupie może zostać przedstawiona wstępnie przygotowana hierarchia, którą mogą zmodyfikować tak, aby odpowiadała ich rozumieniu problemu. Grupa definiuje kwestie, które mają zostać zbadane i dostosowuje przygotowaną hierarchię lub konstruuje nową hierarchię, aby uwzględnić wszystkie ważne kwestie. Gdy grupa ma bardzo różne punkty widzenia, dekonstrukcja problemu na różnych poziomach może pomóc im poczuć się komfortowo ze złożonym problemem. Każdy członek grupy może wcześniej przedstawić swoje obawy i definicje, a grupa może współpracować w celu określenia ogólnej struktury problemu.

W ten sposób można osiągnąć porozumienie co do celów wyższego i niższego rzędu, uwzględniając wszystkie obawy wyrażone przez członków grupy [69]. Jeśli grupa osiągnęła konsensus w sprawie osądu, tylko ten osąd jest wprowadzany. Jeśli grupa nie jest w stanie osiągnąć konsensusu w sprawie oceny, może użyć techniki głosowania lub wybrać "średnią" ocen [65,69]. Grupa może również zdecydować o przyznaniu wszystkim członkom grupy równej wagi lub różnych wag, które odzwierciedlają ich pozycję w projekcie [69].

1.3.1.3 Analityczny proces sieciowy

Analytic Network Process (ANP) jest narzędziem decyzyjnym używanym w wielokryterialnej analizie decyzyjnej. Jest to bardziej ogólna forma AHP i strukturyzuje problemy decyzyjne jako sieć, a nie hierarchię. Podczas gdy AHP zakłada niezależność między elementami, ANP uwzględnia współzależność kryteriów i alternatyw, pozwalając na bardziej złożone podejmowanie decyzji [75].

ANP ma praktyczne zastosowanie w podejmowaniu złożonych decyzji, a na jej temat napisano kilka książek i artykułów naukowych [75-79]. Oprogramowanie ANP może być używane do pracy z wcześniej podjętymi decyzjami, a korzystanie z ANP obejmuje kilka etapów, w tym zrozumienie problemu decyzyjnego, określenie kryteriów kontroli i podkryteriów, skonstruowanie klastrów sieci, wykonanie sparowanych porównań i podjęcie decyzji na podstawie wyników [75, 76].

1.3.2 Best-Worst Method

Metoda Best-Worst Method (BWM) jest najnowszym dodatkiem do niekompletnych metod porównywania parami. Obejmuje ona porównania parami wszystkich kryteriów (lub innych obiektów, takich jak alternatywy, podkryteria itp.) w odniesieniu do najlepszego i najgorszego kryterium [71]. W BWM decydent porównuje każde kryterium tylko z najlepszym i najgorszym kryterium w celu określenia ich względnego znaczenia, a wagi wszystkich kryteriów są określane poprzez rozwiązanie nieliniowego lub liniowego problemu programowania. Atrakcyjność tej metody polega na jej prostocie. Badania wykazały jednak, że jej wiarygodność rankingów obiektów uzyskanych z porównań parami za pomocą BWM jest słabsza niż w przypadku metody średniej geometrycznej i metody wartości własnych [70].

1.3.2.1 Kroki stosowania BWM

Metoda Best-Worst (BWM), wprowadzona w [72,73], jest metodą porównywania parami stosowaną do określania wag kryteriów na podstawie ich względnego znaczenia. BWM jest atrakcyjna ze względu na jej prostotę i przebiega w następujący sposób [71]:

- Najpierw określany jest zestaw kryteriów.
- Po drugie, decydent identyfikuje najlepsze (najbardziej pożądane) i najgorsze kryterium.
- Po trzecie, preferencje najlepszego kryterium w porównaniu do wszystkich innych kryteriów są określane w skali od 1 (równe znaczenie) do 9 (absolutna preferencja).
- Po czwarte, preferencje wszystkich innych kryteriów w odniesieniu do najgorszego kryterium są określane w tej samej skali.
- Wreszcie, optymalne wagi dla wszystkich kryteriów są znajdowane poprzez rozwiązanie odpowiedniego problemu programowania nieliniowego.
- Optymalny wektor wag, $w = (w_1, w_2, \dots, w_n)$, znajduje się poprzez rozwiązanie następującego (1.14, 1.15) problemu optymalizacyjnego [71]:

$$\min \left(\max_j \left\{ \left| \frac{w_B}{w_j} - c_{Bj} \right|, \left| \frac{w_j}{w_W} - c_{jW} \right| \right\} \right), \quad (1.14)$$

$$\text{z zastrzeżeniem } \sum_{j=1}^n w_j = 1, w_j \geq 0, \forall j = 1, \dots, n. \quad (1.15)$$

Liniowa wersja BWM została wprowadzona przez [74] i [73], gdzie litera L oznacza liniową. Liniowa BWM zmniejsza liczbę wymaganych porównań, co może być ważne w przypadku dużej liczby obiektów. Rozwiązanie liniowej wersji BWM, oznaczane jako w^* , uzyskuje się poprzez rozwiązanie następującego (1.16-1.19) liniowego problemu optymalizacyjnego:

$$\min \xi_L \quad (1.16)$$

z zastrzeżeniem

$$|w_B - c_{Bj}w_j| \leq \xi_L, \quad \forall j = 1, \dots, n, \quad (1.17)$$

$$|w_j - c_{jW}w_W| \leq \xi_L, \quad \forall j = 1, \dots, n, \quad (1.18)$$

$$\sum_{j=1}^n w_j = 1, w_j \geq 0, \quad \forall j = 1, \dots, n. \quad (1.19)$$

Wartość ξ_L^* można uznać za stopień niespójności preferencji. Należy zauważyć, że rozwiązanie liniowej wersji BWM zasadniczo różni się od rozwiązania wersji nieliniowej.

1.3.3 Inne metody MCDM wykorzystujące macierze porównań parami

1.3.3.1. VIKOR

Metoda VIKOR jest narzędziem decyzyjnym służącym do szeregowania i wyboru alternatywnych rozwiązań o sprzecznych kryteriach. Stosuje metodę programowania kompromisowego i metrykę LP [140] do opracowania wielokryterialnej miary rankingu kompromisowego. Algorytm polega na określeniu najlepszych i najgorszych wartości

kryteriów, obliczeniu wartości dla każdej alternatywy i uszeregowaniu ich na podstawie miar bliskości idealnej alternatywy.

Rozwiązanie kompromisowe jest proponowane w oparciu o akceptowalną przewagę i stabilność w podejmowaniu decyzji, z zestawem zaproponowanych rozwiązań kompromisowych, jeśli warunki nie są spełnione. VIKOR jest skuteczny w sytuacjach, w których preferencje decydenta są niejasne i może być wykorzystywany jako podstawa do negocjacji przy użyciu wag kryteriów [105, 106].

1.3.3.2. PROMETHEE

PROMETHEE to rodzina metod rankingowych do wielokryterialnego podejmowania decyzji [107]. Obejmuje ona PROMETHEE I dla częściowego rankingu i PROMETHEE II dla pełnego rankingu alternatyw. Inne wersje, takie jak PROMETHEE III, IV, V, VI, GDSS i GAIA zostały również opracowane dla bardziej złożonych sytuacji decyzyjnych [108-114]. PROMETHEE II polega na porównywaniu parami alternatywnych rozwiązań w oparciu o różne kryteria, które należy zmaksymalizować lub zminimalizować. Metoda ta wymaga określenia wag i funkcji preferencji dla każdego kryterium.

Krokowa procedura implementacji PROMETHEE II obejmuje określenie odchyleń, zastosowanie odpowiednich funkcji preferencji, obliczenie globalnego wskaźnika preferencji, dodatnich i ujemnych przepływów wyprzedzających dla każdej alternatywy i częściowego rankingu oraz przepływu wyprzedzającego netto dla każdej alternatywy i pełnego rankingu [115].

1.3.3.3. ELECTRE

Początki metod ELECTRE można prześledzić wstecz do techniki opartej na sumie ważonej zawartej w metodzie MARSAN do wyboru nowych działań, ale zauważono poważne wady w jej zastosowaniu [116]. W rezultacie skonsultowano się z B. Royem, który w tym samym roku opracował metodę ELECTRE [117]. Od tego czasu ELECTRE ewoluowała i dała początek kilku wersjom, w tym ELECTRE IV, która uwzględniała pojęcie progu weta, oraz ELECTRE IS, która była wykorzystywana do modelowania sytuacji, w których dane były względnie niskiej jakości [118]. Inna wersja, znana jako ELECTRE II, została opracowana do planowania mediów, w odniesieniu do definicji planu reklamowego. Inne metody ELECTRE, takie jak ELECTRE III i ELECTRE IV, zostały opracowane później w celu uwzględnienia różnych sytuacji decyzyjnych [119-135].

Można wymienić kilka głównych cech ELECTRE, w tym ich znaczenie w sytuacjach decyzyjnych, modelowanie z relacją wyprzedzającą, ich strukturę, rolę kryteriów i sposób

uwzględnienia niedoskonałej wiedzy [127]. Metody ELECTRE są stosowane, gdy decydent chce uwzględnić co najmniej trzy kryteria i gdy działania są oceniane na skali porządkowej lub słabo interwałowej lub gdy istnieje silna heterogeniczność związana z charakterem skal związanych z kryteriami. Preferencje w metodach ELECTRE są modelowane przy użyciu binarnych relacji out-ranking, a preferencje są definiowane za pomocą dwóch głównych pojęć: zgodności i niezgodności [136].

Metody ELECTRE mają dwie główne procedury, procedurę agregacji wielu kryteriów i procedurę eksploatacji. Względna rola przypisywana kryteriom w metodach ELECTRE jest definiowana przez dwa odrębne zestawy parametrów: względne współczynniki ważności i progi weta. We fragmencie tym zauważono również, że w metodach ELECTRE należy uwzględnić niedoskonałą wiedzę [127].

1.4. Problem niespójności w macierzach porównań parami

W sekcji 2.1 zauważono, że należy zająć się kwestią (nie)spójności macierzy PC. Wskaźnik spójności Saaty'ego (CI) i współczynnik spójności (CR) są powszechnie stosowane do pomiaru spójności danej macierzy $A n \times n$. $CI(A)$ jest obliczany jako (1.20):

$$CI(A) = (\lambda_{max} - n)/(n - 1). \quad (1.20)$$

gdzie λ_{max} jest największą dodatnią wartością własną A . $CR(A)$ jest zdefiniowany jako $CI(A)$ podzielony przez $RI(n)$, który jest losowym wskaźnikiem spójności wyrażonym jako średnia losowo wygenerowanych macierzy porównań parami danego rzędu ze skalą Saaty'ego zależną od n i przedstawiony w Tabeli 1.2 w niniejszym rozdziale, jako:

$$CR(A) = \frac{CI(A)}{RI(n)}. \quad (1.21)$$

1.4.1. Inne indeksy niespójności

Chociaż CR jest najpopularniejszą definicją losowego wskaźnika spójności, w literaturze dostępne są inne definicje, takie jak te w referencjach [80-86] i opisane w dalszych podrozdziałach niniejszego rozdziału.

1.4.1.1. Indeks Koczkodaja

Indeks Koczkodaja (KI) jest miarą stopnia, w jakim macierz narusza silną nierówność trójkąta, która jest właściwością macierzy, w której suma dowolnych dwóch boków trójkąta utworzonego przez macierz jest zawsze większa lub równa trzeciemu bokowi [82].

$$KI(A) = \max_{i,j,k \in \{1, \dots, n\}} \left\{ \min \left\{ \left| 1 - \frac{a_{ij}}{a_{ik}a_{kj}} \right|, \left| 1 - \frac{a_{ik}a_{kj}}{a_{ij}} \right| \right\} \right\}. \quad (1.22)$$

Indeks jest obliczany poprzez iterację po wszystkich możliwych parach elementów w macierzy i porównanie stosunku tych elementów do iloczynu trzeciego elementu w macierzy (1.22). Minimalna wartość dwóch współczynników jest następnie dodawana do listy minimalnych wartości dla tej pary elementów. Proces ten jest powtarzany dla wszystkich możliwych kombinacji elementów w macierzy. Końcowy indeks Koczkodaja jest obliczany poprzez wzięcie minimalnej wartości z listy minimalnych współczynników dla wszystkich par elementów w macierzy. Otrzymana wartość jest miarą stopnia, w jakim macierz narusza silną nierówność trójkąta. Mniejsza wartość wskazuje na silniejsze naruszenie nierówności, podczas gdy wartość zero wskazuje, że macierz spełnia nierówność.

1.4.1.2. Golden-Wang Index

Golden-Wang Index (*GWI*) jest obliczany poprzez obliczenie sumy każdej kolumny macierzy wejściowej i zapisanie jej w tablicy. Następnie normalizuje macierz, dzieląc każdy element każdej kolumny przez odpowiednią sumę kolumny. Kolejny krok polega na obliczeniu dynamicznych wektorów, które reprezentują średnie kolumn znormalizowanej macierzy. Na koniec algorytm oblicza *GWI*, porównując znormalizowaną macierz z wektorami dynamicznymi i biorąc bezwzględną różnicę między każdym elementem macierzy a odpowiadającym mu elementem wektorów dynamicznych [81]:

$$GWI(A) = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^n |\bar{a}_{ij} - \bar{w}_j|. \quad (1.23)$$

Suma tych różnic jest następnie dzielona przez całkowitą liczbę elementów w macierzy celem uzyskania indeksu Golden-Wang (1.23).

1.4.1.3. Palaez-Lamata Index

Palaez-Lamata Index (*PLI*) jest powszechnie stosowaną miarą do określania spójności macierzy porównań parami [83]. Indeks ten oblicza stopień, w jakim porównania dokonane między różnymi elementami macierzy są logiczne i spójne ze sobą:

$$PLI(A) = \frac{6}{n(n-1)(n-2)} \cdot \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=j+1}^n \left(\frac{a_{ik}}{a_{ij}a_{jk}} + \frac{a_{ij}a_{jk}}{a_{ik}} - 2 \right). \quad (1.24)$$

Aby obliczyć *PLI*, zagnieżdżona struktura pętli jest używana do iteracji po wierszach i kolumnach macierzy, a formuła jest stosowana do każdego elementu, który spełnia określone warunki wg (1.24).

1.4.1.4. (Triads) Geometric Consistency Index

Wskaźnik spójności geometrycznej (*GCI*) najpierw oblicza zestaw wektorów średniej geometrycznej. Następnie iteruje przez górną trójkątną część macierzy, obliczając logarytm stosunku dwóch wartości porównawczych i mnożąc ten wynik przez siebie [85]:

$$GCI(A) = \frac{2}{(n-1)(n-2)} \sum_{i=1}^{n-1} \sum_{j=i+1}^n \left(\ln a_{ij} \frac{w_j}{w_i} \right)^2. \quad (1.25)$$

Suma tych wartości jest dzielona przez współczynnik normalizacji i zwracana jako końcowy wskaźnik spójności geometrycznej (1.25). Istnieje również podobny indeks o nazwie Triads Geometric Consistency Index [85]. Podczas gdy pierwszy algorytm oblicza *GCI* na podstawie logarytmu stosunku dwóch wektorów i wartości w macierzy, drugi indeks jest obliczany na podstawie logarytmu iloczynu trzech wartości w macierzy (1.26):

$$TGCI(A) = \frac{2 \sum_{i < j < k} (\log(a_{ij} a_{jk} a_{ki}))^2}{n(n-1)(n-2)}. \quad (1.26)$$

1.4.1.5. Wskaźnik błędu względnego (*REI*)

Wskaźnik błędu względnego (*REI*) obejmuje iterację przez każdy element danej macierzy i obliczenie dwóch części każdego elementu [80]. Pierwszą część określa się, biorąc logarytm bieżącego elementu w macierzy i dzieląc go przez logarytm stosunku między odpowiednimi elementami w wektorze. Wartość ta jest następnie podnoszona do kwadratu i dodawana do poprzedniej wartości. Drugą część oblicza się, biorąc logarytm bieżącego elementu w macierzy, podnosząc go do kwadratu i dodając do poprzedniej wartości. Po przetworzeniu wszystkich elementów macierzy, ostateczna wartość *REI* jest określana przez podzielenie sumy wszystkich wartości z pierwszej części obliczeń przez sumę wszystkich wartości z drugiej części (1.27):

$$REI(A) = \frac{\sum_{i=1}^n \sum_{j=1}^n \left(\log a_{ij} - \log \frac{w_i}{w_j} \right)^2}{\sum_{i=1}^n \sum_{j=1}^n (\log a_{ij})^2}. \quad (1.27)$$

1.4.1.6. Wskaźnik spójności harmonicznej (*HCI*)

Harmonic Consistency Index (*HCI*) składa się z dwóch głównych etapów [104]. W pierwszym kroku obliczany jest wektor dla każdej kolumny macierzy. Dla każdej kolumny i wiersza algorytm sumuje wszystkie elementy. W drugim kroku obliczana jest średnia harmoniczna (*HM*) wektora. Odbywa się to poprzez wzięcie odwrotności każdego

elementu w wektorze, zsumowanie odwrotności, a następnie wzięcie odwrotności sumy. HM jest następnie wykorzystywana do obliczenia wartości HCI (1.28):

$$HCI(A) = \frac{(HM(A)-n)(n+1)}{n(n-1)}. \quad (1.28)$$

Wartość HCI reprezentuje stopień spójności w porównaniach parami. Wartość 0 oznacza całkowitą niespójność, podczas gdy wartość 1 oznacza całkowitą spójność.

1.5. Algorytmy redukcji niespójności

Ponieważ niespójność w macierzach PC może wynikać z błędów w ludzkiej percepcji i ocenie, nie zawsze jest jasne, gdzie te błędy mogą wystąpić. Aby temu zaradzić, opracowano metody i algorytmy wykrywania i (ewentualnego) korygowania błędnych danych w macierzy A w celu zminimalizowania współczynnika niespójności. Istnieją dwie główne grupy algorytmów do zmniejszania niespójności w macierzach porównań parami: nieiteracyjne i iteracyjne [9].

Algorytmy iteracyjne, w szczególności te zaproponowane przez Cao et al. [88], Szybowski [89] oraz Xu i Wei [90] zostały przetestowane w pracy Mazurek et al. [87] (zamieszczonej także w rozdziale 2 tej rozprawy doktorskiej). Chociaż istnieją co najmniej dwa dodatkowe znane algorytmy opisane w literaturze [87], algorytm Kou et al. [91] został uznany za niestabilny i nieefektywny, a algorytm Mazurek et al. [92] okazał się nie zbiegać do zera niespójności w każdym przypadku. Niektóre algorytmy nieiteracyjne, o których warto wspomnieć, zostały przedstawione w referencjach [93-103].

1.5.1 Algorytm Xu i Wei

Algorytm redukcji niespójności Xu i Wei [90] to metoda stosowana do redukcji niespójności w danej macierzy, która jest powszechnie stosowana w procesach decyzyjnych. Algorytm ten przyjmuje macierz, parametr λ i próg współczynnika spójności (CR) jako dane wejściowe. Oblicza maksymalną wartość własną i wskaźnik spójności (CI) macierzy wejściowej i porównuje ją z indeksem losowym (RI), aby uzyskać początkową wartość CR .

Następnie algorytm stosuje serię obliczeń do macierzy, aż CR spadnie poniżej podanego progu. Obliczenia te obejmują obliczenie wektora, który jest używany do modyfikacji każdego elementu macierzy. Nowa macierz jest następnie wykorzystywana do obliczenia nowej wartości CI i CR , która jest porównywana z progiem, aż do osiągnięcia pożądanego poziomu spójności. Wynikiem algorytmu jest zredukowana macierz, a także odpowiadające jej wartości CI i CR .

1.5.2 Algorytm Szybowskiego

Algorytm redukcji niespójności Szybowskiego to metoda redukcji niespójności w danej macierzy, która została po raz pierwszy zaproponowana przez Szybowskiego w 2018 roku [89]. Algorytm opiera się na spostrzeżeniu, że jeśli macierz jest niespójna, to musi istnieć co najmniej jedna para elementów, która narusza własność przechodności. Algorytm Szybowskiego ma na celu identyfikację i systematyczną korektę tych naruszeń.

Algorytm rozpoczyna się od obliczenia początkowych wartości CI i CR macierzy wejściowej, podobnie jak algorytm Xu i Wei [90]. Następnie stosuje serię modyfikacji macierzy, aż CR spadnie poniżej zadanego progu. Modyfikacje polegają na znalezieniu największej wartości e_{ij} , gdzie e_{ij} jest stosunkiem i -tego elementu do j -tego elementu, i użyciu jej do modyfikacji macierzy. W szczególności algorytm wybiera parę elementów (i,j) , takich że e_{ij} jest największą ze wszystkich wartości e_{ij} w macierzy. Następnie ustawia wartość elementu (i,j) na równą średniej geometrycznej i -tego i j -tego elementu, a wartość elementu (j,i) na równą odwrotności elementu (i,j) [89].

Następnie algorytm ponownie oblicza wartości CI i CR zmodyfikowanej macierzy i sprawdza, czy CR jest poniżej progu. Jeśli CR jest nadal zbyt wysokie, algorytm powtarza proces ze zmodyfikowaną macierzą, aż do osiągnięcia pożądanego poziomu spójności [89].

1.5.3 Algorytm Cao

Algorytm redukcji Cao to kolejna metoda redukcji niespójności w danej macierzy [91]. Podobnie jak algorytm Xu i Wei, opiera się na metodzie, która iteracyjnie modyfikuje macierz wejściową, aż spełni ona wymóg spójności.

Algorytm rozpoczyna się od obliczenia początkowych wartości CI i CR macierzy wejściowej, podobnie jak inne opisane algorytmy [89, 90]. Następnie stosuje serię modyfikacji macierzy, aż CR spadnie poniżej podanego progu. Modyfikacje polegają na pomnożeniu każdego elementu macierzy przez ważoną sumę siebie i swojej odwrotności, która jest oparta na wektorach własnych macierzy. Wagi są określane przez parametr λ , który kontroluje równowagę między oryginalnymi i zmodyfikowanymi elementami macierzy [91].

Mówiąc dokładniej, algorytm najpierw oblicza wektor własny dla każdego wiersza macierzy i wykorzystuje je do obliczenia zmodyfikowanej macierzy poprzez ważenie każdego elementu odpowiednim wektorem własnym wiersza. Waga każdego elementu jest następnie modyfikowana za pomocą parametru λ , tak aby elementy o wyższych wagach były pod większym wpływem odpowiadających im wektorów własnych wiersza.

Następnie algorytm ponownie oblicza wartości CI i CR zmodyfikowanej macierzy i sprawdza, czy CR jest poniżej progu. Jeśli CR jest nadal zbyt wysokie, algorytm powtarza proces ze zmodyfikowaną macierzą, aż do osiągnięcia pożądanego poziomu spójności [91].

1.6. Wnioski

W pierwszym rozdziale przedstawiony został kompleksowy przegląd macierzy porównań parami oraz różnych metod Multiple Criteria Decision Making (MCDM), które są wykorzystywane do analizy tych macierzy. Omówione zostały zastosowania macierzy porównań parami oraz zalety i wady różnych metod MCDM w kontekście ich analizy. W ramach tego przeglądu poruszono również kwestię spójności i przechodniości macierzy porównań parami oraz problem niespójności w ocenach ekspertów lub respondentów.

Przedstawiono również różne wskaźniki używane do oceny niespójności macierzy porównań parami, takie jak wskaźnik spójności (CI), współczynnik spójności (CR), wskaźnik Koczkodaja (KI), geometryczny wskaźnik spójności (GCI), Golden-Wang Index (GWI), Palaez-Lamata Index (PLI), Triads Geometric Consistency Index ($TGCI$), Relative Error Index (REI) oraz Harmonic Consistency (HCI).

W ramach rozważań końcowych, poruszono temat problemu redukcji niespójności macierzy porównań parami. Opisano różne techniki i strategie, które można wykorzystać do poprawy spójności macierzy oraz wyeliminowania niespójności. Rozważania te ukierunkowują na potrzebę opracowania skutecznych metod i narzędzi, które mogą pomóc w minimalizowaniu niespójności i zwiększaniu jakości ocen w procesie podejmowania decyzji opartego na analizie macierzy porównań parami.

Wnioski z tego rozdziału wskazują na istotność i kompleksowość problemu niespójności w analizie macierzy porównań parami oraz na potrzebę skutecznych narzędzi do jej rozwiązania.

2. Numeryczne porównanie algorytmów iteracyjnych redukcji niespójności w porównaniach parami

Słowa kluczowe rozdziału:

Indeksy, Wartości własne i funkcje własne, Modelowanie bryłowe, Programowanie, Metody Monte Carlo, transformacje, podejmowanie decyzji, metody iteracyjne ,algebra macierzy, metody Monte Carlo, redukcja niespójności, porównanie numeryczne, algorytmy iteracyjne, symulacje Monte Carlo, macierze porównań parami, fundamentalna skala Saaty'ego, pierwotne zachowanie preferencji, wektor priorytetu, algorytm krok po kroku, macierz porównania parami, Algorytm Xu i Wei, algorytm, spójność, redukcja niespójności, porównania parami

Abstrakt rozdziału (za oryginalną publikacją):

Celem rozdziału jest porównanie wybranych iteracyjnych algorytmów redukcji niespójności w porównaniach parami za pomocą symulacji Monte Carlo. Przeprowadzamy symulacje dla macierzy porównań parami rzędu $n=4$ i $n=8$ z początkową niespójnością $0.10 < CR < 0.80$ i wpisami wylosowanymi z fundamentalnej skali Saaty'ego. Następnie oceniamy wydajność algorytmów w odniesieniu do czterech miar, które wyrażają stopień zachowania oryginalnych preferencji. Nasze wyniki wskazują, że żaden algorytm nie przewyższa wszystkich innych algorytmów pod względem każdej miary zachowania preferencji. Algorytm Xu i Wei jest najlepszy pod względem zachowania oryginalnego wektora priorytetów i rankingu obiektów, algorytm Step-by-Step najlepiej zachowuje oryginalne preferencje wyrażone w postaci macierzy porównań parami, a algorytm Szybowskiego zachowuje najwięcej wpisów macierzy niezmiennych podczas redukcji niespójności.

2.1. Wstęp

Jedną z najczęściej omawianych właściwości porównań parami jest ich kardynałna lub porządkowa niespójność. Ludzcy eksperci rzadko są w pełni spójni w swoich osądach, co jest szczególnie prawdziwe w przypadku większej liczby porównywanych obiektów.

Niespójność porównań parami jest oceniana za pomocą miar (funkcji) zwanych wskaźnikami niespójności. Być może najbardziej znanymi wskaźnikami są wskaźnik spójności Saaty'ego (CI) i wskaźnik spójności (CR) [42] oraz wskaźnik niespójności Koczkodaja (KI) [22].

Ponieważ porównania parami lub macierze porównań parami wynikające z rozwiązań rzeczywistych problemów rzadko są spójne, zazwyczaj tolerowany jest niski stopień niespójności. Saaty zasugerował w swoim procesie hierarchii analitycznej, że niespójność $CR < 0.10$ jest akceptowalna. Podobnie, porównania parami są uważane za akceptowalnie spójne, jeśli $KI < 0,33$ (inne progi niespójności istnieją również dla innych wskaźników).

Gdy niespójność porównań parami jest niedopuszczalnie wysoka, decydent ma dwie możliwości: poprosić eksperta o zmianę jego osądów lub znaleźć macierz porównań parami, która jest wystarczająco spójna, a jednocześnie jest jak najbardziej zbliżona do oryginalnej macierzy wyrażającej preferencje eksperta.

W tym drugim przypadku w ostatnich dziesięcioleciach w literaturze zaproponowano wiele podejść lub algorytmów redukcji niespójności. Być może pierwszy algorytm redukcji niespójności w porównaniach parami został zaproponowany w [19] w 1996 roku. Algorytm oparty na odległości wyszukiwał najbardziej niespójną triadę (pod względem wskaźnika niespójności Koczkodaja) i zastępował ją spójną. Później ten sam pomysł został zastosowany w [23]. Ogólnie rzecz biorąc, algorytmy redukcji niespójności można podzielić na dwie grupy: iteracyjne i nieiteracyjne. Pierwsze algorytmy dostosowują jeden lub więcej (lub wszystkie) elementów macierzy w każdej iteracji, tak aby niespójność całej macierzy porównań parami (PC) stopniowo malała i zatrzymywała się, gdy poziom niespójności spadnie poniżej określonego progu. Opisy algorytmów znajdują się w rozdziale 1.

Jeśli chodzi o liczbowe porównanie wyżej wymienionych metod, Cao i in. [11] porównali swój algorytm z algorytmem Xu i Wei za pomocą jednej macierzy rzędu $n = 8$. Abel et al. [1] zastosowali dwie macierze PC rzędu $n = 6$ i $n = 8$ do porównania pięciu algorytmów, w tym INSITE. Girsang et al. [18] porównali swój algorytm z jedną inną metodą za pomocą jednej macierzy PC rzędu $n = 4$ oraz z dwoma innymi metodami dla macierzy PC rzędu $n = 8$. Pereira i Costa [39] przedstawili jak dotąd najobszerniejsze

porównanie, z maksymalnie ośmioma algorytmami i jedną macierzą PC odpowiednio rzędu $n = 7$, $n = 8$ i $n = 9$.

Jednak porównanie algorytmów za pomocą dwóch lub trzech przykładów ad hoc nie pozwala na wyciągnięcie żadnych ogólnych wniosków na temat wydajności algorytmów w odniesieniu do zachowania początkowych preferencji lub złożoności obliczeniowej (czasochłonności). Co więcej, w literaturze całkowicie brakuje analizy różnic między algorytmami w odniesieniu do stopnia niespójności i rozmiaru początkowej macierzy PC. Stanowi to dużą i poważną lukę w obecnej wiedzy dotyczącej problemu numerycznej redukcji niespójności w porównaniach parami.

Celem rozdziału jest wypełnienie wspomnianej luki i przedstawienie numerycznego porównania wybranych algorytmów iteracyjnych dla niespójnych macierzy PC rzędu $n = 4$, reprezentujących "małe" macierze PC, oraz $n = 8$, reprezentujących "duże" macierze PC, a także odpowiadających najczęstszemu rozmiarowi macierzy PC używanych do porównań w poprzednich badaniach. Algorytmy iteracyjne są bardziej odpowiednie do symulacji numerycznych, ponieważ algorytmy nieiteracyjne są zwykle oparte na rozwiązywaniu problemów programowania nieliniowego, które są NP-trudne, a zatem mniej odpowiednie do modelowania i symulacji.

W zaproponowanym badaniu generowano dużą próbkę losowych niespójnych macierzy PC danego rzędu i oceniano wydajność algorytmów pod względem zachowania oryginalnych preferencji. Co więcej, dzielono losowo wygenerowane niespójne macierze PC na dwie kategorie: mniej niespójne macierze z $0,10 < CR < 0,30$ i bardziej niespójne macierze z $0,30 \leq CR < 0,80$ i badano każdy przypadek osobno, ponieważ zachowanie algorytmów może się różnić przy różnych stopniach niespójności macierzy (wejściowej).

2.2. Miary zachowania preferencji macierzy porównań parami

Powszechnie przyjmuje się, że podczas procesu poprawy zgodności, pierwotne preferencje ekspertów powinny zostać zachowane w jak największym stopniu. W tej sekcji przedstawiono miary zachowania preferencji (wyrażone w postaci porównań parami), które są wykorzystywane w części numerycznej tego badania. Definicja algorytmu redukcji niespójności:

Definicja [35]: Niech A będzie zbiorem macierzy porównań parami rzędu n i niech μ oznacza miarę niespójności taką, że: $\mu: A \rightarrow R_{0+}$. Wówczas algorytm, którego głównym celem jest przekształcenie macierzy PC $A \in A$ z $\mu(A) = m$ w macierz $A' \in A$ tak, aby $\mu(A') < \mu(A)$ i $\mu(A') \leq \varepsilon, \varepsilon > 0$, nazywamy algorytmem redukcji niespójności (AIR).

W literaturze zaproponowano ostatnio różne miary zachowania preferencji (porównania parami). Xu i Wei [47] wprowadzili dwie miary, δ i σ , gdzie δ jest maksymalną różnicą między elementem oryginalnej macierzy a zmodyfikowaną macierzą, a σ jest równe średniej kwadratowej odległości między wszystkimi elementami oryginalnej macierzy i zmodyfikowanej macierzy. Pereira i Costa [39] zaproponowali nową miarę zwaną całkowitą liczbą punktów odchylenia (*TND*). Abel i in. [1] dodali kilka "miar kompromisu": liczbę naruszeń oceny (*NJV*), która odpowiada liczbie elementów macierzy, które zostały dostosowane, całkowite odchylenie oceny (*TJD*), które jest równe odległości L_1 oryginalnej macierzy i zmodyfikowanej macierzy, kwadratowe całkowite odchylenie oceny (*STJD*), wariant *TJD* oraz liczbę odwróceń oceny (*NJR*).

Po opracowaniu wyżej wymienionych miar zachowania preferencji w ramach porównania parami, zdefiniowano następujące miary zachowania preferencji, które zostaną zastosowane w sekcji numerycznej.

Definicja: Niech $A = (a_{ij})$ będzie niespójną macierzą porównań parami rzędu n i niech $w = (w_1, \dots, w_n)$ będzie wektorem priorytetów związanym z A . Niech $A' = (a'_{ij})$ oznacza macierz PC wyprowadzoną z $A = (a_{ij})$ poprzez AIR i niech $w' = (w'_1, \dots, w'_n)$ będzie odpowiadającym jej wektorem priorytetów. Następnie (2.1) opisuje średnią zmianę wektora priorytetów po transformacji z A do A'

$$d(w, w') = \frac{1}{n} \sum_{i=1}^n |w_i - w'_i|. \quad (2.1)$$

Na przykład $d = 0,03$ można interpretować w taki sposób, że każda waga w_i wektora priorytetów w zmieniła się średnio o 3%. Wektor priorytetów zawiera nie tylko wagi wszystkich porównywanych obiektów, ale także ich ranking od najlepszego (z najwyższą wagą) do najgorszego (z najniższą wagą), ewentualnie z uwzględnieniem remisów.

Kolejna miara, odległość *tau* Kendalla [21], ocenia, jak bardzo ten ranking zmienia się po transformacji.

Definicja: Niech r_1 i r_2 będą dwoma rankingami (permutacjami) n obiektów. Wtedy $\tau(r_1, r_2)$ jest równe najmniejszej liczbie zamian dwóch sąsiednich obiektów w rankingu r_1 koniecznych do uzyskania rankingu r_2 .

Poniższa definicja miary D wyrażającej odległość dwóch macierzy ma postać standardowej normy macierzy L_1 .

Definicja: Niech $A = (a_{ij})$ i $A' = (a'_{ij})$ będą odpowiednio oryginalną macierzą i macierzą przekształconą AIR. Następnie odległość D między A i A' jest zdefiniowana w następujący sposób (2.2)

$$D(A, A') = \|A - A'\| = \sum_{i=1}^n \sum_{j=1}^n |a_{ij} - a'_{ij}|. \quad (2.2)$$

Można zauważyć, że D ściśle wiąże się z TJD Abela i in. [1], ponieważ $\frac{D}{n^2} = TJD$.

Definicja: Niech $A=(a_{ij})$ i $A'=(a'_{ij})$ będą odpowiednio oryginalną macierzą i macierzą przekształconą AIR. Wtedy $N = |a_{ij}; a_{ij} \neq a'_{ij}, \forall i, j|$ oznacza liczbę elementów macierzy, które zostały zmienione (dopasowane) w procesie redukcji niespójności.

Im niższe wartości d , τ Kendalla, D i N , tym lepiej AIR zachowuje oryginalne preferencje.

2.3. Iteracyjne algorytmy redukcji niespójności

W tej sekcji artykułu [51] przedstawiono wybrane iteracyjne algorytmy redukcji niespójności zastosowane w części numerycznej rozdziału (w kolejności alfabetycznej). W literaturze zaproponowano kilka innych algorytmów iteracyjnych, jednak wymagają one pomocy decydenta w każdej iteracji, patrz np. [13], [19] lub [20], co czyni je nieodpowiednimi do obszernych symulacji.

Używano współczynnika spójności CR (i wartości progowej $CR = 0,10$) do oceny niespójności macierzy PC podczas procedur redukcji niespójności zgodnie z wcześniejszymi badaniami, patrz [1], [11], [13], [25], [36], [39], [40] lub [47], chociaż inne wskaźniki niespójności, takie jak wskaźnik niespójności Koczkodaja KI , mogą być również zasadniczo stosowane. Opisy iteracyjnych algorytmów redukcji niespójności przedstawiono w rozdziale pierwszym.

2.4. Symulacje Monte Carlo

Na początku eksperymentu wygenerowano losowo dużą liczbę (ponad 10 000 przypadków) macierzy PC rzędu $n = 4, n = 8$ z $CR \geq 0.10$, gdzie wartości macierzy zostały wylosowane z podstawowej skali Saaty'ego w celu zbadania rozkładu współczynnika spójności CR . Stwierdzono, że w przypadku $n = 4$, moda znajdowała się w przedziale $(0.20, 0.30)$, mediana wynosiła 0.86, a średnia arytmetyczna (zgodnie z definicją CR) była równa 1. W przypadku $n = 8$, średnia arytmetyczna, moda i mediana były bliskie 1.

Ponieważ porównania parami są zwykle dostarczane przez osobę posiadającą odpowiednią wiedzę (zwaną "ekspertem"), można bezpiecznie założyć, że preferencje eksperta będą mniej niespójne niż preferencje losowe. Dlatego w badaniu ustalono górną granicę niespójności na poziomie $CR = 0.80$. Aby odróżnić matryce mniej i bardziej niespójne, zastosowano wartość $CR = 0.30$ dla $n = 4$.

Po tej fazie przygotowawczej wygenerowano losowo duże próbki (ponad 10 000 przypadków) macierzy PC rzędu $n = 4$ i $n = 8$ dla rzeczywistych symulacji za pomocą tej

samej procedury. Następnie odfiltrowano macierze PC z $0.10 < CR < 0.80$. Następnie każda macierz PC została użyta jako dane wejściowe dla wszystkich wybranych algorytmów redukcji niespójności. Algorytmy zatrzymywały się, gdy współczynnik spójności CR zmodyfikowanej macierzy spadał poniżej progu 0,10. Dane wyjściowe składały się z końcowej zmodyfikowanej macierzy, maksymalnej wartości własnej i wektora priorytetów w . Na koniec, dla każdej zmodyfikowanej macierzy zostały ocenione miary zachowania preferencji: d , D , N i τ .

Wyniki symulacji - wydajność AIR w odniesieniu do rozmiaru macierzy i początkowej niespójności macierzy - przedstawiono w Tabelach 2.1 - 2.4. Liczby w podpisach tabel wyrażają, ile macierzy z oryginalnego zestawu 10 000 wygenerowanych macierzy PC znalazło się w wyżej wymienionych przedziałach CR (według Lercha i Mudforda [29] liczby te są wystarczające).

Tabela 2.1 Wydajność AIR, średnie wartości dla $n = 4$ i początkowe $0,10 \leq CR < 0,30$, 630 macierzy (najlepsze wartości są pogrubione) – źródło [51]

Algorithm	d	D	τ	k	N
Cao et al. ($\gamma = 0.98$) [11]	0.582	4.74	0.024	16.02	12
Cao et al. II ($\gamma = 0.50$) [11]	0.568	7.94	0.027	1	12
Kou et al. [25]	5.86	10.97	0.57	1	2
Mazurek et al. [35]	2.33	3.94	0.132	4.69	3.29
Szybowski [44]	3.22	5.34	0.185	1.24	2.47
Xu and Wei ($\lambda = 0.5$) [47]	0.212	7.96	0.008	1	12
Xu and Wei II ($\lambda = 0.9$) [47]	0.217	5.20	0.008	3.60	12

Tabela 2.2 Wydajność AIR, średnie wartości dla $n = 4$ i początkowe $0.30 \leq CR < 0.80$, 1193 macierze (najlepsze wartości zostały pogrubione) – źródło [51]

Algorithm	d	D	τ	k	N
Cao et al. [11]	2.132	11.67	0.191	35.50	12
Cao et al. II [11]	1.552	14.44	0.147	1.64	12
Kou et al. [25]	-	-	-	-	-
Mazurek et al. [35]	5.341	8.74	0.595	8.89	4.62
Szybowski [44]	5.827	9.97	0.638	2.19	3.84
Xu and Wei [47]	0.567	14.47	0.039	1.65	12
Xu and Wei II [47]	0.723	12.12	0.049	7.60	12

Tabela 2.3 Wydajność AIR, średnie wartości dla $n = 8$ i początkowe $0.10 < CR < 0.30$, 491 macierzy (najlepsze wartości są pogrubione) – źródło [51]

Algorithm	d	D	tau	k	N
Cao et al. [11]	0.818	37.27	0.727	21.3	56
Cao et al. II [11]	0.629	47.83	0.529	1	56
Mazurek et al. [35]	1.780	29.25	1.610	23.4	17.13
Szybowski [44]	2.357	31.18	2.218	4.8	8.78
Xu and Wei [47]	0.241	48.38	0.188	1	56
Xu and Wei II [47]	0.289	39.73	0.216	4.73	56

Tabela 2.4 Wydajność AIR, średnie wartości dla $n = 8$ i początkowe $0.30 \leq CR < 0.80$, 4082 macierze (najlepsze wartości zostały pogrubione) – źródło [51]

Algorithm	d	D	tau	k	N
Cao et al. [11]	1.980	62.47	2.764	36.30	56
Cao et al. II [11]	1.426	76.03	2.049	1.77	56
Mazurek et al. [35]	3.198	49.52	3.762	41.92	24.94
Szybowski [44]	3.659	49.18	4.048	8.18	16.18
Xu and Wei [47]	0.581	76.79	0.665	1.75	56
Xu and Wei II [47]	0.727	65.03	0.798	7.70	56

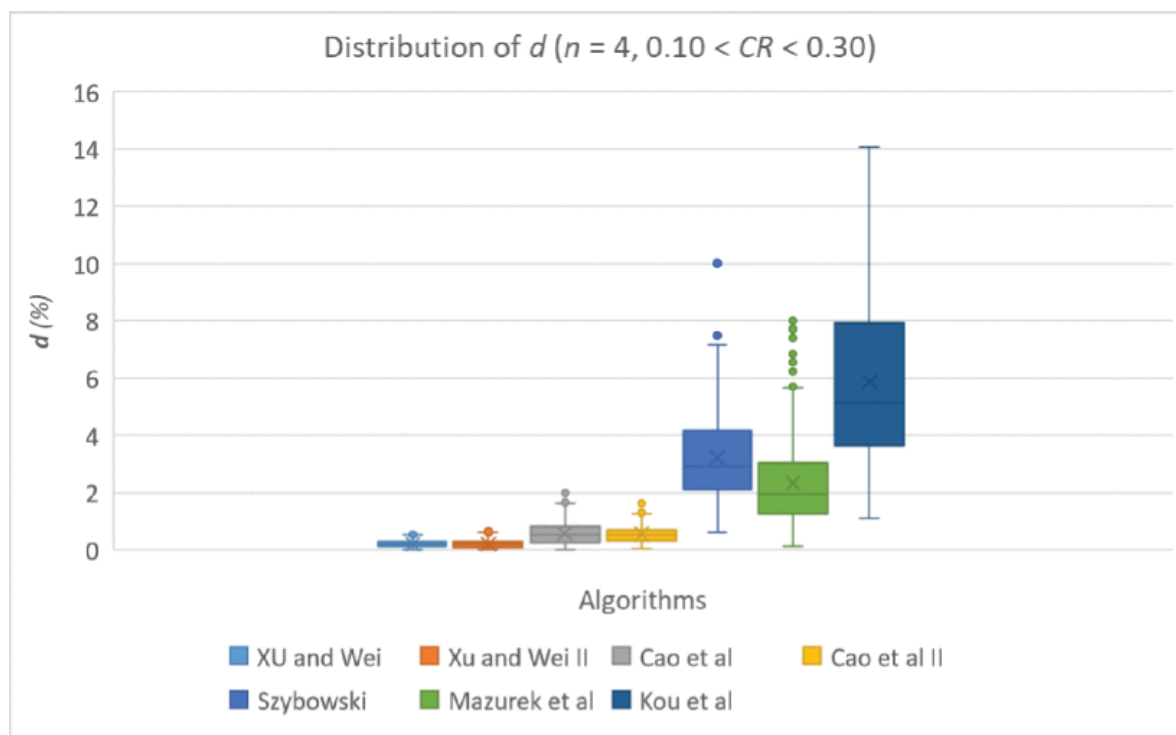
Algorytm Kou i in. nie znalazł wystarczająco spójnej macierzy w 3% przypadków dla $n = 4$ i $0.10 < CR < 0.30$ oraz w 45% przypadków dla $n = 4$ i $0.30 \leq CR < 0.80$. W przypadku $n = 8$ algorytm zawiódł w ponad 60% przypadków. Dlatego podano jego wyniki tylko dla $n = 4$ i $0.10 < CR < 0.30$ (więcej o algorytmie Kou i in. w następnym podrozdziale).

2.5. Dyskusja na temat wyników

Podczas symulacji stwierdzono, że algorytm Kou i in. [25] nie działał poprawnie, gdy jego główna procedura była wykonywana więcej niż jeden raz. W takim przypadku algorytm odbiegał od normy. W konsekwencji algorytm zawiódł w ponad 60% przypadków dla $n = 8$, co czyni porównanie z innymi algorytmami niewykonalnym. Warto zauważyć, że w oryginalnej pracy [25] brakuje dowodu na zbieżność algorytmu. Prawdopodobną przyczyną rozbieżności jest to, że po modyfikacji najbardziej niespójnego elementu w pierwszej iteracji, macierz C nie jest aktualizowana. Aby rozwiązać ten problem, zasugerowano aktualizację macierzy C w każdej iteracji.

Jak można się było spodziewać, redukcja niespójności większych macierzy ($n = 8$) i/lub bardziej niespójnych macierzy ($0.30 \leq CR < 0.80$) doprowadziła do większych wartości, a tym samym niższego zachowania początkowych preferencji, zmiennych d , D i τ , patrz także Rysunki 2.1 - 2.8.

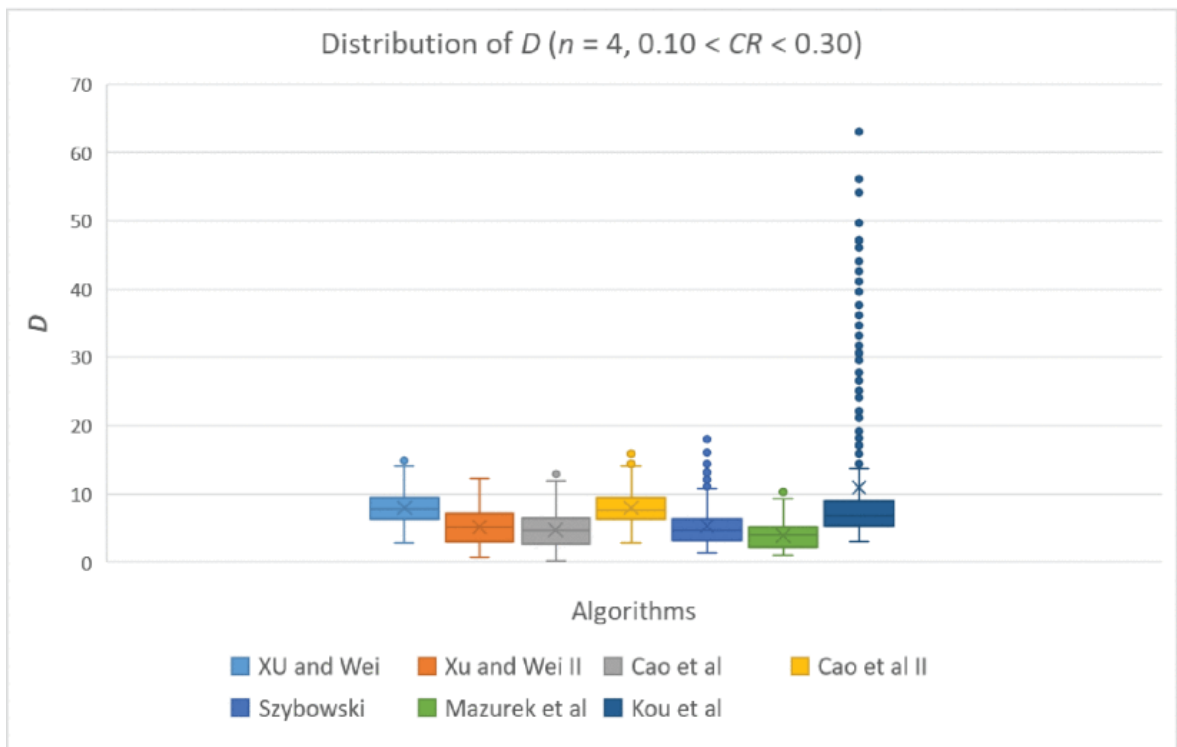
Macierze te wymagały od algorytmów większej ilości iteracji, a w przypadku algorytmów Szybowski i Mazurka i in. także większej liczby dopasowywanych elementów (algorytmy Xu i Wei, Cao i in. oraz Kou i in. są zaprojektowane tak, aby modyfikować wszystkie elementy niediagonalne). Z obliczeniowego punktu widzenia algorytmy Cao i in. oraz Xu i Wei wykazywały najszybszą zbieżność.



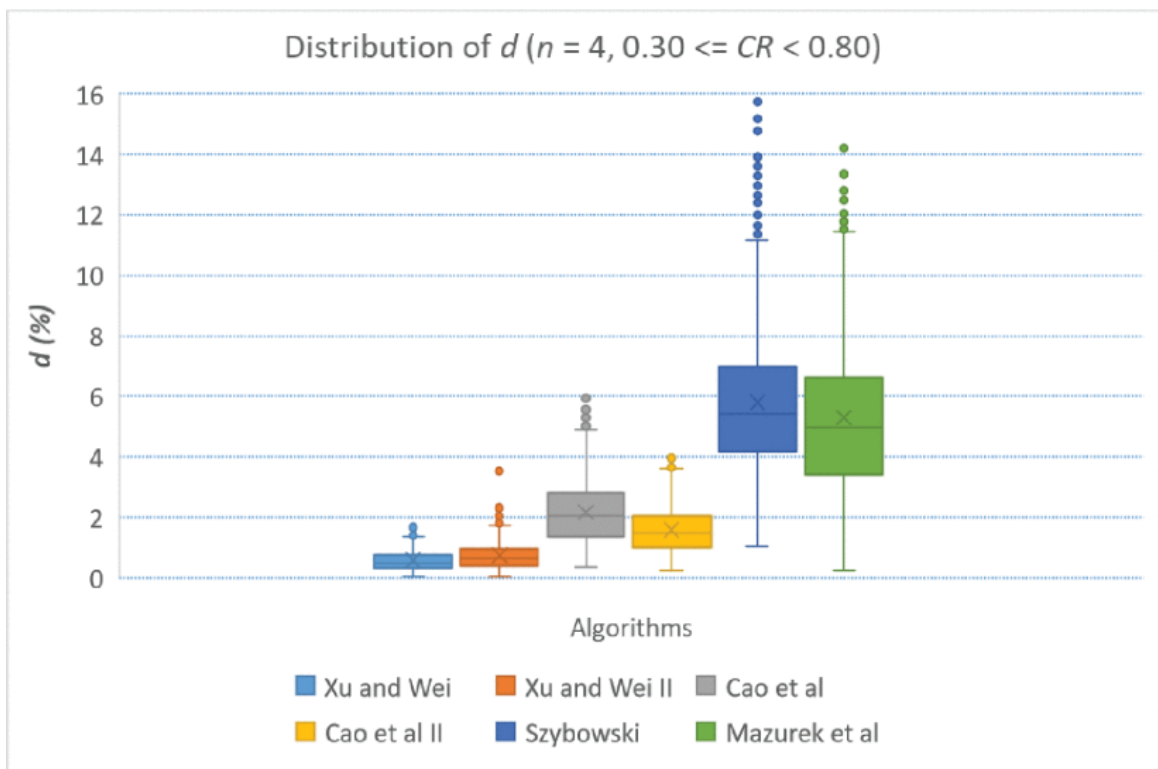
Rys. 2.1. Rozkład d (%), $n = 4$, $0.10 < CR < 0.30$.

Jeśli chodzi o wpływ wartości parametrów γ oraz λ - odpowiednio w algorytmach Cao et al. oraz Xu i Wei, w przypadku algorytmu Cao et al. i $\gamma = 0.98$ modyfikacje macierzy były bardziej wyrafinowane, co wymagało większej liczby kroków do osiągnięcia niespójności progowej niż w przypadku $\gamma = 0.50$.

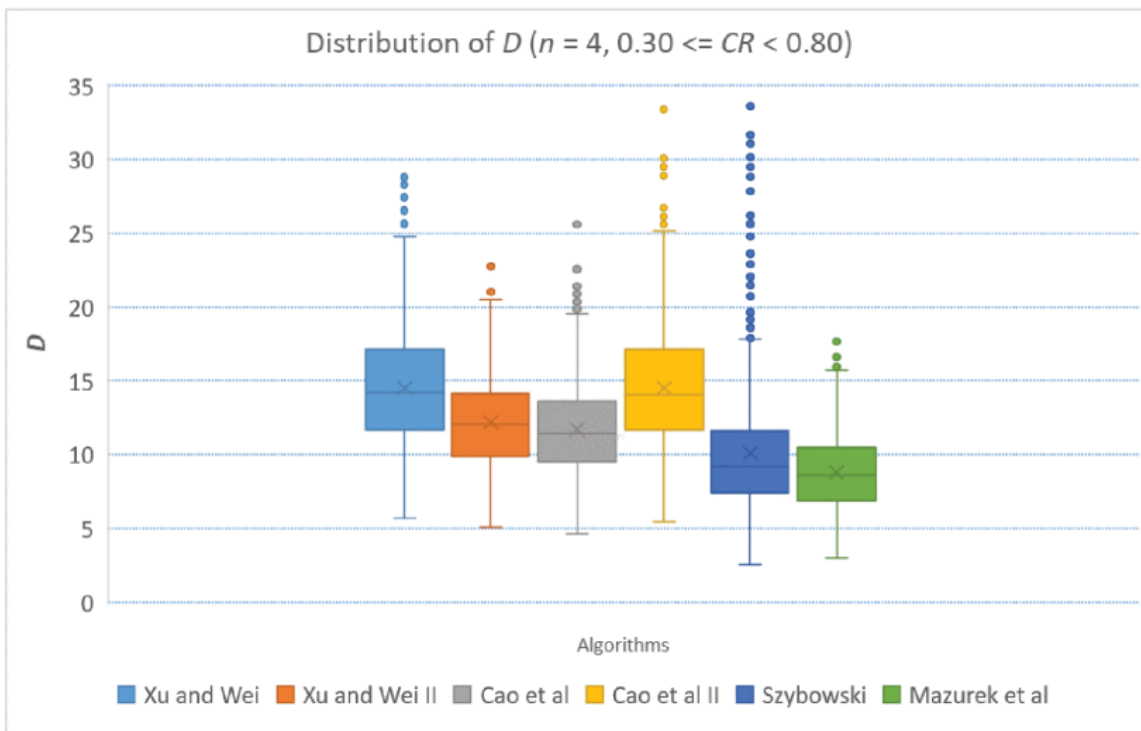
Ta sama obserwacja dotyczy algorytmu Xu i Wei, przypadek z $\lambda = 0,90$ przebiegał w mniejszych krokach w większej liczbie iteracji.



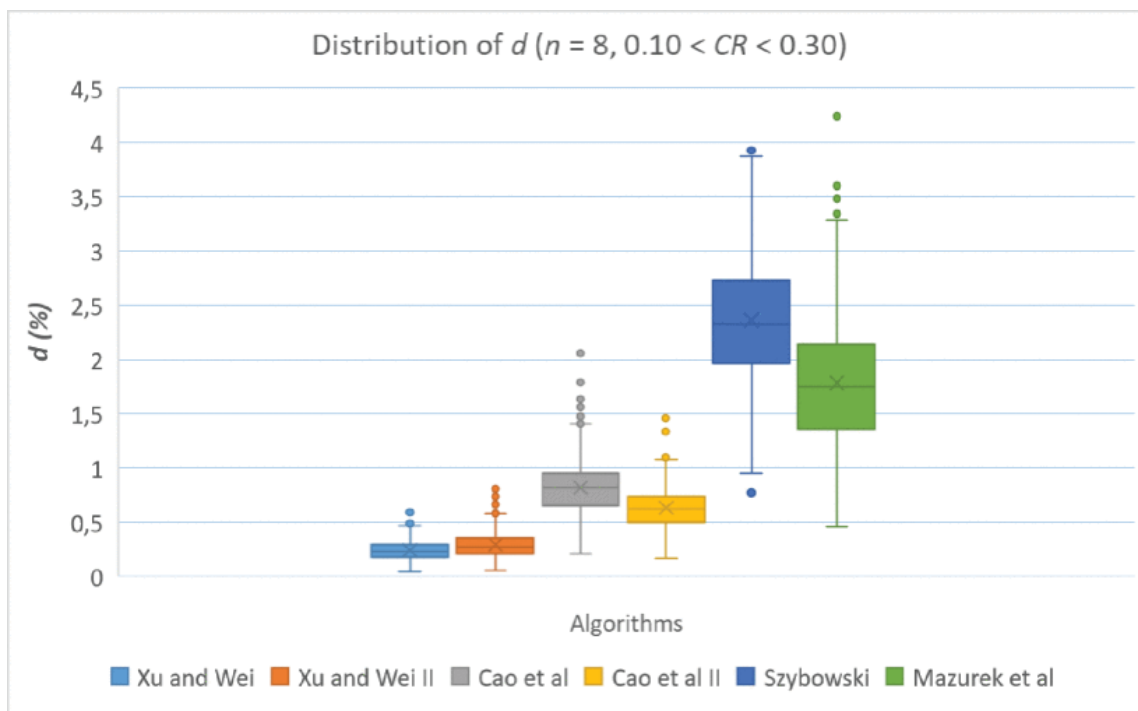
Rys. 2.2. Rozkład D , $n = 4$, $0.10 < CR < 0.30$.



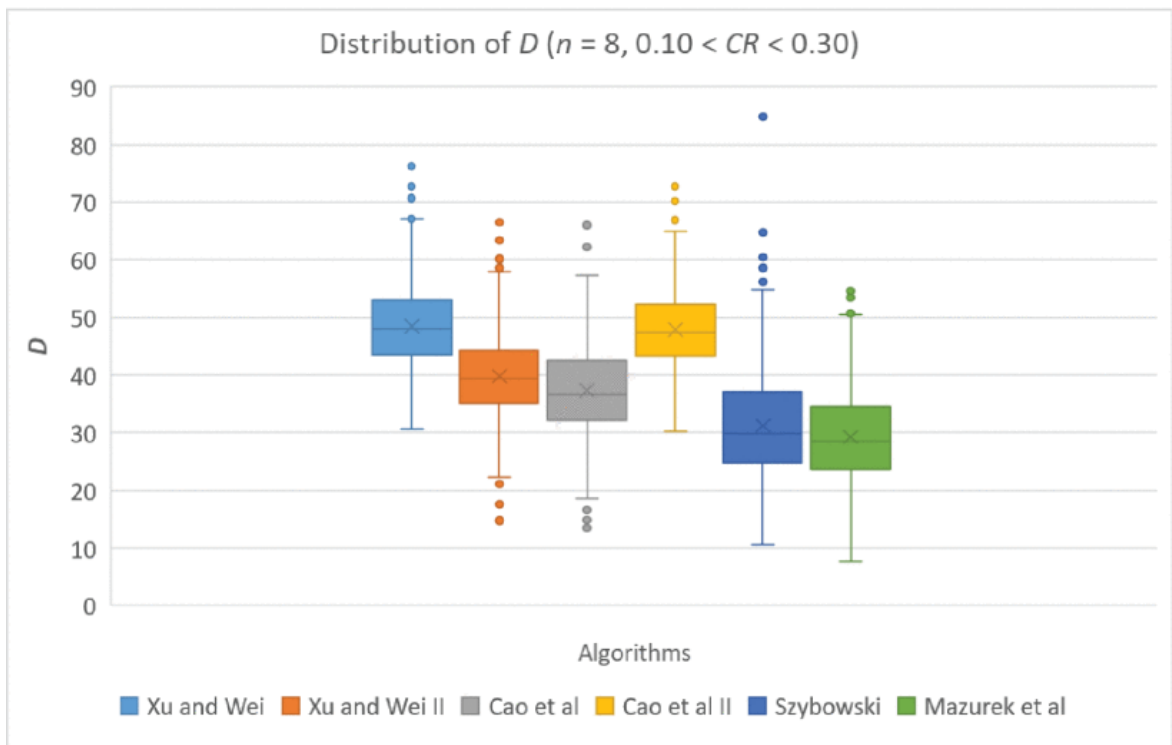
Rys. 2.3. Rozkład $d(\%)$, $n = 4$, $0.30 \leq CR < 0.80$.



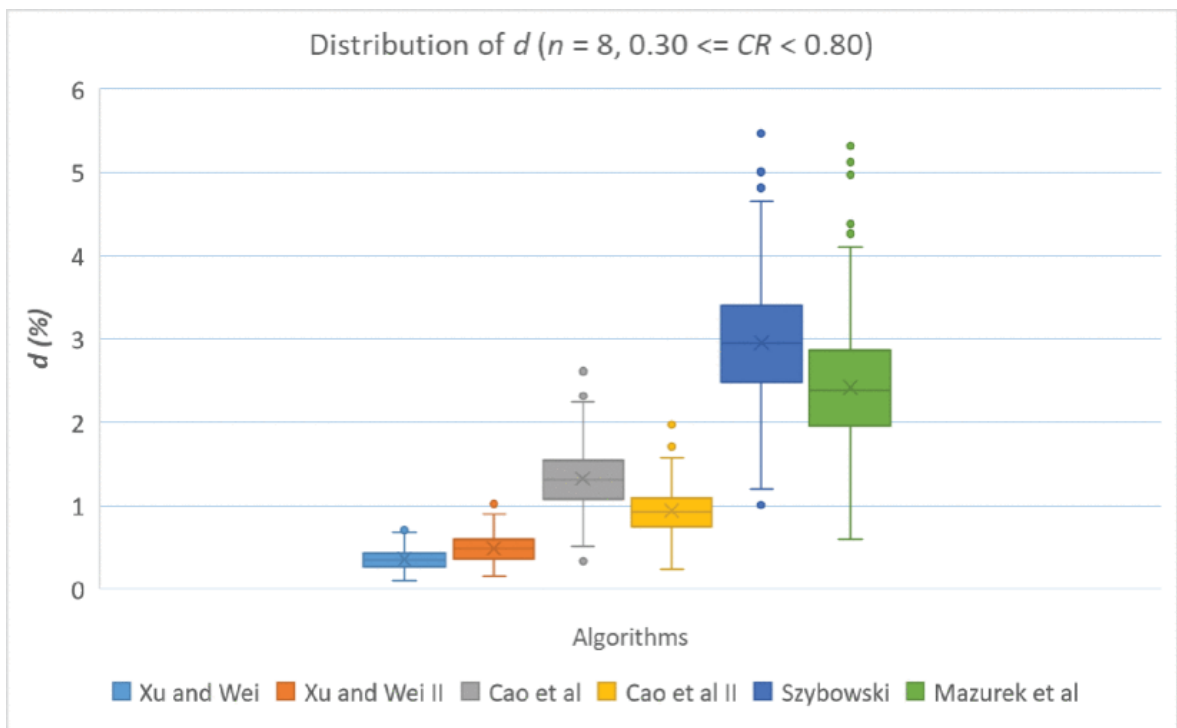
Rys. 2.4. Rozkład D , $n = 4$, $0.30 \leq CR < 0.80$.



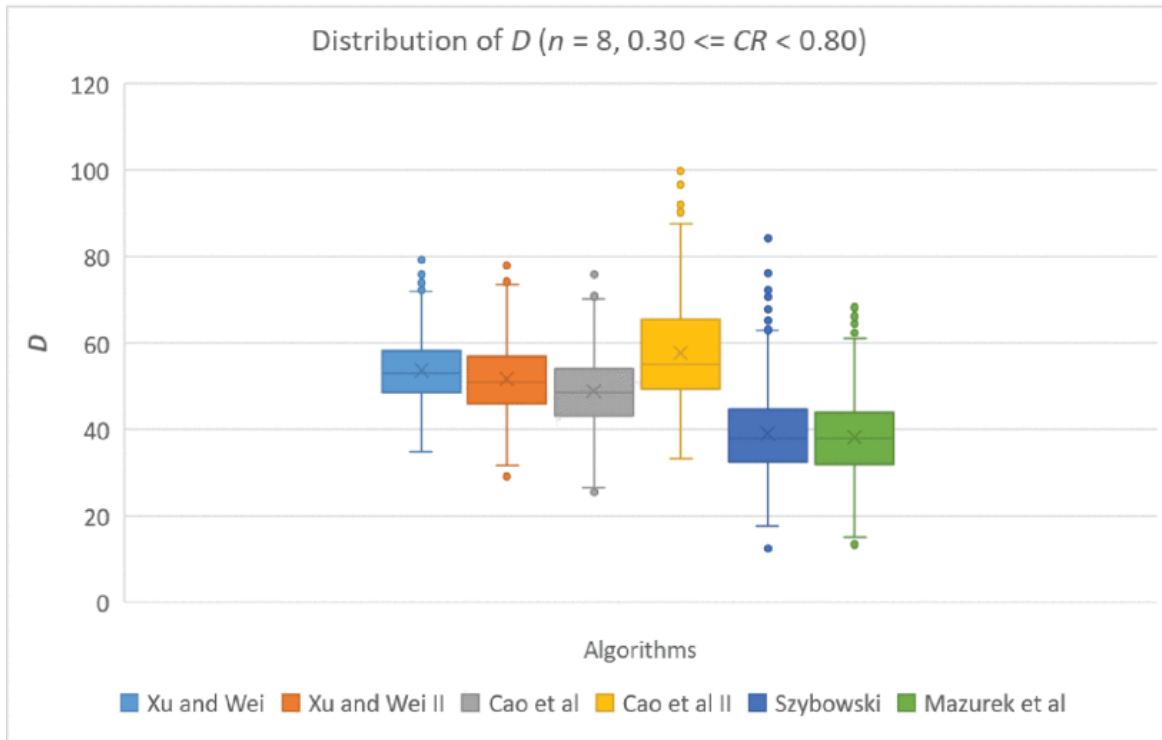
Rys. 2.5. Rozkład $d(\%)$, $n = 8$, $0.10 < CR < 0.30$.



Rys. 2.6. Rozkład D , $n = 8$, $0.10 < CR < 0.30$.



Rys. 2.7. Rozkład $d(\%)$, $n = 8$, $0.30 \leq CR < 0.80$.



Rys. 2.8. Rozkład D , $n = 8$, $0.30 \leq CR < 0.80$.

Test ANOVA (analiza wariancji) potwierdził, że różnice w średnich wartościach d , D i τ dla wszystkich algorytmów (z wyjątkiem algorytmu Kou et al.) były statystycznie istotne przynajmniej na poziomie $p = 10^{-10}$.

Jeśli chodzi o podobieństwa algorytmów, algorytmy Xu i Wei oraz Cao i in. były silnie skorelowane w zmiennej D , z maksymalnym współczynnikiem korelacji Pearsona $\rho = 0,973$ dla $n = 4$ i $0,10 < CR < 0,30$.

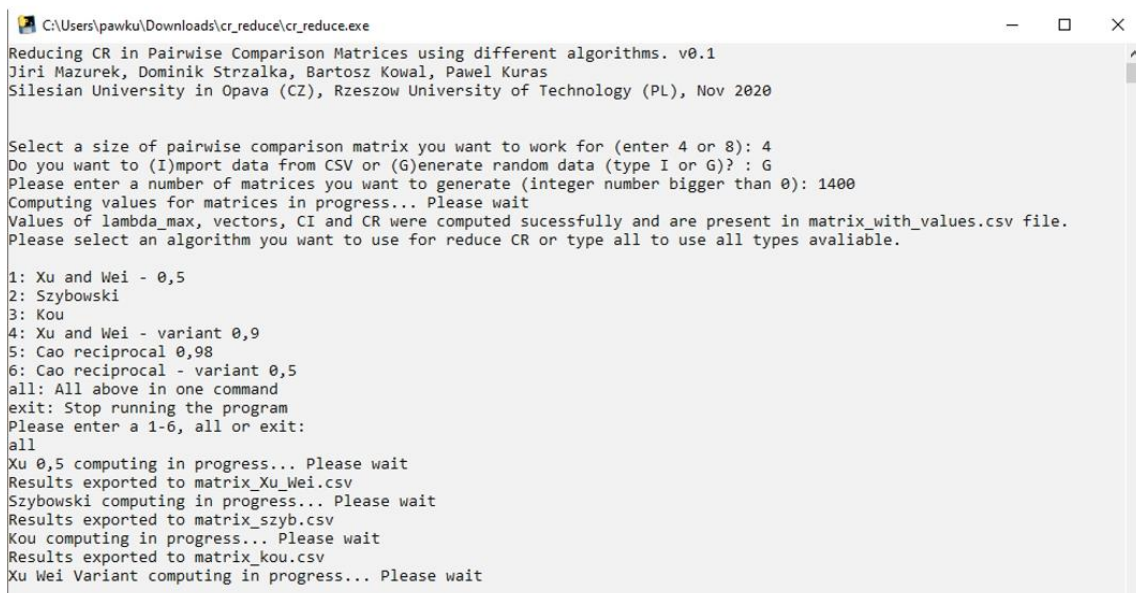
Z drugiej strony, algorytm Szybowskiego korelował najbardziej z algorytmem Mazurka i in., szczególnie dla zmiennej D , z maksymalnym współczynnikiem korelacji Pearsona $\rho = 0,909$ dla $n = 8$ i $0,30 \leq CR < 0,80$.

Wyniki pracy algorytmów były przetwarzane z dokładnością do czterech miejsc po przecinku. Większość programowania została wykonana w Pythonie, reszta w R i C#. Wszystkie dane, algorytmy i raport techniczny można znaleźć w bezpłatnym repozytorium GitHub [50]. W tym repozytorium dostępna jest także aplikacja PCM_CR, opisana w kolejnym podrozdziale.

2.6. Narzędzie PCM_CR

PCM_CR to prosta aplikacja CLI (działająca w trybie tekstowym), która prowadzi użytkownika w procesie redukcji niespójności. Jest ona ograniczona tylko do macierzy 4×4 i 8×8 , gdyż takie były założenia badawcze opisane w pracy [51] - tzw. "małe macierze" i "duże macierze". Program używa bibliotekę reduce.py (opisaną w rozdziale nr. 3) do implementacji algorytmów Xu i Wei, Szybowskiego i Cao w różnych wariantach, a także do generowania macierzy i importowania ich z pliku CSV.

Narzędzie to zawiera również algorytm opisany przez Cao i in. [27], ale algorytm ten nie jest obecny w bibliotece reduce.py, ponieważ badania wykazały, że jest on niestabilny i problematyczny.



```
C:\Users\pawku\Downloads\cr_reduce\cr_reduce.exe
Reducing CR in Pairwise Comparison Matrices using different algorithms. v0.1
Jiri Mazurek, Dominik Strzalka, Bartosz Kowal, Pawel Kuras
Silesian University in Opava (CZ), Rzeszow University of Technology (PL), Nov 2020

Select a size of pairwise comparison matrix you want to work for (enter 4 or 8): 4
Do you want to (I)mport data from CSV or (G)enerate random data (type I or G)? : G
Please enter a number of matrices you want to generate (integer number bigger than 0): 1400
Computing values for matrices in progress... Please wait
Values of lambda_max, vectors, CI and CR were computed sucessfully and are present in matrix_with_values.csv file.
Please select an algorithm you want to use for reduce CR or type all to use all types available.

1: Xu and Wei - 0,5
2: Szybowski
3: Kou
4: Xu and Wei - variant 0,9
5: Cao reciprocal 0,98
6: Cao reciprocal - variant 0,5
all: All above in one command
exit: Stop running the program
Please enter a 1-6, all or exit:
all
Xu 0,5 computing in progress... Please wait
Results exported to matrix_Xu_Wei.csv
Szybowski computing in progress... Please wait
Results exported to matrix_szyb.csv
Kou computing in progress... Please wait
Results exported to matrix_kou.csv
Xu Wei Variant computing in progress... Please wait
```

Rys. 2.9. Narzędzie PCM_CR

Jak widać na rysunku 2.9, użytkownik może wybrać, który algorytm zostanie użyty, lub alternatywnie użyć wszystkich algorytmów do redukcji CR. Wybierając ostatnią opcję, w wyniku działania programu, użytkownik otrzyma szereg plików CSV (matrix_kou.csv, matrix_szyb.csv, matrix_Xu_Wei.csv, matrix_Cao.csv) ze zredukowanymi macierzami porównań parami.

2.7. Wnioski

Celem rozdziału przygotowanego na bazie pracy [51] było przeprowadzenie symulacji Monte Carlo dla porównania wybranych iteracyjnych algorytmów redukcji niespójności pod względem zachowania oryginalnych preferencji w ramach porównywania

parami. Uzyskane wyniki wskazują, że żaden algorytm nie przewyższał wszystkich innych algorytmów pod względem każdej miary zachowania oryginalnych preferencji. Algorytm Xu i Wei był najlepszym algorytmem pod względem zachowania wektora priorytetów i rankingu obiektów. Algorytm Mazurek et al. był najlepszym algorytmem pod względem zachowania preferencji wyrażonych przez wpisy oryginalnej macierzy PC, podczas gdy algorytm Szybowskiego zachował najwięcej wpisów macierzy niezmiennych podczas redukcji niespójności. Dlatego wybór najbardziej odpowiedniego AIR zależy od potrzeb decydenta. Algorytm Kou et al. okazał się rozbieżny w większości przypadków, dlatego nie można go było porównać z innymi algorytmami.

Dalsze badania w tej materii mogą skupić się na porównaniu nieiteracyjnych algorytmów redukcji niespójności lub mogą mieć na celu ramy interwałowych, rozmytych lub rozmytych niezdecydowanych porównań parami.

3. REDUCE.py: moduł Pythona do redukcji niespójności w macierzach porównań parami

Słowa kluczowe rozdziału:

niespójność, wspomaganie podejmowania decyzji, porównania parami, indeks niespójności, redukcja niespójności, moduł Python

Abstrakt rozdziału (za oryginalną publikacją):

Ten rozdział rozprawy doktorskiej prezentuje REDUCE.py, moduł Pythona zaprojektowany do minimalizacji niespójności w wielokrotnych porównaniach parami (PC), stanowiących podstawową technikę w podejmowaniu decyzji wielokryterialnych (MCDM). Porównania parami są szeroko stosowane w różnych dziedzinach, w tym w naukach inżynierskich i metodach symulacji numerycznych, do porównywania różnych opcji na podstawie zestawu kryteriów. Niemniej jednak, ludzkie błędy w percepcji i ocenie często prowadzą do niespójności w macierzach porównań parami (PCM). REDUCE.py rozwiązuje ten problem poprzez implementację kilku algorytmów, które identyfikują i poprawiają nieścisłe dane w PCM, redukując tym samym współczynnik niespójności. Algorytmy te nie wymagają interwencji ekspertów, co czyni REDUCE.py cennym narzędziem zarówno dla badań naukowych, jak i małych i średnich przedsiębiorstw, które mogą nie mieć dostępu do drogiego oprogramowania komercyjnego czy specjalistów od podejmowania decyzji. Główną funkcjonalnością modułu jest inkorporowanie iteracyjnych algorytmów redukcji niespójności. Biblioteka REDUCE.py, napisana w Pythonie i wykorzystująca pomocnicze biblioteki, takie jak NumPy, SciPy i SymPy, oferuje 21 funkcji podzielonych na pomocniki wejścia danych, algorytmy redukcji współczynnika niespójności (CR), indeksy PCM i funkcje pomocnicze. Testy wydajności wskazują, że biblioteka może skutecznie obsługiwać macierze o różnych rozmiarach, szczególnie od 3x3 do 10x10, a jej użycie znacznie przyspiesza proces w porównaniu do arkuszy kalkulacyjnych, zwłaszcza przy obsłudze dużych ilości macierzy. Biblioteka ta była już używana w kilku pracach badawczych i narzędziach aplikacyjnych, a jej dostępność jako darmowego zasobu otwiera możliwości dla małych i średnich przedsiębiorstw do wykorzystania metod podejmowania decyzji wielokryterialnych. Obecnie nie ma dostępnych publicznie bibliotek dla tego rozwiązania. Proponowany moduł może przyczynić się do lepszego procesu podejmowania decyzji w porównaniach parami, nie tylko dla kręgu naukowców, ale także dla małych i średnich przedsiębiorstw, które zazwyczaj nie mogą sobie pozwolić na drogie oprogramowanie komercyjne i nie zatrudniają na pełny etat ekspertów od podejmowania decyzji, polegając na doświadczeniu swoich pracowników i darmowych zasobach online. Powinno to również przyczynić się do przejścia do przemysłu 4.0 oraz zaawansowanych badań w dziedzinach takich jak logika rozmyta, programowanie preferencji i konstruktywne przybliżenia spójne.

3.1. Wstęp

Dotychczas brak było darmowych i publicznie dostępnych bibliotek implementujących rozwiązania prezentowane w poprzednich rozdziałach tej rozprawy doktorskiej (zarówno w Pythonie, jak i w innych językach programowania, na dzień publikacji niniejszego rozwiązania), więc naukowcy zajmujący się tematem porównań parami musieli głównie polegać na pracy w arkuszach kalkulacyjnych, takich jak Microsoft Excel czy LibreOffice Calc [69-71]. W niniejszym rozdziale rozprawy doktorskiej przedstawiony zostanie pakiet Pythona do redukcji niespójności macierzy PC zgodnie z kilkoma znanymi i prezentowanymi w literaturze algorytmami. Program może znaleźć wiele użytecznych zastosowań wśród naukowców i entuzjastów, którzy pracują z systemami decyzyjnymi wielokryterialnymi i ciągle tworzą nowe wyzwania w tej dziedzinie.

3.1.1. Uzasadnienie wyboru algorytmów do konstrukcji modułu

Jak zostało to opisane w rozdziale 1, metody porównań parami to rodzaje metod podejmowania decyzji, które są bezpośrednio związane z metodą wyznaczania priorytetów, która daje wektor priorytetów $w = (w_1, \dots, w_n)$ (znany również jako wektor wag wszystkich n porównywanych obiektów) z macierzy porównań parami. Zazwyczaj wektor priorytetów w jest normalizowany (3.1):

$$\sum_{i=1}^n w_i = 1. \quad (3.1)$$

Wartości w_1, \dots, w_n porównań są używane do rankingu najlepszych i najgorszych elementów. Jak wspomniano w rozdziale 1, powinien być poruszony specjalny problem niespójności PC. Indeks spójności Saaty'ego CI [1] dla macierzy PC $n \times n$ A jest dany jako (3.2), a współczynnik spójności CR jest zdefiniowany jako (3.3):

$$CI(A) = \frac{\lambda_{max} - n}{n - 1}, \quad (3.2)$$

$$CR(A) = \frac{CI(A)}{RI(n)}. \quad (3.3)$$

We wzorze (3.3) $RI(n)$ jest rozumiany jako losowy indeks spójności wyrażony przez średnią arytmetyczną losowo generowanych macierzy PC danego rzędu ze skalą Saaty'ego zależną od n , a λ_{max} jest największą (dodatnią) wartością własną A .

Jednak istnieją również inne definicje indeksów niespójności - dzięki analizie badań przeprowadzonych w ramach książki autorstwa J. Mazurka [2], zdecydowano również o implementacji ważnych indeksów: Relative Error Index (REI) [3], Golden-Wang Index (GWI) [4], Koczkodaj Index [5], Obata Index (OI) [6], Peláez-Lamata Index (PLI) [7],

indeksy GCI i TGCI [8,9] oraz Harmonic Consistency Index (HCI) [10] - definicje których są podane w publikacjach ich autorów, a ich implementacja może być znaleziona w bibliotece opisanej niżej w niniejszym rozdziale rozprawy doktorskiej (z wyjątkiem indeksu Obata, który wykazywał problemy ze stabilnością podczas działania).

Ponieważ główną przyczyną niespójności są błędy w percepcji rzeczywistości i ocenach przez ludzi (nie jest tak oczywiste, gdzie mogłyby one wystąpić) istnieją metody i algorytmy, które są w stanie znaleźć i poprawić błędne dane w macierzy A , aby zminimalizować współczynnik niespójności.

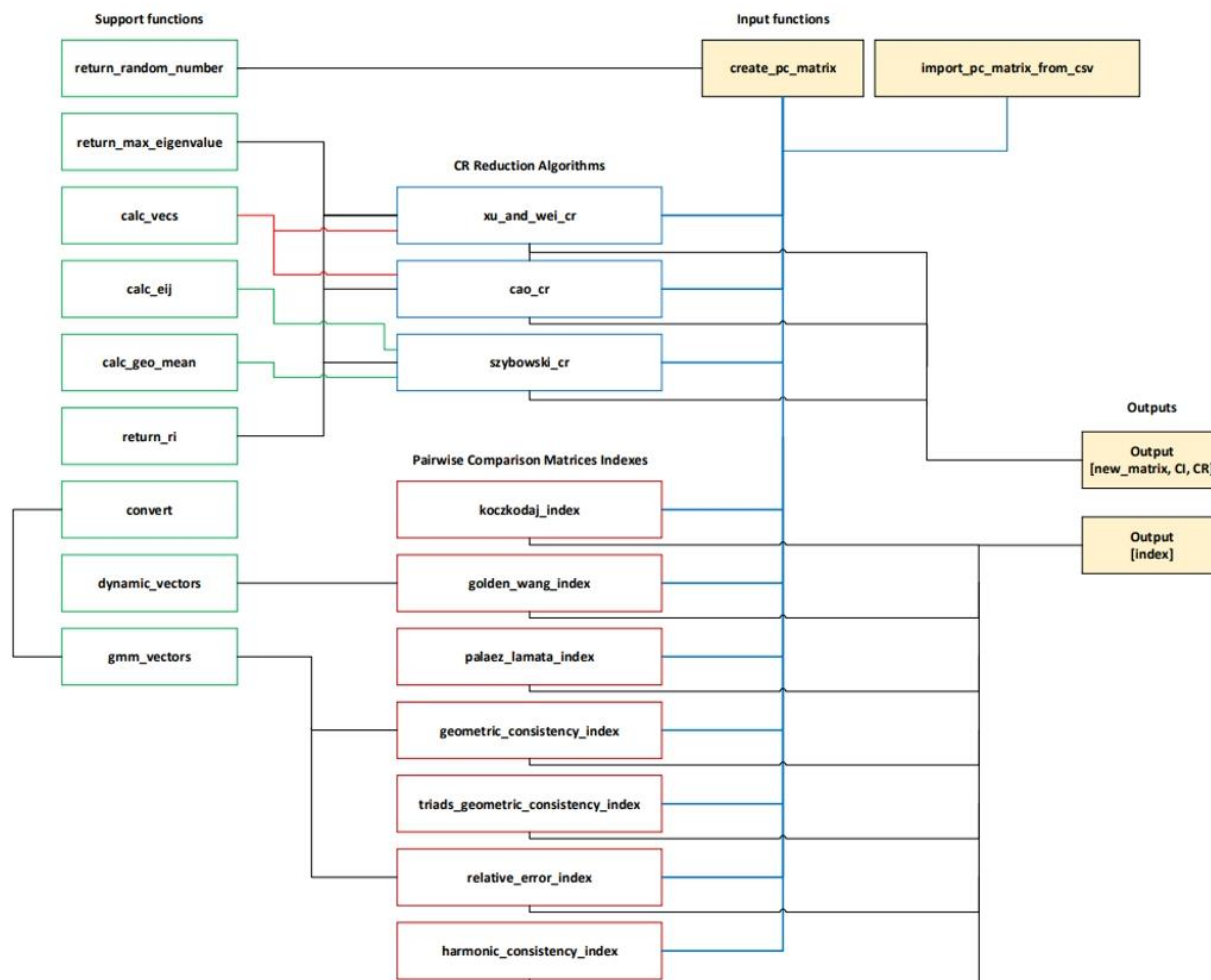
Chociaż istnieje wiele algorytmów do tego celu, zasadniczo (jak wspomniano w rozdziale pierwszym) istnieją dwie grupy algorytmów redukcji niespójności macierzy porównań parami - nieiteracyjne i iteracyjne [2]. Na podstawie wniosków z pracy badawczej zespołu Mazurka i in. [11] (opisanej w drugim rozdziale niniejszej rozprawy doktorskiej), moduł proponowany w tym rozdziale zawiera algorytmy z grupy algorytmów iteracyjnych. Są to algorytmy Cao i in. [12], Szybowskiego [13] oraz Xu i Wei [14]. Chociaż w pracy [11] występują co najmniej dwa inne znane algorytmy, algorytm Kou i in. [15] okazał się niestabilny i trudny do efektywnego zastosowania, podobnie jak algorytm Mazurka i in. [16], który nie zbiega do zera niespójności w każdym przypadku - dlatego te algorytmy zostały pominięte podczas tworzenia biblioteki.

3.2. Architektura i struktura oprogramowania

Biblioteka `reduce.py` została napisana w języku Python, przy użyciu bibliotek pomocniczych NumPy [17], SciPy [18] i SymPy [19]. Repozytorium biblioteki znajduje się pod adresem [20], jest opisane w kolejnym podrozdziale.

Architektura oprogramowania została przedstawiona na Rys. 3.1. Składa się ona w obecnej wersji z 21 funkcji. Wszystkie funkcje są dostępne dla użytkownika końcowego. Mogą być podzielone na cztery kategorie: pomocnicze funkcje wprowadzania danych, algorytmy redukcji CR (współczynnika spójności), indeksy PCM (macierzy porównań parami) oraz funkcje pomocnicze. Pierwsza kategoria zawiera dwie funkcje pomocnicze, które pomagają zarówno: tworzyć losową macierz porównań parami, jak i wykonywać jej odczyt z odpowiednio sformatowanego pliku CSV. Druga kategoria zawiera implementacje algorytmów redukcji niespójności macierzy porównań, opisane w dalszej części rozprawy podobnie jak i ich implementacja. Na wejściu algorytmu jest podawana macierz porównań wraz z wymaganymi parametrami pomocniczymi, natomiast na wyjściu otrzymuje się macierz ze zredukowanym współczynnikiem niespójności (CR). Trzecia kategoria zawiera indeksy używane w pracy badawczej, które można obliczyć, podając macierz porównań

parami bezpośrednio na wejściu, a na wyjściu otrzymując dany indeks. Ostatnia kategoria to funkcje pomocnicze, używane przez główne funkcje do obliczania różnych parametrów niezbędnych do działania głównych algorytmów. Chociaż są to funkcje dodane do działania algorytmów w drugiej i trzeciej kategorii, uznano, że będą one publicznie dostępne dla wszystkich użytkowników, ze względu na ich potencjalną użyteczność.



Rys. 3.1. Struktura biblioteki REDUCE.py

3.2.1. Repozytorium w serwisie GitHub

Repozytorium "reduce" na platformie GitHub [20] zostało stworzone jako przykład współczesnego zarządzania projektem programistycznym, zorganizowanego w sposób, który odzwierciedla najlepsze znane obecnie praktyki w dziedzinie inżynierii oprogramowania i zachęca innych twórców do wspólnego rozwijania projektu. Na chwilę obecną (01.09.2023) składa się z 29 tzw. commitów (zmian wprowadzonych przez autorów) i zawiera pliki napisane głównie w języku Python, a także pliki z wykorzystaniem składni Makefile i Dockerfile [21-23]. Kluczowy moduł "reduce.py" służy do obliczeń związanych z macierzami porównań parami, z wykorzystaniem różnych algorytmów i indeksów.

Repozytorium wykorzystuje szereg nowoczesnych narzędzi i technologii, takich jak Poetry do zarządzania zależnościami, Black i isort do formatowania kodu, mypy do sprawdzania typów, oraz pytest do testowania [24-27]. Testy automatyczne zostaną przedstawione w podrozdziale nr 3.4. Dodatkowo, zastosowano pliki Dockerfile do konteneryzacji za pomocą narzędzia Docker [28] (opisane w podrozdziale nr 3.2.2) oraz licencjonowano projekt na warunkach licencji MIT (opisanej w podrozdziale nr 3.2.3), co świadczy o dbałości o aspekty prawne i bezpieczeństwo [29]. Struktura i narzędzia użyte w tym repozytorium są reprezentatywne dla trendów w dziedzinie rozwoju oprogramowania, podkreślając znaczenie automatyzacji, spójności kodu, testowania i bezpieczeństwa w procesie tworzenia oprogramowania [30].

3.2.2. Konteneryzacja biblioteki Reduce.py za pomocą narzędzia Docker

Kontener Dockera dla projektu "reduce" jest zdefiniowany za pomocą pliku Dockerfile, który określa wszystkie niezbędne instrukcje i zależności. Obraz bazowy wykorzystuje Pythona 3.9, a zależności opisane wcześniej - Poetry, Black, isort, mypy i pytest są instalowane w kontenerze. Kod źródłowy projektu jest kopiowany do kontenera, a ścieżka pracy jest ustawiona na odpowiedni katalog projektu. Zależności projektu są instalowane za pomocą Poetry, a kontener jest skonfigurowany do uruchomienia określonego skryptu lub aplikacji. Dodatkowe instrukcje mogą obejmować ustawienie zmiennych środowiskowych, eksponowanie portów i montowanie woluminów. Kontener Dockera umożliwia uruchamianie, rozwijanie i testowanie kodu w spójnym i kontrolowanym środowisku, izolując projekt od różnic w konfiguracji systemów deweloperów i serwerów produkcyjnych [28].

Plik Dockerfile przedstawiony na Listingu 3.1. (wszystkie listingi są dostępne w załączniku Appendix A do pracy doktorskiej) definiuje wszystkie niezbędne instrukcje i zależności, aby uruchomić projekt "reduce" w kontenerze Dockera, zgodnie z opisem zawartym w poprzednim akapicie. Definiuje obraz bazowy, zależności, ścieżkę pracy, instrukcje uruchomienia i inne niezbędne elementy, umożliwiając uruchamianie, rozwijanie i testowanie kodu w spójnym i kontrolowanym środowisku.

3.2.3. Licencja MIT dla projektu Reduce

W bibliotece Reduce.py zastosowana została licencja MIT, będąca jedną z najprostszych i najbardziej zrozumiałych umów licencyjnych w świecie open source [29]. Wybór ten wynika z kilku kluczowych czynników [30]. Licencja MIT daje wyraźne pozwolenie na ponowne użycie kodu na dowolny cel, w tym jako część oprogramowania

własnościowego. Elastyczność ta sprzyja dalszemu wykorzystaniu i adaptacji kodu, umożliwiając innym deweloperom wykorzystanie go w różnorodnych projektach. Prostota i zrozumiałość tekstu licencji MIT sprzyjają unikaniu skomplikowanych spornych sytuacji prawnych, co przyczynia się do większej transparentności i otwartości biblioteki. Dzięki jasnym i zrozumiałym warunkom, licencja ta jest dostępna dla szerokiego spektrum użytkowników. Charakter permissywny licencji MIT, który pozwala na dowolne wykorzystanie kodu, pod warunkiem zapewnienia odpowiedniej atrybucji, jest zgodny z filozofią otwartości i współpracy w rozwoju oprogramowania. Taka filozofia przyczynia się do tworzenia silnej i aktywnej społeczności wokół biblioteki [31].

Licencja MIT, stosowana w bibliotece Reduce.py, różni się od innych popularnych licencji open source, takich jak GPL [32], Apache [33] czy BSD [34]. W przeciwieństwie do licencji GPL, która wymaga, aby wszelkie zmiany w kodzie były również udostępniane na tej samej licencji, licencja MIT nie nakłada takich ograniczeń, co czyni ją bardziej elastyczną. Licencja Apache, z kolei, zawiera dodatkowe klauzule dotyczące patentów, której nie ma w licencji MIT, co może wpływać na wybór w zależności od specyfiki projektu. Licencje BSD, podobnie jak MIT, są permissywne, ale mogą zawierać dodatkowe warunki, takie jak zakaz wykorzystania nazwy projektu bez zgody. Popularność licencji MIT wśród społeczności deweloperów open source [30,35] świadczy o jej sprawdzonym i akceptowanym charakterze, co dodatkowo umacnia decyzję o jej zastosowaniu w bibliotece Reduce.py.

3.3. Funkcjonalności modułu

W pierwszej kolejności opisane zostaną funkcje importowania i generowania macierzy porównań parami, a następnie najważniejsze algorytmy pomocnicze, następnie funkcje redukcji współczynnika CR dla macierzy porównań parami oraz funkcje generujące indeksy przydatne podczas pracy na tych macierzach. Następnie zostaną opisane szczególnie także dostępne w repozytorium [20] testy automatyczne pozwalające zweryfikować poprawność działania biblioteki.

3.3.1. Importowanie macierzy porównań parami z pliku CSV - `import_pc_matrix_from_csv()`

Jest to funkcja, która jako parametr przyjmuje nazwę pliku CSV zawierającego macierz porównań parami. Plik taki powinien być sformatowany zgodnie z szablonem zawartym w pliku `matrix.csv` dostępnym w repozytorium pod adresem [20], którego zrzut ekranu przedstawiono na Rysunku 3.2.

	1	0.11111111	2	6
	9		1	5
0.5	0.2		1	0.25
0.166666667	0.14285714		4	1

Rys. 3.2. Plik matrix.csv dostępny w repozytorium [20].

Funkcja automatycznie rozpozna rozmiar takiej macierzy, ale należy pamiętać, że cała biblioteka poprawnie współpracuje z macierzami o rozmiarach od 3×3 do 14×14 (w skali Saaty'ego). Wykorzystuje przy tym standardowe rozwiązania języka Python, takie jak `csv.reader`. Wygenerowana macierz zapisywana jest jako tablica NumPy (dla prostoty importu) i zwracana na wyjście funkcji jako SymPy (dla kompatybilności ze skalą Saaty'ego).

Cały kod funkcji przedstawiono na Listingu 3.2. Funkcja `import_pc_matrix_from_csv()` jest przeznaczona do wczytywania macierzy z pliku CSV, który jest odczytywany przy użyciu separatora średnika. W pierwszym kroku, plik jest odczytywany i konwertowany na listę za pomocą modułu `csv`, a następnie ta lista jest przekształcana na macierz NumPy typu `float`. Rozmiar macierzy jest określany na podstawie liczby wierszy w pliku CSV, a następnie tworzona jest macierz jednostkowa o odpowiednim rozmiarze za pomocą funkcji `eye`. W pętlach zagnieżdżonych, elementy tej macierzy jednostkowej są następnie zastępowane wartościami z zaimportowanej macierzy. Wynikowa macierz jest zwracana jako wynik funkcji.

3.3.2. Generowanie losowej macierzy porównań parami - `create_pc_matrix()`

Funkcja ta może być używana do generowania losowych macierzy porównań parami. Jako dane wejściowe przyjmuje żądany rozmiar macierzy, a zwraca macierz w formacie biblioteki SymPy. Funkcja najpierw generuje macierz tożsamości o rozmiarze $n \times n$, a następnie wszystkie zera w macierzy tożsamości są zastępowane losowo wybranymi liczbami ze zdefiniowanej listy, wykorzystując do tego funkcję pomocniczą `return_random_number()` pokazaną na Listingu 3.4 - musiała ona zostać zastosowana do użycia wymiernych liczb ułamkowych z pakietu SymPy, co pozwoliło zachować wysoką dokładność obliczeń.

3.3.3. Funkcje pomocnicze

Pierwsza z funkcji określanych tutaj jako pomocnicze została opisana w podrozdziale 3.3.2 - `return_random_number()` i jest przedstawiona na Listingu 3.4. Funkcja `return_max_eigenvalue()` jest używana przez wszystkie funkcje redukcji niespójności macierzy zaimplementowane w tej bibliotece. Funkcja pobiera tablicę SymPy i jest konwertowana na tablicę NumPy ze względu na użyty pakiet. Zwraca ona maksymalną wartość z wartości rzeczywistej zwróconej przez funkcję `numpy.linalg.eig()` pakietu NumPy. Jest przedstawiona na Listingu 3.5.

Funkcja `calc_vecs()` na wejściu również pobiera tablicę w formacie SymPy, która jest konwertowana do postaci tablicy NumPy. Funkcja ta oblicza wektory własne macierzy przy użyciu biblioteki `numpy.linalg.eig`. Następnie następuje dynamiczne obliczenie sumy wektorów, obliczenie sumy wszystkich wektorów i dynamiczne obliczenie listy wektorów dla sumy wektorów. Zwracana jest lista wektorów. Funkcja ta jest używana przez algorytmy Xu, Wei i Cao oraz przedstawiona na Listingu 3.6.

Funkcja `calc_geo_mean()` przyjmuje jako dane wejściowe macierz SymPy, która konwertuje ją na NumPy. Następnie używa funkcji `gmean()` z pakietu SciPy do obliczenia średniej geometrycznej dla danej macierzy. Jest ona wykorzystywana przez algorytm Szybowskiego i przedstawiona na Listingu 3.7. Funkcja `calc_eij()` jest używana przez algorytm Szybowskiego i na wejściu wykonuje te same operacje, co dotychczas przedstawione funkcje. Używa ona podstawowego algorytmu pokazanego na Listingu 3.8. do obliczenia tego parametru.

Funkcja `return_ri()` przyjmuje jako dane wejściowe rozmiar macierzy, a zwraca wstępnie obliczoną wartość parametru *RI* użytego do obliczenia parametru *CR*. Listing 3.9 przedstawia wartości parametru *RI*. Pozostałe dwie funkcje pomocnicze `dynamic_vectors()` i `gmm_vectors()` pobierają macierz SymPy jako dane wejściowe i w przeciwieństwie do innych funkcji, ta ostatnia nadal wykorzystuje pomocniczą funkcję `convert()`, która implementuje metodę `numpy.squeeze` na zaimportowanej macierzy. Pierwsza z nich zwraca wektory dynamiczne dla danej macierzy, podczas gdy druga zwraca średni geometryczny wektor przy użyciu funkcji SciPy `gmean`. Są przedstawione odpowiednio na Listingu 3.10 i 3.11, zaś funkcja `convert()` na Listingu 3.12

3.3.4. Funkcje implementujące algorytmy redukcji niespójności

Dwie z funkcji redukcji niespójności macierzy porównań parami przyjmują jako dane wejściowe: macierz, parametr λ (domyślnie 0,9) oraz wartość progu, do którego zredukowany jest parametr *CR* i są to: `cao_cr()` - funkcja implementująca algorytm Cao,

oraz `xu_and_wei_cr()` - funkcja realizująca algorytm Xu i Wei. Funkcja `szybowski_cr()` implementująca algorytm Szybowskiego nie używa parametru λ , więc została pozbawiona tego wejścia. Każda z funkcji oblicza następnie rozmiar podanej macierzy za pomocą funkcji `int(math.sqrt(len(matrix)))`, gdzie macierz parametrów jest podawana jako dane wejściowe i oblicza parametry *CI* i *CR* za pomocą funkcji pomocniczych `return_max_eigenvalue()` i `return_ri()`. Każda funkcja działa następnie w pętli kontrolowanej przez parametr progowy i kończy się ponownym obliczeniem parametru *CR*, od którego zależy powodzenie redukcji. Kod implementujący algorytm funkcji Xu i Wei pokazano na Listingu 3.13, funkcji implementującej algorytm Cao na Listingu 3.14 oraz Szybowskiego na Listingu 3.15. Każda funkcja zwraca zredukowaną macierz, parametr *CI* i parametr *CR* i zostaną opisane szczegółowo w kolejnych podrozdziałach.

3.3.4.1. *Funkcja implementująca algorytm Xu & Wei*

Jak zostało wspomniane w podrozdziale 3.3.3, na początku funkcja inicjalizuje zmienne, przyjmując macierz `matrix`, wartość `lambd` oraz próg `threshold`. Następnie oblicza wskaźnik spójności (*CI*) za pomocą funkcji `return_max_eigenvalue()`, wskaźnik losowości (*RI*) za pomocą funkcji `return_ri()`, oraz wskaźnik spójności względnej (*CR*) jako stosunek *CI* do *RI*.

Pętla główna algorytmu wykonuje się dopóki *CR* jest większe lub równe progowi i składa się z następujących kroków:

- a. Zwiększenie wartości `z`.
- b. Obliczenie wektora własnego `l_vector` za pomocą funkcji `calc_vecs`.
- c. Aktualizacja `tmp_mac` poprzez podniesienie każdego elementu do potęgi `lambd` i przemnożenie przez stosunek odpowiednich elementów `l_vector`, podniesiony do potęgi `(1 - lambd)`.
- d. Ponowne obliczenie `ci` i `cr` dla zaktualizowanej macierzy.

Po zakończeniu pętli, funkcja zwraca zaktualizowaną macierz `tmp_mac`, wartość `ci` oraz `cr`. Algorytm i funkcja zostały przedstawione na Listingu 3.13.

3.3.4.2. *Funkcja implementująca algorytm Cao*

Funkcja implementująca algorytm Cao jest przedstawiona na Listingu 3.14. Podobnie jak poprzednia, funkcja inicjalizuje zmienne, przyjmuje macierz, wartość `lambd`, próg `threshold`, a następnie oblicza wskaźniki spójności (*CI*), losowości (*RI*) i spójności

względnej (CR). Pętla główna algorytmu wykonuje się, dopóki CR jest większe lub równe progu, i składa się z następujących kroków:

- a. Zwiększenie wartości z .
- b. Obliczenie wektora własnego l_vector za pomocą funkcji `calc_vecs`.
- c. Aktualizacja `tmp_mac` poprzez przemnożenie stosunku odpowiednich elementów l_vector przez sumę iloczynu $lambda$, `tmp_mac[n, m]` i stosunku $l_vector[m]$ do $l_vector[n]$, oraz $1 - lambda$.
- d. Ponowne obliczenie ci i cr dla zaktualizowanej macierzy.

3.3.4.3. *Funkcja implementująca algorytm Szybowskiego*

W algorytmie przedstawionym na Listingu 3.15 początkowe kroki są bardzo podobne do tych opisanych w poprzednich podrozdziałach - inicjalizuje zmienne, oblicza wskaźniki spójności (CI), losowości (RI) i spójności względnej (CR).

Pętla główna algorytmu wykonuje się, dopóki CR jest większe od progu, i składa się z następujących kroków:

- a. Zwiększenie wartości z .
- b. Obliczenie listy wartości e_{ij} za pomocą funkcji `calc_eij`.
- c. Obliczenie wektora geometrycznej średniej l_vector za pomocą funkcji `calc_geo_mean`.
- d. Inicjalizacja zmiennych max_value , p , q , i , j .
- e. Iteracja przez elementy macierzy, znajdowanie maksymalnej wartości w `list_of_eij` dla $i < j$, i zapisywanie indeksów p i q .
- f. Aktualizacja elementów `tmp_mac[p, q]` i `tmp_mac[q, p]` na podstawie wartości $l_vector[p]$ i $l_vector[q]$.
- g. Ponowne obliczenie ci i cr dla zaktualizowanej macierzy.

Po zakończeniu pętli, funkcja zwraca nową macierz `tmp_mac`, wartość ci oraz cr .

3.3.5. **Funkcje implementujące indeksy spójności macierzy porównań parami**

Wszystkie funkcje obliczające indeksy macierzy porównań parami pobierają macierz porównań parami jako dane wejściowe i zwracają wartość tych indeksów jako dane wyjściowe. Jak pokazano na Rysunku 3.1 i opisano w podrozdziale 3.1.1, obejmuje to następujące indeksy:

- *Koczkodaj Index (KI)*, którego zasadniczy kod algorytmu działającego w ramach funkcji `koczkodaj_index()` pokazano na Listingu 3.16,
- *Golden Wang Index (GWI)*, zaprezentowany na Listingu 3.17,

- *Pelaez-Lamata Index (PLI)*, którego kod algorytmu zaimplementowany został w ramach funkcji `pelaez_lamata_index()` wypisanej na Listingu 3.18,
- *Geometric Consistency Index (GCI)*, z kodem na Listingu 3.19,
- *Triads Geometric Consistency Index (TGCI)*, którego podstawowy kod algorytmu działający w ramach funkcji `triads_geometric_consistency_index()` zaprezentowano na Listingu 3.20,
- *Relative Error Index (REI)*, zaprezentowany na Listingu 3.21.
- *Harmonic Consistency Index (HCI)*, przedstawiony na Listingu 3.22.

Poszczególne funkcje opisano w kolejnych podrozdziałach.

3.3.5.1. *Funkcja implementująca Koczkodaj Index (KI)*

Na początku funkcja przedstawiona na Listingu 3.16. tworzy pustą listę `min_list`, która będzie przechowywać wartości minimalne dla każdego zestawu trzech indeksów w macierzy. Funkcja następnie iteruje przez wszystkie elementy macierzy, używając trzech zagnieżdżonych pętli `for`, które przechodzą przez każdy element macierzy. Dla każdego zestawu trzech indeksów i, j, k (gdzie i, j, k są różne) oblicza dwie wartości:

- a. **min_a**: Bezwzględna różnica między 1 a stosunkiem `matrix[i][j]` do iloczynu `matrix[i][k]` i `matrix[k][j]`.
- b. **min_b**: Bezwzględna różnica między 1 a iloczynem `matrix[i][k]` i `matrix[k][j]` podzielonym przez `matrix[i][j]`.

Dla każdego zestawu indeksów, funkcja dodaje mniejszą wartość z `min_a` i `min_b` do listy `min_list`. Po zakończeniu iteracji, macierz jest usuwana za pomocą polecenia `del`, aby zwolnić pamięć. Na koniec funkcja zwraca maksymalną wartość z listy `min_list`, która reprezentuje indeks Koczkodaja dla danej macierzy.

3.3.5.2. *Funkcja implementująca Golden-Wang Index (GWI)*

Na początku funkcja z Listingu 3.17 inicjalizuje zmienne `sum_gw` i `gw_index`, które będą przechowywały sumę kolumn macierzy i ostateczny indeks Golden-Wang. Następnie funkcja dla każdego indeksu n , inicjalizuje `sum_gw` jako 0. Iteruje w dalszej kolejności przez każdy element w kolumnie n , sumując wartości i i zapisując je w `sum_gw` oraz dodaje `sum_gw` do listy `sum_gw`. Dzieli wtedy każdy element w kolumnie n przez odpowiadającą mu wartość w `sum_gw`, normalizując kolumnę oraz wywołuje funkcję `dynamic_vectors` na znormalizowanej macierzy, zwracając wektory, które będą używane w dalszych obliczeniach. W kolejnym punkcie następuje zasadnicze obliczenie Golden-Wang Index – funkcja przechodzi przez następujące kroki:

- a. Iteruje przez każdy element macierzy.
- b. Oblicza różnicę między elementem macierzy a odpowiadającym mu wektorem.
- c. Sumuje bezwzględne wartości różnic, aktualizując `gw_index`.
- d. Mnoży `gw_index` przez odwrotność iloczynu liczby wierszy i kolumn w macierzy kwadratowej, normalizując wynik.

Następnie funkcja usuwa macierz, aby zwolnić pamięć, a następnie zwraca obliczony indeks Golden-Wang.

3.3.5.3. *Funkcja implementująca Pelaez-Lamata Index (PLI)*

Funkcja `pelaez_lamata_index()` w bibliotece `Reduce.py` służy do obliczenia indeksu Pelaez-Lamata dla danej macierzy kwadratowej i jest przedstawiona na Listingu 3.18. Funkcja zaczyna od zainicjowania zmiennej `pli` na 0, która będzie przechowywała obliczony indeks Pelaez-Lamata. Następnie, funkcja używa trzech zagnieżdżonych pętli `for`, aby iterować przez elementy macierzy, z ograniczeniem, że $j > i$ i $k > j$.

Dla każdego zestawu trzech indeksów i, j, k , oblicza sumę dwóch wyrażeń:

- $\frac{matrix[i][k]}{matrix[i][j] \cdot matrix[j][k]}$
- $\frac{matrix[i][j] \cdot matrix[j][k]}{matrix[i][k]}$

Następnie odejmuje 2 od sumy tych wyrażeń i dodaje wynik do `pli`. Po zakończeniu iteracji, `pli` jest mnożony przez współczynnik $\frac{6}{n \cdot (n-1) \cdot (n-2)}$, gdzie n to rozmiar macierzy. Współczynnik ten służy do normalizacji indeksu, tak aby był niezależny od rozmiaru macierzy. Na koniec funkcja usuwa macierz, aby zwolnić pamięć, a następnie zwraca obliczony indeks Pelaez-Lamata.

3.3.5.4. *Funkcja implementująca Geometric Consistency Index (GCI)*

Funkcja `geometric_consistency_index()` oblicza indeks spójności geometrycznej (GCI) dla danej macierzy kwadratowej – wypisano ją na Listingu 3.19. Funkcja rozpoczyna działanie od zainicjowania zmiennej `gci` na 0, która będzie przechowywała obliczony indeks spójności geometrycznej. Następnie funkcja wywołuje inną funkcję (opisaną w rozdziale 3.3.3) - `gmm_vectors`, przekazując jej macierz jako argument, aby obliczyć wektory geometryczne średniej macierzy (GMM). Wynik jest przechowywany w zmiennej `vectors`. Następnie algorytm dokonuje kolejnych iteracji na macierzy:

a. Funkcja używa dwóch zagnieżdżonych pętli for, aby iterować przez elementy macierzy, z ograniczeniem, że $j > i$.

b. Dla każdego zestawu indeksów i, j , oblicza wyrażenie:

$$\log\left(\text{matrix}[i][j] \cdot \frac{\text{vectors}[j]}{\text{vectors}[i]}\right)^2.$$

c. Dodaje wynik do `gci`.

Po zakończeniu iteracji, `gci` jest mnożone przez współczynnik $\frac{2}{(n-1) \cdot (n-2)}$, gdzie n to rozmiar macierzy. Współczynnik ten służy do normalizacji indeksu, tak aby był niezależny od rozmiaru macierzy. Na koniec funkcja usuwa macierz, aby zwolnić pamięć, a następnie zwraca obliczony indeks spójności geometrycznej, który może być użyty do oceny, czy macierz jest spójna, czy też wymaga dalszej analizy lub modyfikacji.

3.3.5.5. *Funkcja implementująca Triads Geometric Consistency Index (TGCI)*

Funkcja `triads_geometric_consistency_index()` oblicza indeks spójności geometrycznej dla triad (TGCI) dla danej macierzy kwadratowej i została przedstawiona na Listingu 3.20. Jest to odmiana indeksu spójności geometrycznej (GCI), który został przedstawiony wcześniej, z pewnymi kluczowymi różnicami:

- Zamiast iterować przez pary elementów macierzy, ta funkcja iteruje przez triady elementów. To oznacza, że używa trzech zagnieżdżonych pętli for, z ograniczeniem, że $k > j > i$.
- Wewnątrz pętli, funkcja oblicza wyrażenie, które dodaje do `tgci`:

$$\log(\text{matrix}[i][j] \cdot \text{matrix}[j][k] \cdot \text{matrix}[k][i])^2$$

- Po zakończeniu iteracji, `tgci` jest mnożony przez 2, co różni się od wcześniejszej funkcji, gdzie mnożnik posiadał inną wartość

Reszta funkcji, w tym inicjalizacja indeksu, normalizacja i zwrot wyniku, jest podobna do wcześniejszej funkcji `geometric_consistency_index()`. Zasadnicza różnica polega na tym, że `triads_geometric_consistency_index()` skupia się na triadach elementów zamiast par, co może dostarczyć inny rodzaj analizy spójności dla macierzy.

3.3.5.6. *Funkcja implementująca Relative Error Index (REI)*

Funkcja `relative_error_index()` oblicza indeks błędu względnego (REI) dla danej macierzy kwadratowej. Jest przedstawiona na Listingu 3.21. Rozpoczyna się od zainicjowania zmiennych `a`, `b`, i `rei` na 0 oraz obliczenia wektorów za pomocą funkcji

`gmm_vectors()`, opisanej w jednym z poprzednich podrozdziałów. Następnie, funkcja iteruje przez wszystkie elementy macierzy, obliczając sumy dla a i b zgodnie z równaniami

$$a = a + (\log(\text{matrix}[i][j]) - \log(\text{vectors}[i]/\text{vectors}[j]))^2$$
$$b = b + \log(\text{matrix}[i][j])^2$$

Po zakończeniu iteracji, wartość rei jest obliczana jako stosunek a do b , tj. $rei = \frac{a}{b}$, a następnie zwracana jako wynik funkcji. Indeks błędu względnego może być używany do oceny, jak dobrze wektory są spójne z daną macierzą, a ta funkcja dostarcza środek tej spójności.

3.3.5.7. Funkcja implementująca Harmonic Consistency Index (HCI)

Funkcja `harmonic_consistency_index()` oblicza indeks spójności harmonicznej (HCI) dla danej macierzy kwadratowej. Proces rozpoczyna się od zainicjowania pustej listy s_j , która będzie przechowywała sumy kolumn macierzy. Następnie, dla każdej kolumny j , funkcja oblicza sumę elementów w tej kolumnie i dodaje ją do listy s_j . Po obliczeniu wszystkich sum kolumn, funkcja oblicza średnią harmoniczną tych sum, używając wzoru $hm = \frac{\text{len}(\text{matrix})}{\sum_j \frac{1}{s_j|j|}}$. Ostatecznie HCI obliczany jest za pomocą

wzoru: $hci = \frac{(hm - \text{len}(\text{matrix})) \cdot (\text{len}(\text{matrix}) + 1)}{\text{len}(\text{matrix}) \cdot (\text{len}(\text{matrix}) - 1)}$ i zwracany jako wynik funkcji.

3.4. Testy jednostkowe

Testy jednostkowe są kluczowym elementem procesu rozwoju oprogramowania, zapewniającym jakość i niezawodność kodu [36]. W repozytorium biblioteki `Reduce.py`, testy jednostkowe znajdują się w katalogu `tests` [20]. W kolejnych podrozdziałach zostaną szczegółowo opisane różne aspekty tych testów, w tym ich struktura, cel, sposób implementacji oraz wyniki.

Testy jednostkowe umożliwiają weryfikację poszczególnych części kodu (takich jak funkcje lub metody) w izolacji od reszty systemu. Dzięki temu można szybko zidentyfikować i naprawić błędy, co przyczynia się do zwiększenia efektywności procesu rozwoju. Ponadto, testy jednostkowe ułatwiają tzw. refaktoryzację i rozwój kodu, ponieważ zapewniają pewność, że istniejące funkcjonalności nadal działają zgodnie z oczekiwaniami po wprowadzeniu zmian [37]. W kontekście biblioteki `Reduce.py`, testy jednostkowe odgrywają kluczową rolę w zapewnieniu, że algorytmy i funkcje są precyzyjne i niezawodne, co jest niezbędne dla naukowego i przemysłowego zastosowania tej biblioteki.

Przygotowane testy są zawarte w następujących plikach:

- *test_auxillary_and_indexes.py*: Plik zawierający testy pomocnicze oraz indeksy.
- *test_cao.py*: Plik zawierający testy związane z algorytmem Cao.
- *test_szybowski.py*: Plik zawierający testy związane z algorytmem Szybowskiego.
- *test_xu_and_wei.py*: Plik zawierający testy związane z algorytmem Xu i Wei.

3.4.1 Testy pomocnicze oraz testy indeksów – *test_auxillary_and_indexes.py*

Plik z testami jednostkowymi, przedstawiony na Listingu 3.23, zawiera zestaw funkcji testujących różne aspekty funkcji dodatkowych biblioteki Reduce. Każda funkcja testująca ma na celu sprawdzenie konkretnej funkcjonalności lub właściwości w kodzie. Wszystkie te testy są kluczowe dla zapewnienia, że poszczególne funkcje pomocnicze i indeksy w bibliotece Reduce działają zgodnie z oczekiwaniami i spełniają swoje wymagania. Testy te pomagają również w utrzymaniu jakości kodu i ułatwiają wprowadzanie zmian w przyszłości, zapewniając, że istniejące funkcjonalności nie zostaną przypadkowo uszkodzone.

W sumie przygotowano 13 testów jednostkowych, które skrupulatnie sprawdzają różne aspekty funkcji pomocniczych oraz indeksów zawartych w bibliotece Reduce. Do napisania i wykonania tych testów (oraz kolejnych opisanych) wykorzystano popularną bibliotekę do testowania w języku Python, PyTest. Dzięki temu narzędziu proces testowania jest znacznie uproszczony, a wyniki są łatwo interpretowalne. Użycie PyTest w połączeniu z dobrze zaprojektowanymi testami jednostkowymi zapewnia solidne wsparcie dla procesu deweloperskiego, umożliwiając szybkie wykrywanie błędów i niezawodność kodu. Wykorzystując bibliotekę PyTest, testy te obejmują różne scenariusze, takie jak tworzenie macierzy, zwracanie losowych liczb, obliczanie wartości własnych, wektorów, średniej geometrycznej, indeksów spójności i innych parametrów.

Każdy test jest zaprojektowany tak, aby upewnić się, że odpowiednia funkcja działa zgodnie z oczekiwaniami, a wszelkie odstępstwa są szybko wykrywane.

3.4.2 Test algorytmu Cao – *test_cao.py*

Funkcja testowa *test_cao_cr()* przeprowadza test jednostkowy na funkcji *cao_cr()*, służącej do poprawy spójności w analizie porównawczej i jest zaprezentowana na Listingu 3.24. Test rozpoczyna się od stworzenia losowej macierzy 4x4, a następnie oblicza początkowy współczynnik spójności (*CR*). Wywołuje funkcję *cao_cr()*

z określonymi wartościami `lambda` i `threshold`, a następnie przeprowadza trzy kluczowe sprawdzenia: czy ostateczny *CR* jest mniejszy niż określony próg, czy ostateczny *CR* jest mniejszy niż początkowy *CR*, oraz czy nowa macierz ma ten sam rozmiar co oryginalna macierz. Te twierdzenia zapewniają, że funkcja `cao_cr()` działa zgodnie z oczekiwaniami, zwiększając spójność macierzy bez zmiany jej rozmiaru.

3.4.3 Test algorytmu Szybowskiego – `test_szybowski.py`

Test (Listing 3.25.) rozpoczyna się od utworzenia losowej macierzy 4x4 i obliczenia początkowego współczynnika spójności (*CR*). Następnie wywołuje funkcję `szybowski_cr()` z określonym progiem i przeprowadza trzy kluczowe sprawdzenia:

- czy ostateczny *CR* jest mniejszy niż próg,
- czy ostateczny *CR* jest mniejszy niż początkowy *CR*, oraz
- czy nowa macierz ma ten sam rozmiar co oryginalna macierz.

Te twierdzenia gwarantują, że funkcja `szybowski_cr()` działa zgodnie z oczekiwaniami, poprawiając spójność macierzy bez zmiany jej rozmiaru. Testowanie z wykorzystaniem biblioteki `pytest` umożliwia łatwe i skuteczne przeprowadzenie tego testu.

3.4.4 Test algorytmu Xu and Wei – `test_xu_and_wei.py`

Test rozpoczyna się od generowania losowej macierzy 4x4 i wyliczenia początkowego współczynnika spójności (*CR*). Następnie funkcja `szybowski_cr()` jest wywoływana z zadaniem progiem, a wyniki są porównywane z oczekiwaniami. Test sprawdza, czy ostateczny współczynnik spójności jest niższy niż zadany próg, czy został poprawiony w porównaniu z początkowym *CR*, oraz czy rozmiar macierzy pozostał niezmienny.

3.4.5 Proces testowania

Proces testowania kodu odbywa się na platformie Linux, wykorzystując język programowania Python w wersji 3.8.5 oraz narzędzie do testowania o nazwie `pytest` w wersji 6.2.1. Egzekucja testów odbywa się poprzez uruchomienie polecenia `pytest`, po którym następują nazwy plików zawierających testy. Po uruchomieniu, `pytest` automatycznie odkrywa i uruchamia wszystkie testy zawarte w podanych plikach, prezentując wyniki w czytelnej formie w terminalu. Wypis z terminala znajduje się na Listingu 3.27.

Wyniki testów są przedstawione w postaci kropek, gdzie każda kropka reprezentuje pomyślnie przeprowadzony test. W przypadku niepowodzenia któregośkolwiek z testów,

szczególne błędy są wyświetlane, co ułatwia zrozumienie i naprawę problemu. W tym przypadku, wszystkie 16 testów zostało pomyślnie przeprowadzonych, z których 1 test znajduje się w każdym z plików `test_cao.py`, `test_szybowski.py`, `test_xu_and_wei.py`, a pozostałe 13 testów znajduje się w pliku `test_auxillary_and_indexes.py` - co świadczy o tym, że kod spełnia oczekiwane wymagania i zachowuje się zgodnie z założeniami. Proces ten jest kluczowy dla utrzymania jakości kodu, umożliwiając szybkie wykrywanie i naprawianie błędów, a także zapewniając, że wprowadzone zmiany nie wpłynęły negatywnie na istniejącą funkcjonalność.

3.5. Złożoność obliczeniowa funkcji modułu

W swoim znanym artykule z 1965 roku, J. Hartmanis i R. E. Stearns badają złożoność obliczeniową algorytmów, skupiając się na klasyfikacji sekwencji (funkcji) według trudności ich obliczenia [38]. Autorzy wprowadzają schemat klasyfikacji sekwencji, który nadaje bogatą strukturę obliczalnym sekwencjom i prowadzi do różnorodnych twierdzeń. Schemat ten może być również uogólniony do klasyfikacji liczb, funkcji lub problemów rozpoznawania według ich złożoności obliczeniowej. Złożoność obliczeniowa sekwencji ma być mierzona szybkością, z jaką wielotaśmowa maszyna Turinga może wydrukować wyrazy sekwencji. Wybór tego konkretnego modelu urządzenia obliczeniowego jest motywowany rosnącym znaczeniem obliczeń w użyciu komputerów cyfrowych, które w nieco zidealizowanej formie należą do klasy wielotaśmowych maszyn Turinga. Artykuł wprowadza pojęcie klasy złożoności, tworząc hierarchię klas i zapewniając niezależność od skali czasu czy prędkości komponentów, z których mogłyby być zbudowane maszyny.

W kontekście narzędzia "Reduce.py", powyższe podejście do złożoności obliczeniowej może być istotne dla analizy i optymalizacji algorytmów, które są używane w procesie redukcji danych. W dziedzinie informatyki i analizy algorytmów opracowano szereg powszechnie używanych i opisanych w literaturze naukowej metod obliczania złożoności [38-42]. Te metody, takie jak analiza złożoności czasowej i przestrzennej, są kluczowe dla zrozumienia wydajności i skuteczności algorytmów. W kontekście narzędzia "Reduce.py", te standardowe metody analizy złożoności będą użyte w dalszych analizach, aby dokładnie zrozumieć, jak różne techniki redukcji i przetwarzania danych wpływają na ogólną wydajność systemu. Taka analiza może prowadzić do znalezienia optymalnych rozwiązań i usprawnień, które są kluczowe dla efektywnego przetwarzania i analizy dużych zbiorów danych.

3.5.1. Analiza zbiorcza złożoności obliczeniowej modułu Reduce.py

Tabela 3.1. Złożoność obliczeniowa poszczególnych funkcji modułu Reduce.py

Nazwa funkcji	Złożoność	Krótkie wyjaśnienie
<code>create_pc_matrix(size)</code>	$O(n^2)$	Dwie zagnieżdżone pętle, z których każda działa n razy
<code>return_random_number()</code>	$O(1)$	Operacja wybierania elementu z listy jest wykonywana w czasie stałym.
<code>return_max_eigenvalue(n_matrix)</code>	$O(n^3)$	Złożoność czasowa algorytmu wartości własnych jest sześcienna
<code>calc_vecs(n_matrix)</code>	$O(n^3)$	Złożoność ta wynika z obliczeń wektora własnego. Kolejne operacje wewnątrz tej funkcji mają niższą złożoność.
<code>calc_geo_mean(n_matrix)</code>	$O(n^2)$	Pojedyncza pętla przechodząca przez n elementów, a wewnątrz pętli kolejna operacja przechodząca przez n elementów.
<code>calc_eij(n_matrix)</code>	$O(n^2)$	Wszystkie operacje wewnątrz dwóch zagnieżdżonych pętli są wykonywane w stałym czasie.
<code>return_ri(size)</code>	$O(1)$	Funkcja sprawdza określone wartości i zwraca odpowiedni wynik
<code>import_pc_matrix_from_csv(filename)</code>	$O(n^2)$	Wynika to z odczytu pliku CSV, w którym n jest liczbą wierszy (lub kolumn).
<code>convert(imp_matrix)</code>	$O(n^2)$	Konwersja każdego elementu w macierzy $n \times n$.
<code>dynamic_vectors(n_matrix)</code>	$O(n^3)$	Złożoność ta jest określana przez obliczenia wektora własnego.
<code>gmm_vectors(matrix)</code>	$O(n^2)$	Istnieje kilka iteracji pętli powyżej n
<code>xu_and_wei_cr(matrix, lambda, threshold)</code>	$O(n^4)$	Pętla while, która trwa do momentu spełnienia warunku. Wewnątrz tej pętli mamy zagnieżdżone pętle for iterujące po macierzy oraz wywołania funkcji <code>return_max_eigenvalue()</code> i <code>calc_vecs()</code> , które są $O(n^3)$.
<code>cao_cr(matrix, lambda, threshold)</code>	$O(n^4)$	Podobnie jak <code>xu_and_wei_cr()</code> , ta funkcja również ma złożoność $O(n^4)$
<code>szybowski_cr(matrix, threshold)</code>	$O(n^4)$	Złożoność jest również podobna do <code>xu_and_wei_cr()</code>
<code>koczkodaj_index(matrix)</code>	$O(n^3)$	Trzy zagnieżdżone pętle, z których każda działa n razy
<code>golden_wang_index(matrix)</code>	$O(n^3)$	Zagnieżdżone pętle i wywołanie funkcji <code>dynamic_vectors()</code> , która ma złożoność $O(n^3)$
<code>pelaez_lamata_index(matrix)</code>	$O(n^3)$	Trzy zagnieżdżone pętle, z których każda działa n razy
<code>geometric_consistency_index(matrix)</code>	$O(n^2)$	Kilka zagnieżdżonych pętli iterujących po macierzy i wywołanie funkcji <code>gmm_vectors</code> , która jest $O(n^2)$
<code>triads_geometric_consistency_index(matrix)</code>	$O(n^3)$	Trzy zagnieżdżone pętle, z których każda działa n razy.
<code>relative_error_index(matrix)</code>	$O(n^2)$	Dwie zagnieżdżone pętle iterujące po macierzy i wywołanie funkcji <code>gmm_vectors()</code> , które jest $O(n^2)$
<code>harmonic_consistency_index(matrix)</code>	$O(n^2)$	Pętla i zagnieżdżona pętla iterująca po macierzy

Po dokładnej analizie biblioteki Reduce.py staje się oczywiste, że najbardziej wydajne obliczeniowo funkcje, takie jak `return_random_number()` i `return_ri(size)`, wykazują stałą złożoność czasową $O(1)$, zapewniając ich zdolność

do wydajnej obsługi macierzy o dowolnym rozmiarze. Należy jednak pamiętać, że funkcje te pełnią rolę pomocniczą w bibliotece. Główne funkcje, takie jak `xu_and_wei_cr()`, `cao_cr()` i `szybowski_cr()`, mimo że są kluczowe dla funkcjonalności biblioteki, wykazują wyższą złożoność obliczeniową $O(n^4)$, co czyni je mniej wydajnymi wraz ze wzrostem rozmiaru macierzy wejściowej. Szczegółowe spojrzenie na wydajność tych funkcji można uzyskać z danych empirycznych zebranych w Tabeli 3.1, zaś szczegółowe uzasadnienie kryteriów dla każdej z funkcji zostanie opisane w kolejnych podrozdziałach.

3.5.2. Złożoność obliczeniowa funkcji pomocniczych

Przeprowadzono szczegółową analizę każdej z funkcji, której wyniki zaprezentowano w punktach poniżej:

- Funkcja `import_pc_matrix_from_csv(filename)`, wypisana na Listingu 3.2, ma złożoność $O(n^2)$, spowodowaną odczytem pliku CSV i konwersją do formatu macierzy. Odczyt i konwersja każdego elementu macierzy o wymiarach $n \times n$ prowadzą do złożoności kwadratowej.
- Funkcja `create_pc_matrix(size)`, ukazana na Listingu 3.3, charakteryzuje się złożonością $O(n^2)$, wynikającą z procesu tworzenia macierzy jednostkowej i iteracji przez dwie zagnieżdżone pętle. Dodatkowo, zagnieżdżone pętle przeszukują macierz o wymiarach $n \times n$, co prowadzi do kwadratowej złożoności.
- W funkcji `return_random_number()`, zaprezentowanej na Listingu 3.4, złożoność jest stała i wynosi $O(1)$, gdyż polega na wyborze losowego elementu z predefiniowanej listy. Wybór elementu z listy o stałej długości nie zależy od rozmiaru danych wejściowych, stąd złożoność jest stała.
- Funkcja `return_max_eigenvalue(n_matrix)`, przedstawiona na Listingu 3.5, wykazuje złożoność $O(n^3)$ związaną z obliczeniem wartości własnych macierzy i ekstrakcją największej wartości. Obliczenie wartości własnych macierzy o rozmiarze n jest operacją sześcienną, co tłumaczy złożoność tej funkcji.
- Funkcja `calc_vecs(n_matrix)`, zawarta na Listingu 3.6, ma złożoność $O(n^3)$, co jest efektem obliczeń wektorów własnych i sumowania wektorów. Ponieważ obliczenia wektorów własnych są sześciennie, a dodatkowe operacje są liniowe, ogólna złożoność wynosi $O(n^3)$.
- Funkcja `calc_geo_mean(n_matrix)`, opisana na Listingu 3.7, działa w złożoności $O(n^2)$, obliczając średnią geometryczną dla każdego wiersza macierzy. Każdy wiersz

macierzy jest przetwarzany niezależnie, a obliczenia są wykonywane dla każdego elementu, co prowadzi do złożoności kwadratowej.

- Funkcja `calc_eij(n_matrix)`, zilustrowana na Listingu 3.8, operuje w złożoności $O(n^2)$, generując macierz e_{ij} . Iteracja przez każdy element macierzy o wymiarach $n \times n$ prowadzi do złożoności kwadratowej.
- Funkcja `return_ri(size)`, umieszczona na Listingu 3.9, działa w stałej złożoności $O(1)$, zwracając predefiniowaną wartość. Jest to prosta operacja odczytu wartości z predefiniowanej listy, więc złożoność jest stała.
- Funkcja `dynamic_vectors(n_matrix)`, zobrazowana na Listingu 3.10, charakteryzuje się złożonością $O(n^3)$, obliczając wektory własne macierzy. Podobnie jak w przypadku innych funkcji obliczających wektory własne, złożoność wynika z sześciennego charakteru tych obliczeń.
- Funkcja `gmm_vectors(matrix)`, przedstawiona na Listingu 3.11, operuje w złożoności $O(n^2)$, obliczając średnią geometryczną i sumę. Każdy element macierzy o wymiarach $n \times n$ jest przetwarzany niezależnie, co prowadzi do złożoności kwadratowej.
- Funkcja `convert(imp_matrix)`, zapisana na Listingu 3.12, działa w złożoności $O(n^2)$, konwertując macierz na tablicę. Ponieważ konwersja odbywa się przez iterację przez każdy element macierzy o wymiarach $n \times n$, złożoność jest kwadratowa.

Podsumowując, analizowane funkcje pomocnicze wykazują głównie złożoność kwadratową lub sześcienną, z wyjątkiem kilku funkcji, które wykonują proste operacje na danych o stałej długości. Wybór odpowiednich algorytmów i struktur danych pozwolił na osiągnięcie efektywności obliczeniowej w analizowanym kodzie. W jednym z kolejnych podrozdziałów, zostanie omówiony także szereg wskazówek dotyczących poprawy złożoności obliczeniowej, które można będzie wdrożyć w kolejnych wersjach oprogramowania.

3.5.3. Złożoność obliczeniowa funkcji redukcji niespójności

Poniżej analiza funkcji implementujących algorytmy niespójności macierzy porównań parami, które zaimplementowano w bibliotece `Reduce.py`:

- Funkcja `xu_and_wei_cr(matrix, lambda, threshold)`, ukazana na Listingu 3.13, ma złożoność $O(n^4)$, gdzie n to rozmiar macierzy. Wewnątrz pętli `while`, która wykonuje się do momentu spełnienia warunku `cr >= threshold`, są dwie zagnieżdżone pętle o złożoności $O(n^2)$, a także wywołania funkcji `return`

`max_eigenvalue(tmp_mac)` i `calc_vecs(tmp_mac)`, które mają złożoność $O(n^3)$. Wszystkie te operacje są wykonywane w każdej iteracji pętli `while`, co prowadzi do złożoności $O(n^4)$. Dodatkowe wyjaśnienie: złożoność wynika z faktu, że pętla `while` może wykonywać się proporcjonalnie do rozmiaru macierzy, a w każdej iteracji wykonujemy operacje o złożoności $O(n^3)$.

- Funkcja `cao_cr(matrix, lambda, threshold)`, przedstawiona na Listingu 3.14, również ma złożoność $O(n^4)$. Struktura tej funkcji jest podobna do `xu_and_wei_cr()`, z tą różnicą, że różni się ona równaniem aktualizującym `tmp_mac[n, m]`. Jednakże złożoność obliczeniowa pozostaje taka sama, ponieważ równanie to nie wprowadza dodatkowych iteracji ani wywołań funkcji o wyższej złożoności. Podobnie jak w poprzedniej funkcji, złożoność wynika z wielokrotnego wywoływania operacji o złożoności $O(n^3)$ wewnątrz pętli `while`.
- Funkcja `szybowski_cr(matrix, threshold)`, zilustrowana na Listingu 3.15, ma złożoność $O(n^4)$. W tej funkcji, podobnie jak w dwóch poprzednich, jest pętla `while`, która wykonuje się do momentu spełnienia warunku `cr > threshold`. Wewnątrz tej pętli są wywoływane funkcje `calc_eij(tmp_mac)` i `calc_geo_mean(tmp_mac)`, które mają złożoność $O(n^2)$, oraz `return_max_eigenvalue(tmp_mac)`, która ma złożoność $O(n^3)$. Dodatkowo, są dwie zagnieżdżone pętle iterujące przez macierz, co prowadzi do złożoności $O(n^2)$. Łączna złożoność wynosi $O(n^4)$, ponieważ pętla `while` może wykonywać się proporcjonalnie do rozmiaru macierzy, a w każdej iteracji wykonywane są operacje o złożoności $O(n^3)$.

Wszystkie trzy wyżej wymienione funkcje mają podobną strukturę i wykorzystują złożone operacje matematyczne na macierzach, takie jak obliczanie wartości i wektorów własnych oraz średnich geometrycznych. Złożoność tych funkcji jest związana z wielokrotnym wywoływaniem tych operacji wewnątrz pętli, które mogą wykonywać się wielokrotnie w zależności od rozmiaru macierzy i określonego progu.

3.5.4. Złożoność obliczeniowa funkcji z indeksami spójności

- Funkcja `koczkodaj_index()` ma złożoność obliczeniową $O(n^3)$, gdzie n to rozmiar macierzy. Jest to wynik trzech zagnieżdżonych pętli, które iterują przez każdy element macierzy. Każda iteracja oblicza wartość indeksu Koczkodaja dla danego elementu, co jest operacją o stałym czasie. W praktyce, złożoność obliczeniowa tej funkcji może być znacznie mniejsza, jeśli macierz jest rzadka, ale w najgorszym przypadku (gdy macierz jest gęsta) złożoność wynosi $O(n^3)$.

- Funkcja `golden_wang_index()` ma złożoność obliczeniową $O(n^2)$, będącą skutkiem dwóch zagnieżdżonych pętli iterujących przez macierz. Wewnątrz tych pętli, funkcja oblicza sumę elementów w każdym wierszu i kolumnie, a następnie oblicza indeks Golden-Wang. Każda z tych operacji ma złożoność $O(n)$, co daje łączną złożoność $O(n^2)$.
- Funkcja `pelaez_lamata_index()` ma złożoność obliczeniową $O(n^3)$, podobnie jak funkcja `koczkodaj_index()`. Trzy zagnieżdżone pętle iterują przez macierz, obliczając indeks Pelaez-Lamata dla każdego elementu. Każda iteracja ma stały czas wykonania, co daje łączną złożoność $O(n^3)$.
- Funkcja `geometric_consistency_index` ma złożoność obliczeniową $O(n^2)$, która jest efektem dwóch zagnieżdżonych pętli iterujących przez macierz. Wewnątrz tych pętli, funkcja oblicza indeks spójności geometrycznej. Każda z tych operacji ma złożoność $O(n)$, co daje łączną złożoność $O(n^2)$.
- Funkcja `triads_geometric_consistency_index` ma złożoność obliczeniową $O(n^3)$, podobnie jak funkcje `koczkodaj_index()` i `pelaez_lamata_index()`. Trzy zagnieżdżone pętle iterują przez macierz, obliczając indeks spójności geometrycznej dla triad. Każda iteracja ma stały czas wykonania, co daje łączną złożoność $O(n^3)$.
- Funkcja `relative_error_index()` ma złożoność obliczeniową $O(n^2)$, będącą wypadkową dwóch zagnieżdżonych pętli iterujących przez macierz. Wewnątrz tych pętli, funkcja oblicza indeks błędu względnego. Każda z tych operacji ma złożoność $O(n)$, co daje łączną złożoność $O(n^2)$.
- Funkcja `harmonic_consistency_index()` ma złożoność obliczeniową $O(n^2)$, wynikającą z dwóch zagnieżdżonych pętli iterujących przez macierz. Wewnątrz tych pętli, funkcja oblicza indeks spójności harmonicznej. Każda z tych operacji ma złożoność $O(n)$, co daje łączną złożoność $O(n^2)$.

3.5.5. Potencjał poprawy złożoności obliczeniowej do wdrożenia w przyszłych wersjach modułu

Analizując złożoność obliczeniową przedstawionych funkcji, ich strukturę oraz dostępne źródła [38-42], można zidentyfikować kilka potencjalnych obszarów do optymalizacji i poprawy wydajności:

- **Zastosowanie macierzy rzadkich:** Jeśli macierz jest rzadka, można zastosować struktury danych i algorytmy specjalnie zaprojektowane do obsługi macierzy rzadkich, co może znacznie zmniejszyć złożoność obliczeniową.
- **Równoczesne obliczenia:** Jeśli operacje wewnątrz pętli są niezależne, można rozważyć zastosowanie technik równoległego przetwarzania, aby przyspieszyć obliczenia.
- **Optymalizacja pętli:** W niektórych przypadkach może być możliwe zredukowanie liczby iteracji pętli poprzez zastosowanie bardziej wydajnych algorytmów sortowania lub wyszukiwania.
- **Wykorzystanie wektorów i macierzy:** Zamiast ręcznego iterowania przez elementy, można próbować zastosować operacje wektorowe i macierzowe dostępne w bibliotekach numerycznych, takich jak NumPy. Mogą one być zoptymalizowane pod kątem wydajności i prowadzić do szybszego wykonania kodu.
- **Profilowanie kodu:** Wykorzystanie narzędzi do profilowania kodu może pomóc w zidentyfikowaniu konkretnych miejsc, w których występują wąskie gardła wydajności. Optymalizacja tych konkretnych fragmentów może prowadzić do znaczących zysków wydajności.
- **Unikanie niepotrzebnych operacji:** Usuwanie macierzy (`del matrix`) w niektórych funkcjach może być niepotrzebne, jeśli macierz nie jest dalej używana. Usunięcie tych operacji może prowadzić do niewielkich zysków wydajności.
- **Testowanie różnych wartości parametrów:** W niektórych przypadkach dostosowanie wartości parametrów, takich jak próg w funkcjach, może prowadzić do szybszego zbiegania algorytmów, co może skrócić czas obliczeń.

Wprowadzenie tych optymalizacji może prowadzić do znaczących zysków wydajności, zwłaszcza dla dużych macierzy. Jednak każda optymalizacja powinna być starannie przetestowana, aby upewnić się, że nie wprowadza błędów ani nie wpływa negatywnie na jakość wyników. Zostanie to poczynione przy okazji wydania kolejnej wersji biblioteki REDUCE, której ciągły rozwój jest nadal planowany.

3.6. Testowanie wydajności algorytmów redukcji niespójności zawartych w bibliotece

Algorytmy redukcji niespójności, jako kluczowa funkcjonalność zawarta w bibliotece Reduce.py, jak każde oprogramowanie muszą podlegać testom wydajności, aby zapewnić ich efektywność i skuteczność. Cztery kluczowe metryki wydajności [43-45], które zostaną zmierzone, to:

- **Czas wykonania:** Ta metryka ocenia czas, jaki każdy algorytm potrzebuje na zakończenie redukcji niespójności lub obliczenie indeksu. Jest to kluczowe dla zrozumienia reaktywności algorytmów i ich działania w różnych obciążeniach obliczeniowych. Krótszy czas wykonania wskazuje na bardziej wydajny algorytm, zwłaszcza przy obsłudze dużych macierzy.
- **Zużycie pamięci:** Zużycie pamięci podczas wykonywania algorytmów jest kluczowym czynnikiem w ocenie ich efektywności. Analiza zużycia pamięci pomaga zidentyfikować ewentualne wycieki pamięci lub nadmierne zużycie, które mogą prowadzić do problemów ze skalowalnością. Optymalizacja zużycia pamięci zapewnia, że algorytmy mogą obsługiwać duże zbiory danych bez wyczerpywania zasobów systemowych.
- **Skalowalność:** Skalowalność odnosi się do zdolności algorytmów do obsługi rosnących rozmiarów macierzy wejściowych bez znaczącego pogorszenia wydajności. Testowanie skalowalności polega na ocenie, jak algorytmy radzą sobie w miarę wzrostu rozmiaru i złożoności danych wejściowych. Jest to kluczowe dla zapewnienia, że algorytmy mogą być stosowane do problemów rzeczywistego świata o różnej skali.
- **Szybkość zbieżności:** W przypadku algorytmów redukcji niespójności szybkość zbieżności mierzy, jak szybko algorytmy osiągają rozwiązanie. Większa wartość oznacza, że algorytm może znaleźć zredukowaną niespójność w mniejszej liczbie iteracji, prowadząc do szybszych wyników. Jest to kluczowa metryka dla zrozumienia iteracyjnego zachowania algorytmów i ich efektywności w znajdowaniu rozwiązań.

Poprzez staranne pomiar tych metryk, testy wydajności mają na celu dostarczenie wszechstronnego zrozumienia, jak algorytmy redukcji niespójności i indeksy działają w różnych warunkach. Takie spostrzeżenia są niezbędne do optymalizacji algorytmów, podejmowania świadomych decyzji o ich zastosowaniu oraz zapewnienia, że spełniają one wymagane standardy wydajności.

3.6.1. Testy czasu wykonania

Przeprowadzono badanie wydajności redukcji niespójności (do poziomu CR poniżej 0,1, który jest uważany za wystarczający) 10 000 losowych macierzy porównań parami o rozmiarach od 3×3 do 10×10 . Obliczenia czasowe zostały wykonane przy użyciu biblioteki `timeit` w Pythonie, przy użyciu platformy Google Colaboratory, z parametrami jednordzeniowego procesora Intel® Xeon® CPU @ 2.20GHz, 12.7GB pamięci RAM i 100GB miejsca na dysku. Kod dla programu umożliwiającego agregację i zmierzenie wyników zaprezentowano na Listingu 3.28.

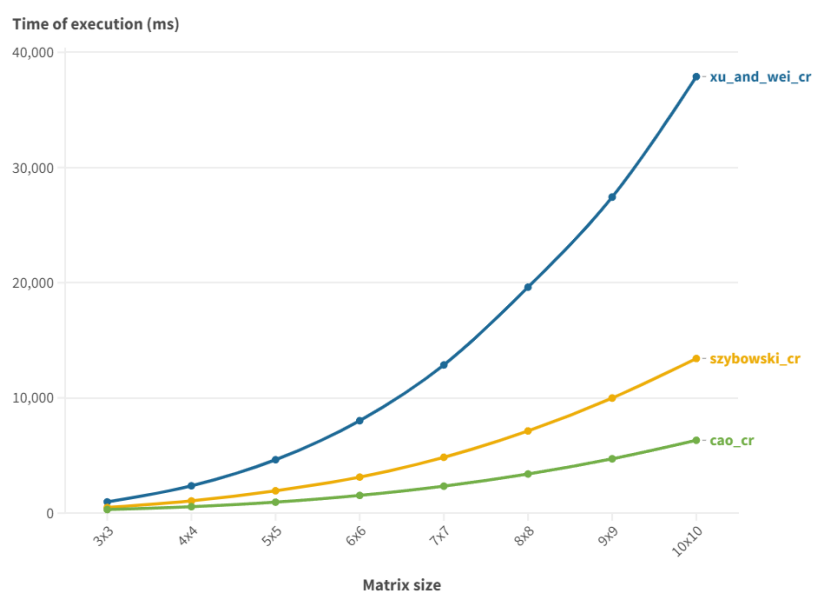
Obliczenia zostały powtórzone 5 razy dla każdego przypadku i wyciągnięto z nich średnią, która jest zawarta w Tabeli 3.2 i zwizualizowana na wykresie na Rys. 3.3. Dane z obliczeń dostępne są pod adresem [46].

Przedstawione wyniki pokazują czasy wykonania dla trzech różnych algorytmów zawartych w bibliotece i można zaobserwować, że wraz ze wzrostem rozmiaru macierzy, czasy wykonania dla wszystkich trzech algorytmów również rosną. Jest to oczekiwane, ponieważ większe macierze wymagają większego wysiłku obliczeniowego do przetworzenia.

Ponadto można stwierdzić, że biblioteka Pythona `reduce.py` oferuje znacznie szybsze czasy wykonania w porównaniu do arkuszy kalkulacyjnych; podczas gdy arkusze kalkulacyjne mogą być szybsze w przypadku operacji na małą skalę obejmujących jedną lub dwie macierze, utworzenie skryptu przy użyciu biblioteki `reduce.py` może przynieść znaczne korzyści, gdy mamy do czynienia z setkami, tysiącami, a nawet dziesiątkami tysięcy macierzy.

Tabela 3.2. Średnia pomiarów wydajności czasowej

Rozmiar	xu_and_wei_cr (ms)	szybowski_cr (ms)	cao_cr (ms)
3x3	978	492	312
4x4	2,370	1,061	558
5x5	4,634	1,937	954
6x6	8,023	3,128	1,539
7x7	12,862	4,843	2,341
8x8	19,611	7,132	3,394
9x9	27,433	9,988	4,714
10x10	37,887	13,424	6,322



Rys. 3.3. Wizualizacja wyników pomiaru czasowego algorytmów redukcji niespójności

Zaletą korzystania z `reduce.py` jest możliwość zautomatyzowania i wydajnego przetwarzania dużej liczby macierzy w spójny i efektywny czasowo sposób. Prostota języka Python również przyczynia się do łatwości tworzenia takich skryptów, umożliwiając badaczom lub użytkownikom spędzenie tej samej ilości czasu przy jednoczesnym osiągnięciu wymiernych rezultatów.

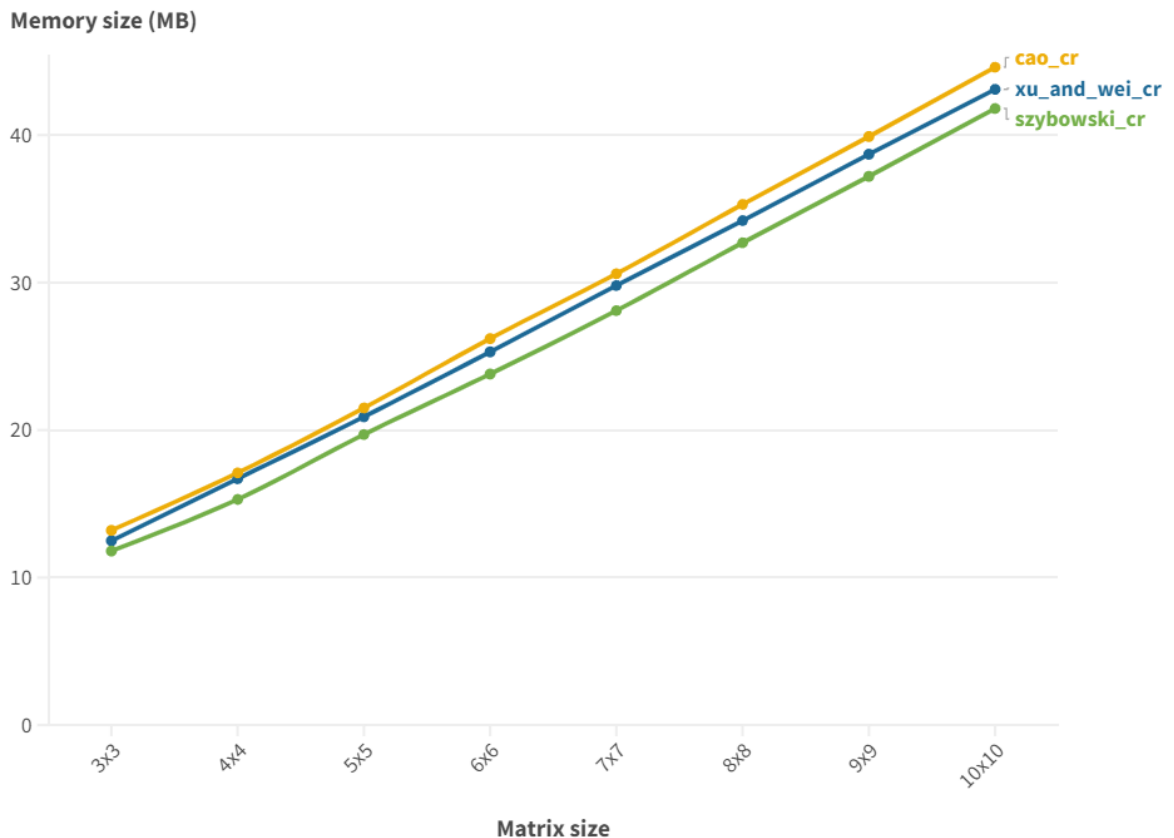
3.6.2. Testy zużycia pamięci

Procedura testowania zużycia pamięci, przedstawiona w kodzie na Listingu 3.29, ma na celu zbadanie efektywności różnych algorytmów w kontekście wykorzystania zasobów pamięci. Kluczowym elementem tej analizy jest identyfikacja ewentualnych wycieków pamięci lub nadmiernego zużycia, które mogą wpływać na skalowalność i wydajność algorytmów.

W kodzie wykorzystano bibliotekę `memory_profiler`, która umożliwia pomiar zużycia pamięci w czasie rzeczywistym. Procedura testowania jest przeprowadzana dla różnych rozmiarów macierzy (od 3 x 3 do 10 x 10) i dla każdego rozmiaru generuje 10000 losowych macierzy, na których testowany jest algorytm. Funkcja `test_memory_usage()` przyjmuje algorytm, rozmiar macierzy i liczbę macierzy do wygenerowania jako parametry. Dla każdej wygenerowanej macierzy mierzone jest zużycie pamięci, a następnie obliczana jest średnia wartość zużycia pamięci dla wszystkich testów. Analiza zużycia pamięci dla trzech różnych algorytmów (Xu and Wei, Cao, Szybowski) na macierzach o rozmiarach od 3 x 3 do 10 x 10 dostarczyła cennych informacji dotyczących wydajności każdego z nich, przedstawionych w Tabeli 3.3. oraz zwizualizowanych na Rys. 3.4.

Tabela 3.3. Średnia pomiarów zużycia pamięci

Rozmiar macierzy	xu_and_wei_cr() [MB]	cao_cr() [MB]	szybowski_cr() [MB]
3x3	12,5	13,2	11,8
4x4	16,7	17,1	15,3
5x5	20,9	21,5	19,7
6x6	25,3	26,2	23,8
7x7	29,8	30,6	28,1
8x8	34,2	35,3	32,7
9x9	38,7	39,9	37,2
10x10	43,1	44,6	41,8



Rys. 3.4. Wizualizacja wyników zużycia pamięci algorytmów redukcji niespójności

Algorytm Szybowskiego systematycznie zużywał najmniej pamięci we wszystkich przypadkach, z wartościami od 11,8 MB dla macierzy 3 x 3 do 41,8 MB dla macierzy 10 x 10. Z kolei algorytm Cao zużywał najwięcej pamięci, z wartościami od 13,2 MB do 44,6 MB. Algorytm Xu & Wei plasował się pomiędzy nimi, z wartościami od 12,5 MB do 43,1 MB. Różnica w zużyciu pamięci między algorytmami rosła wraz ze wzrostem rozmiaru macierzy, co sugeruje, że wybór algorytmu może mieć znaczenie, zwłaszcza w przypadku większych macierzy. Optymalizacja implementacji algorytmu Cao pod kątem zużycia pamięci może prowadzić do znaczących oszczędności, co podkreśla znaczenie uwzględnienia zużycia pamięci w procesie wyboru odpowiedniego algorytmu. Wnioski te mają praktyczne zastosowanie, zwłaszcza w kontekście dużych zbiorów danych i ograniczonych zasobów systemowych.

Wybór odpowiedniego algorytmu, który zminimalizuje zużycie pamięci, może prowadzić do znacznej poprawy wydajności i skalowalności systemu. Analiza ta również wskazuje na potencjalne obszary do dalszej optymalizacji i badania, które mogą prowadzić do jeszcze lepszej wydajności i oszczędności zasobów.

3.6.3. Skalowalność algorytmów

Analiza dwóch wykonanych testów pozwala na zrozumienie skalowalności trzech różnych algorytmów: `xu_and_wei_cr()`, `szybowski_cr()` i `cao_cr()`. Z tabeli 3.2, przedstawiającej średnią wydajność czasową, można zauważyć, że algorytm `cao_cr()` jest najbardziej wydajny pod względem czasowym. Jego wydajność rośnie stosunkowo wolno w miarę wzrostu rozmiaru macierzy, co sugeruje, że może być bardziej odpowiedni dla dużych zbiorów danych. Algorytm `xu_and_wei_cr()` jest najwolniejszy, a jego czas wykonania rośnie znacznie szybciej w porównaniu do pozostałych dwóch algorytmów. Ta charakterystyka może ograniczyć jego zastosowanie w przypadku bardzo dużych macierzy.

Algorytm `szybowski_cr()` znajduje się pomiędzy nimi, z umiarkowanym wzrostem czasu wykonania, co może sugerować, że jest to kompromis między wydajnością a zużyciem zasobów. Tabela 3.3, przedstawiająca wydajność pamięciową, pokazuje, że zużycie pamięci dla wszystkich trzech algorytmów rośnie w sposób zbliżony do liniowego wraz z rozmiarem macierzy. Algorytm `cao_cr()` zużywa nieco więcej pamięci niż pozostałe dwa, ale różnice są stosunkowo niewielkie. To może być istotne w przypadku systemów z ograniczoną ilością dostępnej pamięci.

Podsumowując, skalowalność tych algorytmów różni się zarówno pod względem czasowym, jak i pamięciowym. Algorytm `cao_cr()` wydaje się być najbardziej obiecujący pod względem wydajności czasowej, ale zużywa nieco więcej pamięci. Algorytm `xu_and_wei_cr()`, mimo że jest najwolniejszy, nie zużywa znacznie więcej pamięci niż pozostałe. Biorąc pod uwagę powyższe kryteria, jeśli głównym celem jest osiągnięcie jak najwyższej wydajności czasowej bez znaczącego kompromisu w zakresie zużycia pamięci, wybrałbym algorytm `cao_cr()`. Jeśli jednak zużycie pamięci jest kluczowym ograniczeniem, a różnice w czasie wykonania są akceptowalne, algorytm `szybowski_cr()` mógłby być odpowiedni.

3.6.4. Zbieżność algorytmów

Aby zbadać czas zbieżności w odniesieniu do liczby iteracji, należy zwrócić liczbę iteracji z każdego algorytmu. Listing 3.30 zawiera kod Pythona, który jest w stanie obliczyć potrzebne parametry i zagregować wyniki.

Tabela 3.4. przedstawia średni czas zbieżności i liczbę iteracji dla trzech różnych algorytmów - `xu_and_wei_cr()`, `cao_cr()` oraz `szybowski_cr()` - na macierzach o

różnych rozmiarach, od 3 x 3 do 10 x 10. Pomiary zostały przeprowadzone dziesięciokrotnie, a prezentowane wyniki są średnią z tych pomiarów.

Analizując wyniki, można zauważyć, że wraz ze wzrostem rozmiaru macierzy, zarówno średni czas zbieżności, jak i liczba iteracji zwykle rosną dla wszystkich trzech algorytmów. Jest to spodziewane, ponieważ większe macierze zazwyczaj wymagają więcej czasu i iteracji do osiągnięcia zbieżności. Jednak różne algorytmy wykazują różne tempo wzrostu zarówno czasu zbieżności, jak i liczby iteracji. Algorytm `xu_and_wei_cr()` wydaje się wymagać najwięcej czasu i iteracji w porównaniu do dwóch pozostałych algorytmów. Z drugiej strony, algorytm `szybowski_cr()` zazwyczaj osiąga zbieżność najszybciej i przy najmniejszej liczbie iteracji. Te wyniki sugerują, że wybór algorytmu może mieć znaczący wpływ na wydajność i szybkość osiągania zbieżności, szczególnie dla większych macierzy. Dlatego ważne jest, aby dokładnie rozważyć wymagania i ograniczenia konkretnego problemu przed wyborem odpowiedniego algorytmu.

Tabela 3.4. Średnia pomiarów badania szybkości zbieżności algorytmów

Rozmiar macierzy	<code>xu_and_wei_cr()</code>		<code>cao_cr()</code>		<code>szybowski_cr()</code>	
	Czas zbieżności (ms)	Liczba iteracji	Czas zbieżności (ms)	Liczba iteracji	Czas zbieżności (ms)	Liczba iteracji
3x3	978	9	492	8	312	7
4x4	2370	22	1061	15	558	12
5x5	4634	31	1937	22	954	18
6x6	8023	38	3128	28	1539	24
7x7	12862	47	4843	35	2341	30
8x8	19611	61	7132	42	3394	36
9x9	27433	69	9988	48	4714	43
10x10	37887	82	13424	55	6322	49

3.7. Przykład ilustrujący zastosowanie modułu

Aby pokazać działanie tej biblioteki w praktyce, Listing 3.31 przedstawia przykładową aplikację w Pythonie, która implementując bibliotekę przedstawioną w tym artykule, wygeneruje losową macierz porównań o rozmiarze 6×6 , a następnie zmniejszy jej współczynnik spójności (CR) do wartości poniżej 0,1, przy użyciu wszystkich algorytmów dostępnych w bibliotece.

Oprócz tego oblicza również wszystkie indeksy oferowane przez bibliotekę dla już zredukowanych macierzy, a wynik jest zapisywany do pliku CSV, co można zobaczyć na Rys. 3.5. Wykorzystuje w tym celu funkcję pomocniczą `save_matrices_to_csv()`, której implementacja została pominięta na Listingu 3.31 ze względu na jej nieistotność w niniejszej prezentacji. Funkcja pomocnicza to również `multiple_append()` i wraz

3.8.1. Potencjalny wpływ na rozwój małych i średnich przedsiębiorstw, w badaniach i zastosowania przemysłowe modułu

Pojawienie się darmowej biblioteki redukcji niespójności porównań parami wraz z narzędziami przedstawionymi w niniejszym artykule oraz fakt, że aplikacje te są w stanie efektywnie liczyć najważniejsze wskaźniki niespójności macierzy, może umożliwić małym i średnim przedsiębiorstwom wykorzystanie możliwości wielokryterialnych metod podejmowania decyzji - do tej pory musiały one wykorzystywać w tym celu albo uciążliwe i czasochłonne arkusze kalkulacyjne, albo płatne oprogramowanie, takie jak Expert Choice [52]. Wdrażanie metod decyzyjnych jest również ważną częścią przejścia do Przemysłu 4.0 [53], który z perspektywy dzisiejszych zastosowań przyniósł już wymierne efekty również w zastosowaniach przemysłowych [54, 55]. Należy pamiętać, że z naukowego punktu widzenia, redukcja niespójności w elementach macierzy porównań parami jest ważnym elementem badań w logice rozmytej [56], programowaniu preferencji [57], czy konstruktywnych spójnych aproksymacjach [54].

3.9. Wnioski i kierunek dalszego rozwoju

W wykonanych pracach badawczych i rozwojowych po raz pierwszy przygotowano bibliotekę Pythona do redukcji niespójności macierzy porównań parami - `reduce.py` - dostępną publicznie i bez dodatkowych opłat. Biblioteka ta może pomóc w ulepszeniu badań w takich tematach jak AHP, BWM lub inne metody wspomagania decyzji, które stosują macierze porównań parami. Użytkownicy mogą korzystać z biblioteki albo importując ją samodzielnie do swojego projektu Python, albo korzystając z narzędzi opartych na bibliotece. Zaprezentowane wyniki pokazują, że rozwiązanie to jest bardziej efektywne niż inne ogólnodostępne narzędzia dla naukowców i MŚP, w postaci arkuszy kalkulacyjnych takich jak LibreOffice czy Excel, a dzięki łatwości zastosowania i prostocie języka Python, podmioty takie nie będą zmuszone do korzystania z płatnych rozwiązań takich jak Expert Choice, aby skorzystać z możliwości tych metod.

Chociaż biblioteka wykorzystuje obecnie najskuteczniejsze algorytmy dostępne w tym momencie (2023 r.), będzie ona stale rozwijana przez autorów o inne nowe pojawiające się algorytmy, o ile zostaną one wystarczająco przetestowane pod kątem skuteczności i wydajności. Spośród algorytmów nieiteracyjnych warto wspomnieć o kilku z nich: [32-42], który stanie się częścią modułu REDUCE w następnej planowanej wersji, ponieważ projekt jest nadal w fazie rozwoju.

4. REDUCE (reduce.prz.edu.pl): internetowe narzędzie wspomaganie decyzji oraz redukcji niespójności w multiplikatywnych porównaniach parami

Słowa kluczowe rozdziału:

wspomaganie podejmowania decyzji, porównania parami, redukcja niespójności, małe/średnie przedsiębiorstwa, Przemysł 4.0

Abstrakt rozdziału (za oryginalną publikacją):

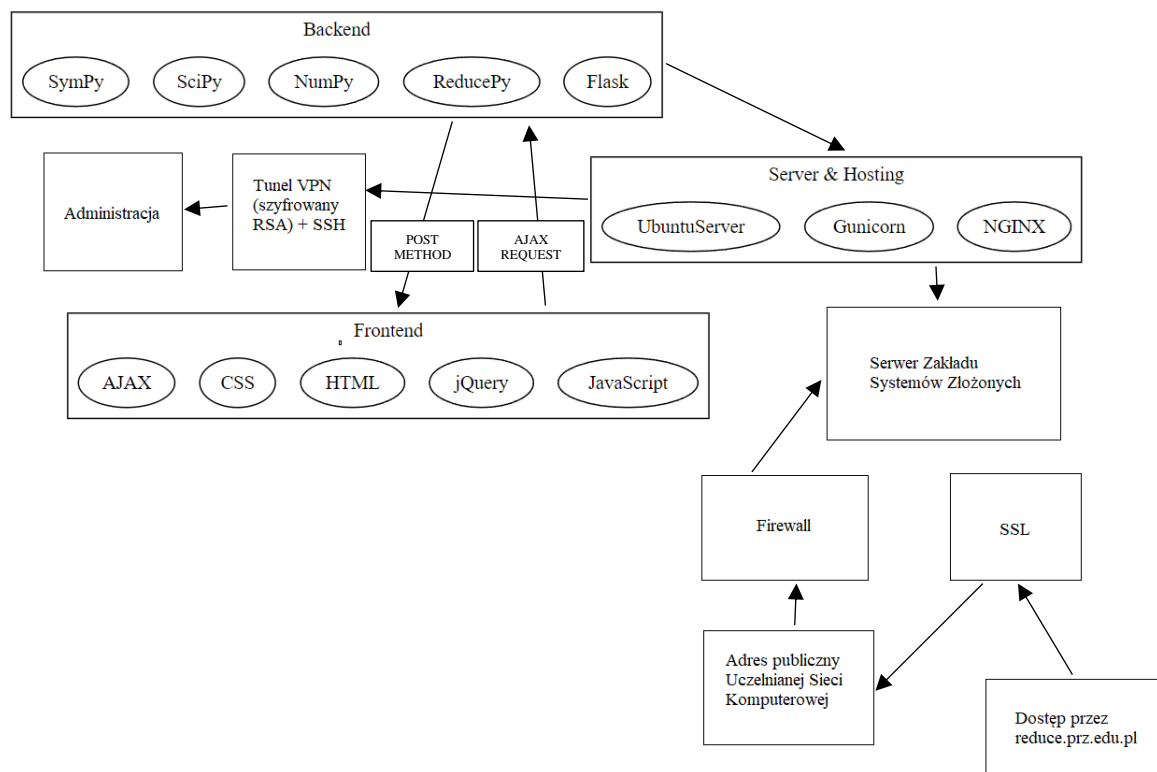
Celem publikacji jest przedstawienie REDUCE, bezpłatnego internetowego narzędzia wspomagającego podejmowanie decyzji w celu zmniejszenia niespójności w ramach multiplikatywnych porównań parami. Porównania parami (PC) stanowią popularną technikę wielokryterialnego podejmowania decyzji, której celem jest przypisanie wag porównywanym podmiotom, zapewniając w ten sposób ich ranking od najlepszego do najgorszego. Jednak porównania parami dostarczane przez ludzkich ekspertów są często niespójne. Niskie poziomy niespójności, poniżej $CR = 0,10$, gdzie CR oznacza współczynnik spójności Saaty'ego, są zwykle tolerowane. Gdy niespójność jest wyższa, komputery PC muszą zostać skorygowane. Może to stanowić problem w praktyce, ponieważ ekspert dostarczający oryginalne preferencje może nie być dostępny lub istnieją inne przeszkody uniemożliwiające rewizję. REDUCE to autonomiczne narzędzie do redukcji niespójności oparte na kilku algorytmach, które nie wymaga pomocy eksperta. Porównania parami (kompletna macierz PC) stanowią dane wejściowe REDUCE. Następnie użytkownik określa, który algorytm ma zostać zastosowany i jaki poziom niespójności ma zostać osiągnięty. Macierz PC, która jest wystarczająco spójna i zachowuje oryginalne preferencje w maksymalnym stopniu, stanowi wynik REDUCE. Co więcej, REDUCE zapewnia również obliczanie wag porównywanych obiektów, dzięki czemu może być również w pewnym stopniu wykorzystywany w procesie hierarchii analitycznej (AHP). Uważamy, że REDUCE może przyczynić się do lepszego podejmowania decyzji w porównaniach parami, ponieważ małe i średnie przedsiębiorstwa zazwyczaj nie mogą sobie pozwolić na drogie oprogramowanie komercyjne i nie zatrudniają pełnoetatowych ekspertów w podejmowaniu decyzji, ponieważ polegają na doświadczeniu swoich pracowników i bezpłatnych zasobach internetowych. Jest to również część architektury rozwoju innowacyjnych systemów komputerowych działających zgodnie z ideą Przemysłu 4.0 z metodami, algorytmami, wbudowanymi podsystemami i dedykowanym oprogramowaniem, które umożliwia wdrażanie inteligentnych systemów dla przedsiębiorstw i procesów.

4.1. Wstęp

Do chwili utworzenia tego narzędzia, nie było znane żadne darmowe (niekomercyjne) narzędzie online do redukcji niespójności w porównaniach parami. Niespójność jest wszechobecnym problemem w tej problematyce, ponieważ ludzcy eksperci rzadko są w pełni konsekwentni w wyrażaniu swoich ocen, zwłaszcza gdy liczba porównywanych obiektów jest stosunkowo wysoka. Niski poziom niespójności jest zwykle tolerowany, ale gdy niespójność przekracza pewien próg, na przykład $CR = 0,10$, porównania parami muszą zostać zrewidowane. Jednak w wielu sytuacjach rewizja porównań parami przez eksperta może nie być możliwa. W takim przypadku do akcji wkracza odpowiednie oprogramowanie.

Dlatego celem niniejszego rozdziału jest przedstawienie i zademonstrowanie wykorzystania REDUCE, bezpłatnego i internetowego narzędzia zaprojektowanego do redukcji niespójności w ramach PC. Ponieważ małe i średnie przedsiębiorstwa zwykle nie mogą sobie pozwolić na drogie oprogramowanie komercyjne, REDUCE może rozwiązać problem z redukcją niespójności, torując w ten sposób drogę do lepszego i szybszego podejmowania decyzji.

4.1.1 Ogólna struktura i działanie aplikacji



Rys. 4.1. Struktura aplikacji internetowej REDUCE.prz.edu.pl

REDUCE to narzędzie online dostępne pod adresem [1], patrz Rys. 4.1. Opiera się na Flask - frameworku, który może obsługiwać logikę Pythona w aplikacji internetowej [2]. Wybór języka Python był niezbędny ze względu na fakt iż biblioteka REDUCE.py [3], zaprezentowana w rozdziale 3, została stworzona właśnie w tym języku. Po stronie front-end [11] aplikacji zastosowano standardowe [3-7] technologie internetowe - HTML, CSS, JavaScript - z dodatkiem biblioteki jQuery [8], która obsługuje żądania AJAX [10] do usługi Flask [9]. W tzw. back-endzie [11], poza Pythonem, Reduce.py, Flaskiem, wykorzystano kilka dodatkowych pakietów – NumPy [12], SciPy [13] i SymPy [14] (używanych także przez Reduce.py).

Biblioteki te ułatwiły implementację po stronie back-endu algorytmów redukcji, w szczególności przekształceń macierzy oraz obliczeń wektorów własnych i dodatkowych indeksów. Po wykonaniu obliczeń, wyniki zwracane są metodą POST [15] do aplikacji po stronie klienta. Witryna jest hostowana na serwerze Zakładu Systemów Złożonych [16] z systemem Ubuntu Server [17] i zainstalowanymi usługami Gunicorn [19], NGINX [20] i Certbot [21] do zarządzania certyfikatami SSL w ramach usługi Let's Encrypt. Rozwiązanie jest chronione przez firewall Cisco ASA 5505 [23], system detekcji intruzów Snort [18] oraz wirtualną zaporę systemu Ubuntu Server (ufw [22]) oraz udostępnione dla użytkowników sieci Internet za pośrednictwem adresu publicznego w ramach Uczelnianej Sieci Komputerowej [24]. Struktura aplikacji została zilustrowana na Rys 4.1.

Dane wejściowe REDUCE to macierz porównań parami $n \times n$, gdzie $n \leq 10$. Poszczególne porównania parami są ręcznie wstawiane do odpowiednich komórek. Skala porównań jest w przedziale z odwrotnościami, co odpowiada (całkowitej) podstawowej skali Saaty'ego od 1 do 9 [25, 26]. Należy zauważyć, że po redukcji niektóre wpisy macierzy mogą przekroczyć wartość 9. Do redukcji niespójności dostępne są dwa algorytmy: Xu i Wei [27] oraz Szybowskiego [28]. Jednak ten drugi algorytm może być stosowany tylko dla wartości całkowitych (i ich odwrotności) porównań parami, zaś pierwszy obsługuje także wartości rzeczywiste. Ogólnie rzecz biorąc, użytkownik ma następujące dostępne opcje:

- rozmiar macierzy porównań parami,
- próg niespójności CR [26],
- możliwość użycia algorytmu Szybowskiego oraz Xu and Wei (z różnymi wartościami parametru lambda)
- możliwość wygenerowania losowej macierzy porównań parami,
- możliwość zresetowania wartości w macierzy porównań parami.

Wynik REDUCE składa się z macierzy o zredukowanym poziomie niespójności, końcowego wskaźnika spójności CI [26] i współczynnika spójności CR [26], wektora

priorytetów w oraz wszystkich indeksów oferowanych przez bibliotekę Reduce.py (*KI* [29], *GCI* [30], *GWI* [31], *PLI* [32], *TGCI* [33], *REI* [34], *HCI* [35]), patrz Rys. 4.2 i 4.3.



RZESZÓW UNIVERSITY
OF TECHNOLOGY



DEPARTMENT OF
COMPLEX SYSTEMS
RZESZÓW UNIVERSITY OF TECHNOLOGY

REDUCE

A simple online tool for reducing inconsistency in Pairwise Comparison Matrices

IMPORTANT NOTE: If you want to use this tool in your scientific work, please cite the paper in which this tool was presented. The paper is available by clicking on this [LINK](#).

[How to use it?](#)

Size of a matrix:
 Threshold of reducing CR:
 Algorithm for reducing CR:

[Generate random matrix](#)

[Reset matrix](#)

1	1/3	3	6	8	2	9
3	1	2	1/5	6	1/4	1/2
1/3	1/2	1	7	1/6	7	5
1/6	5	1/7	1	4	4	1/2
1/8	1/6	6	1/4	1	6	4
1/2	4	1/7	1/4	1/6	1	1/5
1/9	2	1/5	2	1/4	5	1

CI: 1.1202
 CR: 0.8349
 W: w1: 0.2604(7), w2: 0.1585(6), w3: 0.1583(5), w4: 0.1357(3), w5: 0.1391(4), w6: 0.0676(1), w7: 0.0803(2)
 KI: 0.9940
 GC: 2.4259
 GW: 0.1181
 PLI: 23.6737
 TGCI: 2.4259
 RE: 0.7372
 HC: 1.0995

[Reduce!](#)

Rys. 4.2. Narzędzie REDUCE.prz.edu.pl

1	8	1/7	9
1/8	1	1/5	8
7	5	1	1
1/9	1/8	1	1

CI: 1.0487
 CR: 1.1895
 W: w1: 0.3086(3), w2: 0.1327(2), w3: 0.4734(4), w4: 0.0853(1)
 KI: 0.9841
 GC: 3.3712
 GW: 0.1890
 PLI: 28.3899
 TGCI: 3.3713
 RE: 0.5095
 HC: 0.8147

Reduce!

1 3.5953 0.4071 5.1411
 0.2781 1 0.2416 2.6465
 2.4566 4.1383 1 3.4415
 0.1945 0.3779 0.2906 1

CI: 0.0852
 CR: 0.0966
 W: w1: 0.3186(3), w2: 0.1249(2), w3: 0.48(4), w4: 0.0764(1)
 KI: 0.727520244886144
 GC: 0.3319843736564664
 GW: 0.06875983928970936
 PLI: 1.1080887846738188
 TGCI: 0.3319998462783594
 RE: 0.10355933800742884
 HC: 0.06643085367326274

Rys. 4.3. Narzędzie REDUCE.prz.edu.pl

Powracając do schematu na Rys. 4.1 - w warstwie programistycznej walidacja wpisywanych danych odbywa się po stronie klienta aplikacji - ponieważ użytkownik musi podać wartości z zakresu (0 ... 9), a w przypadku wybrania algorytmu Szybowskiego, gdzie liczby muszą być całkowite, funkcja odpowiedzialna za walidację danych najpierw sprawdza wybrany algorytm. Następnie użytkownik jest bezpośrednio zabezpieczony przed wpisaniem błędnych znaków do komórek danych za pomocą drzewa decyzyjnego reguł. Mechanizmy te zostaną omówione i opisane szczegółowo w kolejnych podrozdziałach.

Po wypełnieniu wszystkich komórek, specjalna funkcja słuchacza wysyła dane do zaplecza aplikacji, aby uzyskać parametry takie jak *CI*, *CR* i kilka innych dla wpisanej macierzy. Funkcja ta aktywuje również przycisk redukcji. W tym momencie dane ze wszystkich wejść formularza macierzy są serializowane i wysyłane za pomocą metody POST w zapytaniu AJAX do serwera. Serwer odbiera serializowane dane i konwertuje je na obiekt macierzy SymPy, z którym wygodnie jest pracować. Po obliczeniu parametrów

serwer zwraca wartości w obiekcie słownika, który jest odczytywany przez klienta i umieszczany na stronie internetowej. Wysyłanie danych po kliknięciu przycisku "REDUCE" działa podobnie, ale serwer zwraca nie tylko parametry, ale także nową zredukowaną macierz - również w obiekcie słownika.

4.1.2 Uzasadnienie wybranych technologii

W procesie wyboru technologii dla projektu REDUCE.prz.edu.pl brano pod uwagę różne aspekty i cechy, które były kluczowe dla spełnienia wymagań projektu. Wybór odpowiednich narzędzi był kluczowy dla zapewnienia wydajności, elastyczności i bezpieczeństwa aplikacji.

Flask został wybrany jako kluczowy framework w projekcie REDUCE.prz.edu.pl, głównie ze względu na jego prostotę i elastyczność [9]. Jako mikroframework, Flask oferuje minimalny zestaw narzędzi i funkcji [36], co pozwala na szybkie tworzenie prostych aplikacji internetowych bez nadmiernego obciążenia [37]. W porównaniu z pełnoprawnym frameworkiem, takim jak Django, Flask daje programistom więcej kontroli nad strukturą i zachowaniem aplikacji [9], umożliwiając dostosowanie do specyficznych wymagań projektu [37]. Wybór Flaska był również spójny z wyborem Pythona jako głównego języka programowania, co ułatwiło integrację z istniejącą biblioteką REDUCE.py [3]. Ostatecznie, wybór Flaska odzwierciedlał potrzebę lekkości [39], elastyczności i skoncentrowania się na kluczowych aspektach logiki aplikacji [39], bez konieczności radzenia sobie z dodatkową złożonością i narzędziami [40], które nie byłyby niezbędne dla projektu. Porównanie cech Flaska oraz Django zestawiono w Tabeli 4.1.

Python został wybrany jako główny język programowania, głównie ze względu na jego wszechstronność i względną [41] odpowiedniość do rozwoju aplikacji internetowych. Istnienie biblioteki REDUCE.py [2], stworzonej w Pythonie, była również kluczowym czynnikiem wpływającym na ten wybór. W porównaniu z R, Python oferuje bogatszy wybór bibliotek [42] i jest bardziej odpowiedni do ogólnego przeznaczenia [43], w tym zastosowań webowych. Język R, znany głównie jako narzędzie do analizy statystycznej [44] i wizualizacji danych [45-48], nie jest tradycyjnie kojarzony z rozwojem aplikacji internetowych. Jednak istnieją pewne zastosowania i narzędzia, które umożliwiają wykorzystanie R w kontekście web developmentu [49-54], tym niemniej, są to rozwiązania niszowe [55] (rozwiązania takie jak Flask czy Django są dość powszechnie wykorzystywane również przy bardzo dużych i powszechnie rozpoznawalnych projektach internetowych, patrz [56-65]). Podsumowane zostało to w Tabeli 4.3.

W projekcie REDUCE.prz.edu.pl wybrano jQuery [8] jako główną bibliotekę do obsługi interakcji front-endu. Wybór ten był wynikiem porównania z Vanilla (a więc pozbawionym rozszerzeń) JavaScript [66] i React [67]. Cechy tych narzędzi zostały zbiorczo przedstawione w Tabeli 4.4. jQuery, będąc prostą i wszechstronną biblioteką, oferuje łatwość użycia [68] i wsparcie dla żądań AJAX [69- 71], co było kluczowe dla projektu. W porównaniu z Vanilla JavaScript, jQuery znacznie upraszcza pisanie kodu [72] i zarządzanie interakcjami z użytkownikiem [73].

Tabela 4.1 Porównanie cech frameworków Flask oraz Django

Cecha	Flask	Django
Typ	Mikroframework	Pełnoprawny framework
Elastyczność	Wysoka	Niższa
Wszechstronność	Tak	Nie
Złożoność	Niska	Wyższa
Wbudowane narzędzia	Nie	Tak
Spoleczność	Aktywna	Bardzo aktywna

Tabela 4.2 Porównanie cech systemów Ubuntu Server oraz Windows Server

Cecha	Ubuntu Server	Windows Server
Koszt	Darmowy	Płatny
Bezpieczeństwo	Wysokie	Zależy od konfiguracji
Wsparcie	Spoleczność	Komercyjne
Wydajność	Wysoka	Zależy od konfiguracji
Łatwość zarządzania	Średnia	Wyższa

Tabela 4.3 Porównanie cech języków Python i R

Cecha	Python	R
Wszechstronność	Tak	Nie
Specjalizacja	Ogólnego przeznaczenia	Analiza statystyczna
Biblioteki	Bogaty wybór	Ograniczone
Wydajność	Wysoka	Wysoka w statystyce
Spoleczność	Ogromna	Skoncentrowana na statystyce

Tabela 4.4 Porównanie cech jQuery, Vanilla JavaScript oraz React

Cecha	jQuery	Vanilla JavaScript	React
Prostota	Tak	Nie	Nie
Komponenty	Nie	Nie	Tak
Zależności	Tak	Nie	Tak
Wsparcie dla AJAX	Tak	Trudniejsze	Możliwe z dodatkami
Spoleczność	Duża	Uniwersalna	Bardzo aktywna

Tabela 4.5 Porównanie cech NGINX i Apache

Cecha	NGINX	Apache
Wydajność	Wysoka	Niższa
Konfiguracja	Elastyczna	Mniej elastyczna
Wsparcie	Aktywne	Silne wsparcie społeczności
Skalowalność	Wysoka	Średnia
Bezpieczeństwo	Wysokie	Dobrze ugruntowane

Tabela 4.6 Porównanie Let's Encrypt oraz płatnych certyfikatów SSL

Cecha	Certbot (Let's Encrypt)	Płatne certyfikaty SSL
Koszt	Darmowy	Płatny
Automatyzacja	Tak	Zależy od dostawcy
Dodatkowe funkcje	Nie	Tak
Wsparcie	Społeczność	Komercyjne
Bezpieczeństwo	Wysokie	Zależy od dostawcy

Choć Vanilla JavaScript oferuje pełną kontrolę nad kodem [74], jego użycie mogłoby znacznie zwiększyć złożoność projektu [75]. React, z kolei, jest zaawansowanym [76] frameworkiem do budowy interfejsów użytkownika, który oferuje komponenty i zarządzanie stanem. Jednak jego zastosowanie w projekcie mogłoby być nadmiernie skomplikowane [77], biorąc pod uwagę wymagania i zakres projektu. Ostatecznie, wybór jQuery był zgodny z potrzebą prostoty i elastyczności [78], umożliwiając szybkie i wydajne tworzenie interfejsu użytkownika. Wybór ten również uwzględniał fakt, że jQuery jest dobrze udokumentowany [79] i ma dużą społeczność [80], co może ułatwić rozwój i wsparcie projektu w przyszłości.

Wybór NGINX [20, 81] jako serwera internetowego w projekcie REDUCE.prz.edu.pl był strategicznym i przemyślanym ruchem, mając na uwadze specyficzne wymagania i cele projektu. NGINX, znany ze swojej wysokiej wydajności [82] i niezawodności [83], oferuje doskonałą skalowalność [84] i elastyczność konfiguracji [85].

Jego zdolność do obsługi dużej liczby jednoczesnych połączeń przy niskim zużyciu zasobów [86] uczyniła go idealnym wyborem dla projektu, który wymagał stabilnego i wydajnego serwera. W porównaniu z innymi serwerami, takimi jak Apache [87], NGINX zapewnia lepszą kontrolę nad przetwarzaniem żądań [88] i oferuje nowoczesne funkcje [89], takie jak wsparcie dla protokołu HTTP/2 [90]. Ostateczny wybór NGINX odzwierciedlał dążenie do optymalizacji wydajności [91,95] i bezpieczeństwa [92-95], zapewniając jednocześnie elastyczność [95] niezbędną do spełnienia unikalnych wymagań projektu i został przedstawiony w Tabeli 4.5.

Gunicorn [19] wydawał się oczywistym (poprzez analizę używanych narzędzi w projektach badawczych [96-101]) wyborem jako serwer aplikacji w połączeniu z NGINX i Flaskiem w projekcie REDUCE.prz.edu.pl. Jest serwerem WSGI HTTP dla Pythona, który jest szczególnie dobrze dostosowany do obsługi aplikacji Flask [102]. Jego integracja z NGINX zapewnia wydajne zarządzanie połączeniami i równoważenie obciążenia [103], co jest kluczowe dla zapewnienia płynnej i responsywnej obsługi użytkowników [104]. Gunicorn jest również znany ze swojej prostoty konfiguracji i obsługi [105], co czyni go atrakcyjnym wyborem dla deweloperów.

Wybór Ubuntu Server [17] jako systemu operacyjnego dla serwera hostującego projekt REDUCE.prz.edu.pl był wynikiem starannego rozważenia różnych opcji i wymagań projektu. Ubuntu Server jest znane ze swojej stabilności [106], bezpieczeństwa [107-109] i wsparcia społeczności [110-114], co czyni go popularnym wyborem wśród deweloperów i administratorów systemów. W porównaniu z alternatywami, takimi jak Windows Server, Ubuntu oferuje większą elastyczność i kontrolę nad środowiskiem [115-117], a także lepszą integrację z otwartoźródłowymi narzędziami i technologiami, takimi jak NGINX [118], Gunicorn [97] i Python [119]. Dodatkowo, Ubuntu Server jest dostępne bezpłatnie, co może być atrakcyjne z perspektywy kosztów [120]. Przedstawia to Tabela 4.2.

Certbot [21], narzędzie do zarządzania certyfikatami SSL w ramach usługi Let's Encrypt, zostało wybrane ze względu na jego darmowy charakter i automatyzację [121-130]. W porównaniu z płatnymi certyfikatami SSL, Certbot oferuje wysokie bezpieczeństwo bez dodatkowych kosztów, co jest widoczne w Tabeli 4.6.

Decyzja o zastosowaniu firewalla Cisco ASA 5505 [23] w połączeniu z systemem detekcji intruzów Snort [18] oraz wirtualną zaporą systemu Ubuntu Server (ufw [22]) w projekcie REDUCE.prz.edu.pl była świadomym i celowym wyborem, mając na uwadze specyficzne zasoby i środowisko jednostki naukowej, w której pracuje autor rozprawy. Sam Cisco ASA 5505 był uznawany za solidne i niezawodne rozwiązanie zapewniające zaawansowaną ochronę przed różnymi zagrożeniami sieciowymi [131-135], jednak niestety może nie być gotowy na najnowsze rodzaje zagrożeń sieciowych ze względu na postępujący wiek tego rozwiązania sprzętowego. Dlatego w połączeniu ze Snortem, uznawanym za bycie skutecznym systemem detekcji i prewencji intruzów [136-140], oraz ufw, prostą i spełniającą swoje zadanie zaporą systemu Ubuntu [141-145], tworzy to zintegrowany system zabezpieczeń wystarczający do ochrony naszego oprogramowania.

4.2. Analiza budowy aplikacji

Wracając do Rys. 4.1 można przyjąć, że aplikacja REDUCE dzieli się na cztery podstawowe obszary:

- warstwę front-end (HTML + CSS + JS) odpowiadającą za wygląd i obsługę aplikacji oraz walidację danych wejściowych, w tym warstwę pośredniczącą (AJAX) zapewniającą komunikację pomiędzy warstwą front-end i back-end, „zintegrowaną” z front-endem
- warstwę back-end (Flask + biblioteki Python) wykonującą obliczenia po stronie serwera i przekazującą wyniki do warstwy front-end
- rozwiązania serwerowe i administracyjne (Ubuntu Server, NGINX, Gunicorn, Certbot)

Kolejne podrozdziały, będą szczegółowo omawiały poszczególne aspekty budowy oprogramowania REDUCE.

4.2.1. Warstwa front-end

4.2.1.1 Struktura sekcji body kodu HTML

Kod HTML, który zostanie poddany analizie, stanowi strukturę i szkielet narzędzia "REDUCE". Jest to język znaczników, który definiuje elementy i strukturę strony internetowej, umożliwiając przeglądarkom internetowym interpretację i wyświetlanie jej zawartości. Poniżej przedstawiono szczegółowy opis kodu HTML, który obejmuje każdy element i atrybut, wyjaśniając ich znaczenie i rolę w kontekście narzędzia "REDUCE":

- **Główny Kontener** (`<div class="container">`): zawiera całą zawartość strony. Klasa "container" może być używana do stylizacji, takiej jak wyśrodkowanie zawartości lub ustawienie szerokości.
- **Formularz** (`<form id="matrixForm" autocomplete="off">`): Wewnątrz głównego kontenera znajduje się formularz o identyfikatorze `matrixForm`. Choć nie jest obszerny w treść, jest to kluczowy element struktury, który pełni kilka istotnych funkcji:
 - Umieszczenie całej zawartości witryny w formularzu ułatwia serializację danych. Oznacza to, że wszystkie dane wprowadzone przez użytkownika, takie jak wybór rozmiaru macierzy, próg redukcji *CR*, wybór algorytmu itp., mogą być łatwo zebrane i przekształcone w odpowiedni format. Serializacja ta jest niezbędna do przetworzenia danych przez serwer i skrypt po stronie

klienta, co umożliwia dynamiczną interakcję i odpowiedź na żądania użytkownika.

- Atrybut **autocomplete="off"** jest stosowany, aby wyłączyć funkcję automatycznego uzupełniania danych formularza przez przeglądarkę. Jest to szczególnie ważne w narzędziach naukowych i analitycznych, gdzie precyzja i kontrola nad wprowadzanymi danymi są kluczowe. Automatyczne uzupełnianie mogłoby prowadzić do niechcianych błędów lub nieścisłości w danych.
- Umieszczenie wszystkich elementów interaktywnych witryny wewnątrz formularza zapewnia strukturalną integralność. Ułatwia to zarządzanie danymi i interakcjami użytkownika, ponieważ wszystkie elementy wejściowe są zawarte w jednej logicznej jednostce. Ułatwia to także utrzymanie kodu i ewentualne modyfikacje, ponieważ zmiany w logice formularza mogą być wprowadzane w jednym miejscu.
- **Loga i nagłówek** (kod przedstawiony na Listingu 4.1., zaś wygląd na Rys. 4.4)
 - Dwa linki prowadzące do stron Politechniki Rzeszowskiej i Zakładu Systemów Złożonych, każdy z nich zawiera obraz logo. Atrybuty `height`, `style`, `border`, i `alt` są używane do stylizacji i opisu obrazów. Nagłówek "REDUCE" jest przedstawiony jako nagłówek pierwszego poziomu. Krótki opis narzędzia oraz ważna uwaga są zawarte w paragrafach.
- **Instrukcje użycia** (kod znajduje się na Listingu 4.2., zaś wygląd na Rys. 4.5)
 - Ta sekcja zawiera instrukcje, jak korzystać z narzędzia. Paragraf z klasą `"toggle js-toggle"` służy jako przełącznik do wyświetlania i ukrywania listy instrukcji. Lista uporządkowana zawiera szczegółowe instrukcje, w tym zagnieżdżone listy nieuporządkowane, które wyjaśniają różne opcje i kroki.



REDUCE

A simple online tool for reducing inconsistency in Pairwise Comparison Matrices

IMPORTANT NOTE: If you want to use this tool in your scientific work, please cite the paper in which this tool was presented. The paper is available by clicking on this [LINK](#).

Rys. 4.4 Wygląd sekcji loga i nagłówka

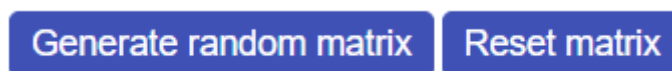
Size of a matrix:

Threshold of reducing CR:

Algorithm for reducing CR:

Rys. 4.5. Wygląd wyboru rozmiaru macierzy, progu redukcji CR oraz wyboru algorytmu redukcji niespójności

- **Wybór rozmiaru macierzy, progu redukcji CR oraz wybór algorytmu redukcji niespójności** (kod znajduje się na Listingu 4.3, zaś wygląd sekcji na Rys. 4.6): Ta sekcja zawiera po pierwsze etykietę i menu rozwijane do wyboru rozmiaru macierzy, gdzie klasa `js-clear-reduced-matrix` będzie używana do interakcji z JavaScriptem, opisaney w dalszej części tego rozdziału. Następnie etykieta oraz menu rozwijane do wyboru progu redukcji CR (interakcja przez tę samą klasę) oraz wyboru algorytmu – Szybowskiego albo Xu & Wei (oprócz poprzedniej klasy, także klasa `js-algorithm`)
- **Przyciski do generowania losowej macierzy i resetowania wpisanych wartości** (wygląd sekcji na Rys. 4.7): Ta sekcja zawiera dwa przyciski: jeden do generowania losowej macierzy, a drugi do resetowania macierzy. Klasy `generateButton` i `resetButton` oraz `js-generate-button` i `js-reset-button` są używane do stylizacji i interakcji z JavaScriptem.



Rys. 4.7. Wygląd przycisków do generowania losowej macierzy oraz resetowania wpisanych wartości

- **Wprowadzanie macierzy** (Rys. 4.8, Rys. 4.9 oraz Listing 4.4.): Macierz jest przedstawiona jako zestaw list ``, gdzie każda lista reprezentuje wiersz macierzy. W każdym wierszu znajduje się wiele elementów ``, które reprezentują komórki macierzy. Każda komórka macierzy zawiera element `<input>`, który umożliwia wprowadzanie danych. Niektóre z tych elementów mają wartość domyślną równą 1 i są tylko do odczytu (atrybut `readonly`), co jest związane z konstrukcją macierzy porównań parami, gdzie elementy na przekątnej są zawsze równe 1. Rozmiar macierzy może być sterowany za pomocą pola "Size of a matrix" (opisanego wcześniej, widoczne na Rys. 4.9). Skrypt

JavaScript może ukrywać lub pokazywać odpowiednie elementy, aby dostosować rozmiar macierzy. Elementy po przekątnej mogą być automatycznie uzupełniane, co będzie opisane później w sekcji dt. funkcji JS - macierz jest symetryczna i wartości po przekątnej są odwrotnością siebie. W funkcjach JS znajdują się również mechanizmy walidacji danych wejściowych do tej macierzy.

1		5				
	1	4	1/3			
1/5	1/4	1				
	3		1			
				1		
					1	
						1

Rys. 4.8 Wygląd sekcji wprowadzania macierzy

Size of a matrix:
 Threshold of reducing CR:
 Algorithm for reducing CR:

1	1/6	1/9	7	6	9	1/3	1/4	1/4	6
6	1	8	3	1/7	1/9	4	1/9	6	8
9	1/8	1	1/6	7	1/7	8	5	1/7	1/7
1/7	1/3	6	1	1/5	1/8	9	3	9	4
1/6	7	1/7	5	1	1/9	1/7	1/4	1/7	1/8
1/9	9	7	8	9	1	6	5	8	5
3	1/4	1/8	1/9	7	1/6	1	1/4	8	1/6
4	9	1/5	1/3	4	1/5	4	1	5	6
4	1/6	7	1/9	7	1/8	1/8	1/5	1	1/3
1/6	1/8	7	1/4	8	1/5	6	1/6	3	1

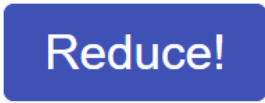
Rys. 4.9 Zmiana rozmiaru wprowadzanej macierzy

- **Indeksy macierzy** (Rys. 4.10 oraz Listing 4.5.): Ta sekcja kodu HTML przedstawia listę różnych miar indeksów związanych z wcześniej wspomnianą macierzą. Każda miara indeksu jest reprezentowana przez skrót (np. "CI", "CR", "W" itd.) i ma odpowiadającą jej klasę, która zaczyna się od "js-". Każda z tych

klas odnosi się do funkcji JavaScript, które będą opisane później w kodzie. Te funkcje będą komunikować się z backendem, aby pobrać odpowiednie wartości dla tej macierzy.

- **Przycisk „Reduce!” oraz kolejne niewidoczne sekcje** (wygląd na Rys. 4.10 oraz kod w Listingu 4.6): Jest to przycisk typu "submit", który ma klasę `reduceButton` oraz `js-reduce-button`. Klasa `js-reduce-button` jest używana do identyfikacji przycisku w skryptach JavaScript. Przycisk staje się aktywny tylko wtedy, gdy spełnione są pewne warunki, takie jak wprowadzenie odpowiednich danych do macierzy, za co odpowiada początkowy atrybut `disabled`. Gdy przycisk jest kliknięty, macierz jest przesyłana do funkcji JavaScript, która następnie przesyła ją do backendu w celu przetworzenia. W kolejnych sekcjach z Listingu 4.6. klasa `js-reduced-matrix` jest używana przez skrypty JavaScript do umieszczania w tym miejscu zredukowanej macierzy oraz związanych z nią parametrów po otrzymaniu odpowiedzi z backendu.

CI: 1.6143
CR: 1.8311
W: w1: 0.177(1), w2: 0.4243(4), w3: 0.2138(3), w4: 0.1849(2)
KI: 0.9974
GC: 4.0849
GW: 0.1568
PLI: 99.7036
TGCI: 4.0845
RE: 0.5959
HC: 0.3340



Reduce!

Rys. 4.10. Parametry wprowadzonej macierzy oraz przycisk Reduce!

4.2.1.2 Atrybuty stylu graficznych CSS

Stylowanie jest kluczowym elementem każdej nowoczesnej aplikacji internetowej [146-149], a CSS jest językiem, który umożliwia tworzenie atrakcyjnych i spójnych interfejsów użytkownika. W przypadku aplikacji REDUCE, zaprezentowane w dalszej części fragmenty kodu CSS odgrywają kluczową rolę w definiowaniu wyglądu i zachowania

różnych elementów strony, od globalnego resetowania stylów po szczegółowe formatowanie poszczególnych komponentów.

Poniżej znajduje się szczegółowy opis poszczególnych klas i grup klas zawartych w tym fragmencie kodu, który wyjaśnia, jak każda z nich przyczynia się do ogólnego wyglądu i funkcjonalności aplikacji REDUCE:

- **Globalne resetowanie stylów** (Listing 4.7): Wszystkie podstawowe elementy HTML (takie jak `html`, `body`, `div`, `span`, `h1`, `h2`, itd.) są zresetowane do wartości domyślnych, takich jak `margin: 0`, `padding: 0`, `border: 0`, itd. Jest to często stosowany zabieg [150-154], który eliminuje różnice między przeglądarkami.
- **Strukturalne elementy blokowe** (Listing 4.8): Klasy takie jak `article`, `aside`, `details`, `figcaption`, itd. są ustawione na `display:block`, co umożliwia lepszą organizację struktury strony.
- **Stylowanie ciała strony** (Listing 4.9): Klasa `body` ma ustawioną wysokość linii na 1, a `html` i `body` mają ustawioną wysokość na 100% oraz określoną czcionkę. Klasa `label` definiuje szerokość i padding, podczas gdy `.container` określa minimalną wysokość, marginesy i padding. Klasy `.footer` i `.push` są odpowiedzialne za stylowanie stopki, w tym wysokość, rozmiar czcionki, kolor i inne właściwości.
- **Stylowanie macierzy porównań parami** (Listing 4.10): Klasy związane z macierzą, takie jak `.matrix`, `.matrix td`, `.matrix__row`, `.matrix__row__item`, itd., definiują stylowanie tabeli macierzy, w tym wysokość, szerokość, wyświetlanie i inne właściwości.
- **Stylowanie tytułów i opisów** (Listing 4.11): Klasy `.title` i `.description` są odpowiedzialne za stylowanie tytułów i opisów, w tym rozmiar czcionki, kolor, waga i padding.
- **Stylowanie przycisków** (Listing 4.12): Klasy związane z przyciskami, takie jak `button`, `.generateButton`, `.resetButton` itd., są odpowiedzialne za stylowanie przycisków, w tym kolor tła, rozmiar czcionki, marginesy, efekty hover i focus oraz inne właściwości.
- **Stylowanie list oraz przełączników** (Listing 4.13): Klasy `.list` i `.list li` są odpowiedzialne za stylowanie list, w tym styl listy, padding i kolor tła. Klasa `.toggle` jest odpowiedzialna za stylowanie przełącznika, w tym margines, kursor, padding, kolor i efekty przejścia.

Te klasy i grupy klas współpracują ze sobą, aby stworzyć spójny i atrakcyjny interfejs użytkownika dla aplikacji REDUCE, zapewniając jednolitość i spójność w całym designie.

4.2.1.3 Skrypty JavaScript oraz jQuery

Funkcje języka JavaScript (oraz jQuery) przedstawione w tym podrozdziale (i jego kolejnych podsekcjach) są kluczowym elementem narzędzia REDUCE. Obejmują zarówno interakcje z użytkownikiem na poziomie interfejsu, jak i komunikację z backendem w celu przetwarzania danych. Wykorzystuje różne techniki, takie jak manipulacja DOM, żądania AJAX, walidacja danych i obsługa zdarzeń, aby zapewnić płynne i efektywne działanie narzędzia.

Funkcje takie jak `setMatrixVisibility()`, `setMatrixSize()`, `setMatrixValue()` służą do manipulacji wyglądem i wartościami macierzy na stronie. Pozwalają na dynamiczną zmianę rozmiaru macierzy, ustawianie wartości i kontrolowanie widoczności elementów. `generateRandomMatrix()` i `resetMatrix()` to funkcje, które umożliwiają użytkownikowi generowanie losowych wartości macierzy oraz resetowanie jej do stanu początkowego. Funkcje takie jak `checkMatrix()` i `checkMatrixItem()` zapewniają, że wprowadzone przez użytkownika dane są zgodne z oczekiwanym formatem i ograniczeniami. Funkcja `getCICR()` jest odpowiedzialna za wysyłanie żądania AJAX do backendu w celu obliczenia wskaźników spójności (*CI*) i współczynnika spójności (*CR*) dla danej macierzy. Funkcja `sendMatrix()` wysyła macierz do backendu, gdzie jest przetwarzana w celu redukcji niespójności. Wynik jest następnie ustawiany na stronie za pomocą funkcji `setNewMatrix()`.

W sekcji `$(document).ready` znajdują się zdarzenia, które reagują na różne interakcje użytkownika, takie jak zmiana rozmiaru macierzy, generowanie losowej macierzy, resetowanie macierzy, wysyłanie macierzy do redukcji itp. Kod zawiera również logikę odpowiedzialną za dynamiczną zmianę interfejsu użytkownika w odpowiedzi na różne akcje, takie jak wyświetlanie i ukrywanie instrukcji, blokowanie i odblokowywanie przycisków itp. Użytkownik może wybrać różne algorytmy do redukcji niespójności, takie jak "Xu and Wei" z różnymi wartościami parametru λ lub "Szybowski".

4.2.1.3.1 Funkcja `setMatrixVisibility()`

Funkcja `setMatrixVisibility()` (widoczna na Listingu 4.14) jest odpowiedzialna za kontrolowanie widoczności poszczególnych elementów macierzy na stronie. Jest to kluczowa część interakcji z użytkownikiem, która pozwala na dynamiczną zmianę wyglądu macierzy w zależności od wybranego rozmiaru. Jej parametry to:

- **item** - jest to obiekt jQuery reprezentujący pojedynczy element macierzy, który ma zostać ukryty lub pokazany. Element ten ma atrybut `name`, który zawiera informacje o jego położeniu w macierzy (numer wiersza i kolumny).
- **size** - jest to liczba całkowita reprezentująca rozmiar macierzy, który ma zostać wyświetlony. Wszystkie elementy macierzy o numerach wierszy i kolumn większych niż `size` zostaną ukryte.

Za pomocą metody `item.attr('name')` pobierana jest wartość atrybutu `name` dla danego elementu. Ta wartość zawiera informacje o położeniu elementu w macierzy (numer wiersza i kolumny). Wywołanie funkcji `getParam(name)` z nazwą elementu jako argumentem zwraca obiekt z numerami wiersza i kolumny. Ta funkcja analizuje wartość atrybutu `name` i ekstrahuje z niej te informacje. Metoda `item.parent().removeClass('show')` usuwa klasę CSS o nazwie `show` z rodzica danego elementu. Dzięki temu element zostaje ukryty na stronie. Warunek `parseInt(param.row) <= size && parseInt(param.column) <= size` sprawdza, czy numer wiersza i kolumny danego elementu jest mniejszy lub równy rozmiarowi macierzy, który ma zostać wyświetlony. Jeśli tak, element jest w zakresie i powinien zostać pokazany. Jeśli warunek rozmiaru jest spełniony, metoda `item.parent().addClass('show')` dodaje klasę CSS o nazwie `show` do rodzica danego elementu. Dzięki temu element zostaje pokazany na stronie.

4.2.1.3.2 Funkcja `setMatrixSize()`

Funkcja `setMatrixSize()` (Listing 4.15) jest odpowiedzialna za ustawienie rozmiaru macierzy na stronie. Jest to część większego systemu, który pozwala użytkownikowi na dynamiczną interakcję z macierzą. Pozwala na dynamiczną zmianę rozmiaru macierzy, ukrywając lub pokazując poszczególne elementy w zależności od wybranego rozmiaru. Jest to osiągnięte przez połączenie z poprzednio opisywaną funkcją, `setMatrixVisibility()`, która kontroluje widoczność poszczególnych elementów. Funkcja ta jest wywoływana, gdy użytkownik zmienia rozmiar macierzy za pomocą interfejsu użytkownika, poprzez wybór opcji z listy rozwijanej.

Za pomocą metody jQuery `$('#matrixSize').val()` pobierana jest wartość elementu o identyfikatorze `matrixSize`. Zakłada się, że jest to element formularza (na przykład pole wyboru), który pozwala użytkownikowi na wybór rozmiaru macierzy. Wartość ta jest przechowywana w zmiennej `size`. Za pomocą selektora jQuery `$('.js-matrix-item')` pobierane są wszystkie elementy macierzy, które mają klasę `js-matrix-item`. Zwrócone elementy są przechowywane w zmiennej `items`. Metoda

`items.each(function(){ ... })` iteruje przez wszystkie elementy macierzy. Dla każdego elementu wykonywane są następujące kroki:

- Dany element jest konwertowany na obiekt jQuery za pomocą `$(this)`, co pozwala na wykorzystanie metod jQuery na tym elemencie. Wynik jest przechowywany w zmiennej `item`.
- Wywołanie funkcji `setMatrixVisibility(item, parseInt(size))` ustawia widoczność danego elementu macierzy. Funkcja `setMatrixVisibility()` została opisana wcześniej i kontroluje, czy dany element powinien być widoczny czy ukryty, w zależności od rozmiaru macierzy. Rozmiar macierzy jest konwertowany na liczbę całkowitą za pomocą `parseInt(size)`.

4.2.1.3.3 Funkcja `getParam()`

Funkcja `getParam()` (widoczna na Listing 4.16) jest funkcją pomocniczą, która analizuje specjalnie sformatowany ciąg znaków reprezentujący nazwę i zwraca obiekt zawierający numery wiersza i kolumny. Jest to kluczowa funkcja pomocnicza, która jest wykorzystywana w innych częściach kodu do manipulowania macierzą. Dzięki niej możliwe jest dynamiczne zarządzanie macierzą na stronie, na przykład zmiana jej rozmiaru czy ustawienie wartości poszczególnych elementów. Na przykład, jeśli funkcja zostanie wywołana z parametrem `name` równym `matrix[3][5]`, zwróci obiekt `{ row: '3', column: '5' }`.

Funkcja jako parametr przyjmuje `name` - ciąg znaków, który ma zostać przeanalizowany. Ma on specjalny format, w którym numery wiersza i kolumny są umieszczone w nawiasach kwadratowych, na przykład: `"matrix[3][5]"`. Za pomocą metody `indexOf('][')` znajduje się indeks pierwszego wystąpienia separatora `']['` w ciągu znaków `name`. Wynik jest przechowywany w zmiennej `temp`.

Za pomocą metody `substring` wyciąga się numer wiersza z ciągu znaków `name`. Wykorzystuje się do tego indeks otwierającego nawiasu kwadratowego oraz indeks separatora `temp`. Wynik jest przechowywany w zmiennej `row`.

Podobnie jak w przypadku numeru wiersza, numer kolumny jest wyciągany za pomocą metody `substring`, ale tym razem wykorzystuje się indeks separatora `temp` oraz długość ciągu znaków `name`. Wynik jest przechowywany w zmiennej `column`. Na koniec funkcja zwraca obiekt zawierający numery wiersza i kolumny w postaci `{ row: row, column: column }`.

4.2.1.3.4 Funkcja *setMatrixValue()*

Funkcja `setMatrixValue()` (Listing 4.17) jest kluczową częścią kodu odpowiedzialną za manipulację wartościami w macierzy. Działa w zgodzie z określonymi regułami i algorytmami dla macierzy porównań parami, zapewniając spójność i poprawność danych w macierzy. Jest wykorzystywana w różnych częściach kodu do dynamicznego zarządzania macierzą na stronie.

Jako parametr funkcja ta przyjmuje `item` - obiekt jQuery reprezentujący element macierzy, którego wartość ma zostać ustawiona. Funkcja pobiera atrybut `name` i wartość `value` z elementu `item` za pomocą metod jQuery `attr` i `val`. Wywołuje funkcję pomocniczą `getParam` z nazwą elementu, aby uzyskać numery wiersza i kolumny. Wynik jest przechowywany w zmiennej `param`.

Na podstawie numerów wiersza i kolumny znajduje odpowiedni element macierzy, który ma zostać zaktualizowany. Wynik jest przechowywany w zmiennej `itemInv`. Następnie funkcja pobiera wartość wybranego algorytmu z elementu o klasie `.js_algorithm`. Wynik jest przechowywany w zmiennej `algorithm`. W kolejnym kroku następuje sprawdzenie, czy wybrany algorytm nie jest równy '3'. Jeśli tak, to przypisuje wartość `true` do zmiennej `allowDec`, w przeciwnym razie `false` (dzieje się tak ponieważ tylko funkcja Xu and Wei jest w stanie obsłużyć wartości rzeczywiste).

Na podstawie wartości `value` oblicza się wynik, który ma zostać ustawiony dla elementu `itemInv`. Proces ten obejmuje kilka warunków:

- Jeśli `value` jest puste, wynik jest pusty.
- Jeśli `value` ma długość 1, wynik jest równy '1' lub '1/' połączone z wartością.
- Jeśli `value` zawiera '/', wynik jest równy wartości po '/'.
- Jeśli `value` zawiera '.', oblicza odwrotność wartości i formatuje ją zgodnie z określonymi regułami.

Na samym końcu funkcja ustawia wartość `result` dla elementu `itemInv` za pomocą metody jQuery `val`. Jako przykład można podać sytuację w której jeśli funkcja zostanie wywołana z elementem o nazwie `matrix[3][5]` i wartości "2", a wybrany algorytm nie będzie równy '3', wynik będzie równy '1/2', a odpowiedni element macierzy zostanie zaktualizowany tą wartością.

4.2.1.3.5 Funkcja *generateRandomMatrix()*

Funkcja `generateRandomMatrix()` (Listing 4.18) jest odpowiedzialna za generowanie losowej macierzy porównań parami, wypełniając ją określonymi wartościami

według skali Saaty'ego. Na przykład, jeśli macierz zawiera 9 x 9 elementów, a wszystkie są zapisywalne (nie mają atrybutu `readonly`), funkcja wypełni macierz losowymi wartościami z tablicy `values` i zaktualizuje stan przycisku „Reduce!” na aktywny. Funkcja ta pobiera wszystkie elementy o klasie `.js-matrix-item` i przechowuje je w zmiennej `item`. Tworzy tablicę `values`, która zawiera zestaw wartości, które mogą zostać użyte do wypełnienia macierzy.

Następnie ustawia puste wartości dla wszystkich elementów o klasie `.js-matrix-item`, które nie mają atrybutu `readonly`. Jest to krok przygotowawczy przed wypełnieniem macierzy nowymi wartościami. Następnie funkcja:

- Przechodzi przez wszystkie elementy w zmiennej `item`, używając metody jQuery `each`.
- Dla każdego elementu sprawdza, czy jego wartość jest pusta i czy nie ma atrybutu `readonly`.
- Jeśli oba warunki są spełnione, losuje wartość z tablicy `values` i ustawia ją dla bieżącego elementu.
- Wywołuje funkcję `setMatrixValue` z bieżącym elementem, aby zaktualizować wartość zgodnie z określonymi regułami.

Na samym końcu wywołuje się funkcję `setReduceButton`, aby zaktualizować stan przycisku „Reduce!” na podstawie nowo wygenerowanej macierzy.

4.2.1.3.6 Funkcja `resetMatrix()`

Funkcja `resetMatrix()` (Listing 4.19) służy do zresetowania stanu macierzy oraz powiązanych z nią elementów interfejsu użytkownika. Dla wszystkich elementów o klasie `.js-matrix-item`, które nie mają atrybutu `readonly`, ustawia puste wartości. Jest to odpowiednik wyczyszczenia wszystkich zapisywalnych komórek macierzy. Ustawia też pusty tekst dla różnych elementów, które są zidentyfikowane przez swoje klasy CSS. Te elementy mogą reprezentować różne wskaźniki, wartości lub metryki związane z macierzą, takie jak *CI* (Consistency Index), *CR* (Consistency Ratio), wektor priorytetów i inne.

W praktyce oznacza to, że wszelkie informacje, które mogły być wcześniej wyświetlane w tych elementach, zostaną usunięte, a te elementy zostaną zresetowane do stanu początkowego.

4.2.1.3.7 Funkcja `setReduceButton()`

Funkcja `setReduceButton()` (Listing 4.20) jest odpowiedzialna za zarządzanie stanem przycisku redukcji oraz powiązanimi z nim elementami interfejsu użytkownika

w kontekście macierzy. Jeśli macierz jest kompletna i wszystkie wymagane komórki mają wartości, przycisk redukcji zostanie włączony, a funkcja `getCICR()` (opisana w następnym podrozdziale) zostanie wywołana. Jeśli jakkolwiek wymagana komórka jest pusta, przycisk redukcji zostanie wyłączony, a powiązane elementy interfejsu użytkownika zostaną zresetowane. Funkcja `setReduceButton()` odgrywa kluczową rolę w zarządzaniu interakcjami użytkownika z macierzą, kontrolując, czy przycisk redukcji może być użyty, oraz aktualizując powiązane elementy interfejsu użytkownika. Dzięki temu zapewnia spójność i odpowiednią reakcję interfejsu użytkownika na różne stany macierzy.

Funkcja rozpoczyna od sprawdzenia stanu macierzy - zmienna `disabled` jest zainicjowana jako `false`, po czym przechodzi przez wszystkie elementy o klasie `.js-matrix-item`, sprawdzając, czy ich rodzic ma klasę `show`. Jeśli rodzic ma klasę `show`, a wartość elementu jest pusta (`'`), zmienna `disabled` zostaje ustawiona na `true`, a pętla zostaje przerwana. Po tym funkcja ustawia właściwość `disabled` przycisku o klasie `.js-reduce-button` na wartość zmiennej `disabled`. Jeśli `disabled` jest prawdziwe, przycisk zostaje wyłączony; w przeciwnym razie zostaje włączony.

Jeśli przycisk nie jest wyłączony (`disabled` jest fałszywe), wywoływana jest funkcja `getCICR()`, która może obliczać różne wskaźniki związane z macierzą. Jeśli przycisk jest wyłączony, wtedy różne elementy interfejsu użytkownika, takie jak `.js-CI`, `.js-CR`, `.js-EV`, itp., są resetowane do stanu początkowego, a ich tekst zostaje wyczyszczony. Ponadto, element o klasie `.js-reduced-matrix` ma wyczyszczony swój kod HTML.

4.2.1.3.8 Funkcja `getCICR()`

Funkcja `getCICR()` jest odpowiedzialna za wywołanie asynchronicznego żądania do serwera w celu obliczenia pewnych wartości związanych z macierzą, takich jak *CI* (Consistency Index) i *CR* (Consistency Ratio), oraz aktualizacji odpowiednich elementów interfejsu użytkownika na podstawie odpowiedzi serwera.

Ta funkcja pobiera formularz o identyfikatorze `#matrixForm` oraz serializuje dane formularza do formatu, który może być przesłany przez żądanie HTTP, i przypisuje je do zmiennej `serializedData`. Następnie wywołuje asynchroniczne żądanie HTTP typu POST na adres URL `/computeCI_CR` (na który odpowiada funkcja back-endu, opisana w jednym z kolejnych podrozdziałów), przekazując zserializowane dane formularza jako dane żądania. W przypadku sukcesu (funkcja `success`) dzieją się kolejno następujące akcje:

- Iteruje przez odpowiedź serwera, która jest obiektem zawierającym różne wartości.
- Dla każdego klucza (indeksu) i wartości w odpowiedzi:

- Jeśli klucz nie jest równy 'EV', szuka elementu o klasie `.js-{index}`.
- Jeśli element istnieje, formatuje wartość do 4 miejsc po przecinku (jeśli wartość nie jest niezdefiniowana ani pusta) i ustawia ją jako HTML elementu.
- Jeśli klucz jest równy EV, ustawia wartość jako tekst elementu o klasie `.js_EV`.

W przypadku błędu funkcja `error` wyświetla błąd w konsoli przeglądarki.

4.2.1.3.9 Funkcja `sendMatrix()`

Funkcja `sendMatrix()` (zaprezentowana na Listingu 4.22) służy do komunikacji z serwerem w celu przetworzenia danych macierzy. Wykorzystuje technologię AJAX do asynchronicznego wysłania danych do serwera i odbioru odpowiedzi. Sukces żądania prowadzi do wywołania innej funkcji (opisanej w kolejnej podsekcji), która może dalej przetworzyć odpowiedź, podczas gdy błąd jest rejestrowany w konsoli. Ta funkcja jest kluczowym elementem interakcji między klientem a serwerem w kontekście operacji na macierzy.

Funkcja ta pobiera formularz o identyfikatorze `#matrixForm`. Serializuje dane formularza do formatu, który może być przesłany przez żądanie HTTP, i przypisuje je do zmiennej `serializedData`. Następnie wywołuje asynchroniczne żądanie HTTP typu POST na adres URL `/reduceButton`, przekazując zserializowane dane formularza jako dane żądania. W przypadku sukcesu (funkcja `success`), wywołuje się funkcję `setNewMatrix(response)`, przekazując odpowiedź serwera jako argument. Ta funkcja jest odpowiedzialna za dalsze przetworzenie odpowiedzi i aktualizację interfejsu użytkownika, zaś w przypadku błędu wyświetlany jest błąd w konsoli przeglądarki.

4.2.1.3.10 Funkcja `setNewMatrix`

Funkcja `setNewMatrix(data)` jest odpowiedzialna za ustawienie nowej macierzy i wyświetlenie jej wraz z dodatkowymi informacjami na stronie internetowej. Zaprezentowano ją na Listingu 4.23. Metoda `setNewMatrix(data)` przetwarza dane macierzy i dodatkowe wartości, a następnie aktualizuje interfejs użytkownika, wyświetlając nową macierz w formie tabeli oraz dodatkowe informacje w formie tekstowej.

Jako parametr funkcja przyjmuje obiekt `data` zawierający dane macierzy oraz dodatkowe wartości, takie jak `CI`, `CR`, `W`, `KI` itp. Następnie funkcja przechodzi do inicjalizacji danych:

- Pobiera element o klasie `.js-reduced-matrix` i przypisuje go do zmiennej `wrapper`.

- Pobiera wartość rozmiaru macierzy z elementu o identyfikatorze `#matrixSize` i przypisuje ją do zmiennej `size`.
- Oblicza wartość `temp` jako kwadrat rozmiaru macierzy.
- Inicjalizuje pusty ciąg znaków `matrix`, który będzie przechowywał kod HTML nowej macierzy.

Następnie funkcja iteruje przez obiekt `data`, przetwarzając każdą parę klucz-wartość, a jeśli indeks jest mniejszy lub równy `temp`, przetwarza tą wartość jako element macierzy:

- Oblicza tagi otwierające i zamykające wiersza tabeli (`<tr>` i `</tr>`) na podstawie indeksu.
- Formatuje wartość do 4 miejsc po przecinku, jeśli nie jest liczbą całkowitą.
- Dodaje komórkę tabeli (`<td>`) z wartością do zmiennej `matrix`.

Pobiera dodatkowe wartości z obiektu `data`, takie jak `CI`, `CR`, `W`, `KI` itp., i przypisuje je do odpowiednich zmiennych i ustawia kod HTML w elemencie `wrapper`, tworząc tabelę z macierzą i dodając paragraf z dodatkowymi wartościami.

4.2.1.3.11 Funkcja `checkMatrix()`

Funkcja `checkMatrix()` jest używana do sprawdzenia, czy wartości w macierzy są zgodne z wybranym algorytmem (zaprezentowana została na Listingu 4.24). Jeśli algorytm nie dopuszcza wartości zmiennoprzecinkowych, funkcja usuwa takie wartości z macierzy i aktualizuje interfejs użytkownika. Jest to niezbędne w kontekście zapewnienia, że dane wejściowe są zgodne z oczekiwaniami wybranego algorytmu. Funkcja sprawdza wartość wybranego algorytmu z elementu o klasie `.js-algorithm` i przypisuje ją do zmiennej `algorithm`.

Sprawdza, czy algorytm nie dopuszcza wartości zmiennoprzecinkowych (jeśli `algorithm` jest różne od '3', a więc wartości przypisanej algorytmowi Szybowskiego) i przypisuje wynik do zmiennej `allowDec`. Jeśli `allowDec` jest fałszywe, oznacza to, że algorytm nie dopuszcza wartości zmiennoprzecinkowych. W takim przypadku:

- Iteruje przez wszystkie elementy o klasie `.js-matrix-item`, reprezentujące komórki macierzy.
- Dla każdego elementu pobiera jego wartość i sprawdza, czy zawiera kropkę (czyli czy jest liczbą zmiennoprzecinkową).
- Jeśli wartość zawiera kropkę, usuwa wartość elementu, ustawiając ją na pusty ciąg znaków.

- Jeśli wartości zmiennoprzecinkowe zostały usunięte, wywołuje funkcję `setReduceButton()`, aby zaktualizować stan przycisku „Reduce!” na interfejsie użytkownika.

4.2.1.3.12. Funkcja `checkMatrixItem()`

Funkcja `checkMatrixItem(item)` jest używana do sprawdzania i modyfikowania wartości pojedynczej komórki macierzy zgodnie z określonymi regułami i wybranym algorytmem. Działa zarówno dla algorytmów dopuszczających, jak i niedopuszczających wartości zmiennoprzecinkowych, i zawiera wiele warunków, aby zapewnić, że wartość jest zgodna z oczekiwaniami. Po zakończeniu sprawdzania, funkcja aktualizuje macierz i interfejs użytkownika, aby odzwierciedlić wszelkie zmiany.

Jako parametr wejściowy funkcja przyjmuje element `item` reprezentujący komórkę macierzy, którą należy sprawdzić, zaś funkcję przedstawiono na Listingu 4.25. Funkcja ta, podobnie do poprzedniej pobiera wartość `item` i przypisuje ją do zmiennej `value`, pobiera wartość wybranego algorytmu z elementu o klasie `.js-algorithm` i przypisuje ją do zmiennej `algorithm` oraz sprawdza, czy algorytm dopuszcza wartości zmiennoprzecinkowe (jeśli `algorithm` jest różne od '3') i przypisuje wynik do zmiennej `allowDec`. Następnie następuje walidacja wartości dla algorytmu, który dopuszcza zmienne przecinkowe, a więc w naszym przypadku Xu & Wei – jeśli `allowDec` jest prawdziwe, funkcja:

- Konwertuje wartość na zmienną zmiennoprzecinkową (`f`) i całkowitą (`i`).
- Oblicza odwrotność wartości odwrotne (`inv`).
- Usuwa wartość, jeśli odwrotność jest większa lub równa 9.
- Usuwa wartość, jeśli jest równa 0.
- Usuwa wartość, jeśli jej długość wynosi 2 i zawiera kropkę na drugiej pozycji.
- Ustawia wartość na wartość całkowitą, jeśli wartość zmiennoprzecinkowa i wartość całkowita są równe, ale wartość zawiera kropkę.
- Usuwa wartość, jeśli jej długość wynosi więcej niż 6 znaków.

Jeśli `allowDec` jest fałszywe i długość wartości wynosi więcej niż 3 znaki, usuwa wartość.

Następuje jeszcze dodatkowa walidacja i poprawa danych, gdyż funkcja:

- ustawia wartość na '1', jeśli wartość jest równa '1/1'.
- usuwa wartość, jeśli jej długość wynosi 2 i zawiera '/' na drugiej pozycji.
- usuwa wartość, jeśli zaczyna się od '.' lub '/'.
- usuwa wartość, jeśli '.' lub '/' znajduje się po pierwszym znaku.

Na samym końcu wywoływana jest funkcja `setMatrixValue(item)`, aby zaktualizować wartość macierzy, a także `setReduceButton()`, aby zaktualizować stan przycisku reduce na interfejsie użytkownika.

4.2.1.3.13. Funkcja `$(document).ready()`

Kod przedstawiony na Listingu 4.26. jest typowym przykładem inicjalizacji różnych zdarzeń i funkcji, które mają zostać wykonane po załadowaniu dokumentu [155]. Jest to często stosowany wzorzec w jQuery, który zapewnia, że cały kod JavaScript zostanie wykonany dopiero po pełnym załadowaniu strony.

Oto szczegółowy opis tego, co robi każda część funkcji:

- **Inicjalizacja macierzy i przycisku Reduce:**
 - Wywołuje `setMatrixSize()` do ustawienia początkowego rozmiaru macierzy.
 - Wywołuje `setReduceButton()` do ustawienia początkowego stanu przycisku „Reduce!”.
- **Zmiana rozmiaru macierzy:**
 - Nasłuchuje na zmianę wartości elementu o identyfikatorze `#matrixSize`.
 - Po zmianie wywołuje `setMatrixSize()` i `setReduceButton()`, a następnie wywołuje zdarzenie zmiany rozmiaru okna.
- **Zmiana wartości komórki macierzy:**
 - Nasłuchuje na zmianę wartości elementów o klasie `.js-matrix-item`.
 - Po zmianie wywołuje `checkMatrixItem(item)` dla zmienionego elementu.
- **Obsługa wprowadzania danych w komórce macierzy:**
 - Nasłuchuje na zdarzenie `keypress` dla elementów o klasie `.js-matrix-item`.
 - Sprawdza, czy wprowadzony znak jest dozwolony na podstawie wybranego algorytmu i obecnej wartości.
 - Jeśli znak nie jest dozwolony, zapobiega jego wprowadzeniu.
- **Resetowanie macierzy:**
 - Nasłuchuje na zdarzenie `click` dla elementu o klasie `.js-reset-button`.
 - Wywołuje `resetMatrix()` i czyści wyświetlaną zredukowaną macierz.
- **Czyszczenie wyświetlanej zredukowanej macierzy:**
 - Nasłuchuje na zmianę elementu o klasie `.js-clear-reduced-matrix`.

- Czyści wyświetlaną zredukowaną macierz.
- **Przełączanie listy:**
 - Nasłuchuje na zdarzenie click dla elementu o klasie `.js-toggle`.
 - Przełącza widoczność elementu o klasie `.js-list` za pomocą efektu ślizgania.
- **Sprawdzanie macierzy po zmianie algorytmu:**
 - Nasłuchuje na zmianę elementu o klasie `.js-algorithm`.
 - Wywołuje `checkMatrix()`.

Podsumowując, wyżej omówiony kod jest odpowiedzialny za inicjalizację różnych zdarzeń i funkcji związanych z manipulacją i interakcją z macierzą na stronie internetowej. Obejmuje to zmianę rozmiaru macierzy, wprowadzanie danych, generowanie losowej macierzy, resetowanie macierzy, wysyłanie macierzy, a także różne interakcje z interfejsem użytkownika, takie jak przełączanie listy i czyszczenie wyświetlanej zredukowanej macierzy. Każde z tych zdarzeń jest starannie obsługiwane, aby zapewnić spójność i poprawność danych oraz interakcji użytkownika.

4.2.2. Warstwa back-end

Back-end aplikacji REDUCE zbudowany został przy użyciu frameworka Flask. Wykorzystuje różne biblioteki, takie jak `numpy`, `scipy` i `sympy`, a przede wszystkim także bibliotekę `REDUCE.py` (opisaną w Rozdziale 3.) które są powszechnie stosowane w obliczeniach naukowych i inżynierskich. Import bibliotek oraz dwie podstawowe funkcje związane ze środowiskiem Flask znajdują się na Listing 4.27.

W kwestii podstawowych funkcji przedstawionych na wspomnianym listingu można przedstawić ich działanie następująco:

- **main()** - Ta funkcja odpowiada za renderowanie strony głównej aplikacji. Jest mapowana na główny adres URL ("/") i wykorzystuje plik szablonu HTML (opisany w części front-end) do wyświetlenia odpowiedniej strony użytkownikowi.
- **if name == "main": app.run** - Ta linijka kodu uruchamia aplikację, jeśli plik jest wykonywany jako główny skrypt. Jest to typowy sposób uruchamiania aplikacji Flask w środowisku deweloperskim.

Pozostałe dwie funkcje zostaną opisane w kolejnych podsekcjach.

4.2.2.1. Funkcja `computeCI_CR()`

Funkcja jest dostępna pod adresem `/computeCI_CR` i obsługuje żądania typu POST (ukazana na Listing 4.28). Służy do obliczania wartości współczynników dla macierzy,

jeszcze przed kliknięciem przycisku Reduce! w warstwie front-end. Funkcja odbiera dwa parametry z żądania POST:

- **_matrixSize:** Rozmiar macierzy, który jest przekształcany na wartość całkowitą.
- **_reduceThreshold:** Parametr, który jest odczytywany, ale nie jest używany w przedstawionym kodzie.

Na podstawie wartości `_matrixSize_val`, inicjalizowana jest macierz `mx` o odpowiednich wymiarach, wypełniona wartościami 1. Wartości są odczytywane z żądania POST i przypisywane do odpowiednich komórek macierzy `mx` i konwertowane na typ `float` za pomocą dodatkowej funkcji `convertValueToFloat`. Dalej, wykonywane są różne obliczenia na macierzy, takie jak obliczanie wartości własnych, wskaźników spójności, błędów względnych itp. gdzie wykorzystywane są różne funkcje biblioteki REDUCE.

Funkcja zwraca słownik `send_dict`, który zawiera wyniki obliczeń, a więc obliczone wartości CI (Consistency Index), CR (Consistency Ratio), EV (Eigenvalue), KI (Koczkodaj Index), GC (Geometric Consistency Index), GW (Golden Wang Index), PLI (Pelaez Lamata Index), TGCI (Tirads Geometric Consistency Index), RE (Relative Error), HC (Harmonic Consistency Index).

4.2.2.2. Funkcja `reduceButton()`

Funkcja `reduceButton` (Listing 4.29) jest funkcją back-endu, która obsługuje żądanie POST na ścieżce `/reduceButton`. Obsługuje ona naciśnięcie przycisku „Reduce!”, gdy stanie się on aktywny. Poniżej znajduje się szczegółowy opis działania tej funkcji:

- Funkcja pobiera dane z żądania POST, takie jak rozmiar macierzy (`_matrixSize`), próg redukcji (`_reduceThreshold`) i wybrany algorytm (`_selectedAlgorithm`).
- Rozmiar macierzy jest konwertowany na wartość całkowitą (`_matrixSize_val`).
- Na podstawie wartości `_matrixSize_val`, inicjalizowana jest macierz `mx` o odpowiednim rozmiarze, wypełniona wartościami 1.
- Funkcja wypełnia macierz `mx` wartościami z żądania POST, korzystając z funkcji `convertValueToFloat`. Wypełnianie odbywa się w sposób warunkowy, w zależności od rozmiaru macierzy.
- Na podstawie wartości `_reduceThreshold`, ustalany jest próg redukcji (`reduceThreshold_val`).

- W zależności od wybranego algorytmu (`_selectedAlgorithm`), wywoływana jest odpowiednia funkcja (`xu_wei()` lub `szybowski()` z pakietu `Reduce`), która przetwarza macierz `m \times n` z określonym progiem redukcji i rozmiarem macierzy.
- Wykonywane są dodatkowe obliczenia, takie jak:
 - **EV**: Wektor priorytetów.
 - **GC**: Spójność geometryczna.
 - **KI**: Indeks Koczkodaja.
 - **GW**: Golden Wang.
 - **PLI**: Pelaez Lamata.
 - **TGCI**: Spójność geometryczna Tirads.
 - **RE**: Błąd względny.
 - **HC**: Spójność harmoniczna.
- Wyniki są zapisywane w słowniku `dict_1`, który jest następnie aktualizowany słownikiem `dict_2`, zawierającym dodatkowe wyniki obliczeń.
- Funkcja zwraca słownik `dict_1`, który zawiera wyniki obliczeń.

W przedstawionym kodzie funkcji (również tej z poprzedniego podrozdziału) nie ma bezpośredniej obsługi błędów na poziomie serwera, ponieważ dane wejściowe są zabezpieczone na froncie aplikacji. Oznacza to, że wszelkie walidacje i ograniczenia dotyczące danych wejściowych są już przeprowadzane na poziomie interfejsu użytkownika, zanim zostaną przesłane do serwera.

Dzięki temu ryzyko otrzymania nieprawidłowych lub nieoczekiwanych danych na serwerze jest znacznie zminimalizowane. Jest to powszechna praktyka w projektowaniu aplikacji webowych, która pozwala na bardziej efektywną i responsywną interakcję z użytkownikiem, jednocześnie redukując obciążenie serwera.

4.2.3. Rozwiązania serwerowe i administracyjne

W rozwoju nowoczesnej aplikacji, kluczowym elementem jest wybór odpowiednich rozwiązań serwerowych, które zapewnią nie tylko wydajność i skalowalność, ale także bezpieczeństwo i niezawodność systemu. W kontekście opisywanej aplikacji, zastosowano zestaw narzędzi i technologii, które wspólnie tworzą spójną i zintegrowaną architekturę serwerową. Wykorzystanie Gunicorn jako serwera aplikacji, współpracującego z NGINX jako serwerem proxy, zapewnia optymalizację zarządzania zasobami i równowagę obciążenia. Integracja z Certbotem umożliwia zarządzanie certyfikatami SSL, co jest

kluczowe dla bezpieczeństwa komunikacji. Dodatkowo, zastosowanie firewalle ufw oraz systemów wykrywania włamań takich jak Snort i rozwiązań Cisco, tworzy wielowarstwową ochronę przed potencjalnymi zagrożeniami. Wszystkie te elementy razem tworzą spójny ekosystem, który jest dostosowany do specyficznych wymagań i celów aplikacji, a także odzwierciedla najlepsze praktyki w dziedzinie zarządzania serwerami i bezpieczeństwa sieciowego. W dalszej części rozdziału przyjrzymy się szczegółowo, jak te rozwiązania zostały zintegrowane i skonfigurowane, aby sprostać wyjątkowym wyzwaniom związanym z funkcjonowaniem opisywanej aplikacji.

4.2.3.1 Konfiguracja Gunicorn oraz NGINX

Gunicorn działa jako serwer aplikacji WSGI, który obsługuje żądania do aplikacji "REDUCE". NGINX działa jako odwrotny serwer proxy, który przekierowuje ruch do Gunicorna i obsługuje statyczne pliki oraz SSL. Ta kombinacja zapewnia wydajne i elastyczne rozwiązanie do obsługi ruchu sieciowego, zabezpieczając jednocześnie połączenia i umożliwiając łatwe skalowanie aplikacji. Plik konfiguracyjny dla usługi Gunicorn znajduje się na Listing 4.30. a tłumacząc poszczególne jego elementy należy zwrócić uwagę na:

- **Bind:** Adres IP i port, na którym Gunicorn będzie nasłuchiwał. W tym przypadku jest to lokalny adres IP, a port to 8000.
- **Workers:** Liczba workerów (procesów) obsługujących żądania. Jest to zależne od liczby dostępnych rdzeni CPU, co pozwala na efektywne wykorzystanie zasobów.
- **Timeout:** Maksymalny czas oczekiwania na odpowiedź od workera. Jeśli worker nie odpowie w tym czasie, zostanie zrestartowany.
- **Worker Class:** Typ workera używany przez Gunicorn. W tym przypadku używany jest typ synchroniczny.
- **LogLevel:** Poziom logowania, który określa, jakie informacje są rejestrowane.
- **Errorlog i Accesslog:** Ścieżki do plików z logami błędów i dostępu, które pomagają w monitorowaniu i diagnozowaniu problemów.

Uruchomienie Gunicorna z tym (Listing 4.31) plikiem konfiguracyjnym można zrealizować za pomocą polecenia: `gunicorn -c gunicorn_config.py reduce_app:app`. Następnym krokiem jest konfiguracja NGINX, która znajduje się na Listing 4.31. Ta konfiguracja zapewnia, że ruch do domeny `reduce.prz.edu.pl` jest obsługiwany przez NGINX i przekierowywany do Gunicorna, z zachowaniem bezpiecznych połączeń HTTPS i optymalnym serwowaniem statycznych plików.

Konfiguracja serwera NGINX dla aplikacji REDUCE składa się z dwóch głównych bloków. Pierwszy blok nasłuchuje na porcie 80 i przekierowuje cały ruch na HTTPS dla domeny reduce.prz.edu.pl, zapewniając bezpieczne połączenie. Drugi blok nasłuchuje na porcie 443 (HTTPS) i zawiera ścieżki do certyfikatów SSL wygenerowanych przez Certbot dla tej domeny. W ramach tej konfiguracji są również zdefiniowane bloki lokalizacji, które obsługują statyczne pliki i przekierowują ruch do Gunicorna. Dodatkowo, pliki logów są nazwane zgodnie z domeną, co ułatwia zarządzanie i diagnozowanie problemów, tworząc spójny i bezpieczny system zarządzania ruchem dla aplikacji.

Po utworzeniu pliku konfiguracyjnego NGINX, należy go aktywować i zrestartować serwer co jest widoczne na Listingu 4.32.

4.2.3.2 Konfiguracja narzędzia Certbot (Let's Encrypt)

Generowanie certyfikatów SSL za pomocą narzędzia Certbot dla aplikacji REDUCE jest niezbędnym elementem w zapewnieniu bezpiecznych połączeń HTTPS. Proces ten rozpoczyna się od instalacji Certbota na serwerze, co można zrealizować na systemie Ubuntu za pomocą poleceń przedstawionych na Listingu 4.33. Następnie, certyfikat dla domeny reduce.prz.edu.pl generowany jest za pomocą polecenia z Listingu 4.34.

Opcja `--nginx` informuje Certbota o użyciu serwera NGINX, a opcja `-d` określa domenę. Ponieważ certyfikaty Let's Encrypt mają okres ważności 90 dni, konieczne jest skonfigurowanie automatycznego odnowienia certyfikatów. Można to zrobić, dodając do crontaba [156,157] polecenie widoczne na Listingu 4.35.

4.2.3.3 Konfiguracja firewalli i zabezpieczeń

Obrona aplikacji REDUCE jest złożonym procesem, który wykorzystuje różnorodne narzędzia i technologie w celu zapewnienia pełnego zabezpieczenia. Kluczowymi elementami tej obrony są: Uncomplicated Firewall (ufw), system wykrywania i zapobiegania włamaniom Snort oraz sprzętowy firewall Cisco ASA. Każde z tych narzędzi odgrywa unikalną rolę w ochronie aplikacji przed różnymi rodzajami zagrożeń i ataków.

Uncomplicated Firewall (ufw) to narzędzie do zarządzania iptables dostępne na systemach Linux. Jest to intuicyjny interfejs do zarządzania zasadami firewalla, który może być wykorzystany do zabezpieczenia aplikacji REDUCE. Na Listingu 4.36 przedstawiono konfigurację ufw, który jest integralną częścią tej wielowarstwowej strategii obrony.

Najpierw aktywowany jest firewall za pomocą pierwszego polecenia. Następnie otwierane są porty 80 (HTTP) i 443 (HTTPS) do obsługi ruchu sieciowego, a także port 22 dla połączeń SSH, co umożliwi zdalne zarządzanie serwerem. W celu zwiększenia

bezpieczeństwa, wszystkie inne porty są blokowane, a status i aktywne reguły można sprawdzić za pomocą polecenia ostatniego. Ta konfiguracja zapewnia niezbędne zabezpieczenia dla aplikacji REDUCE, umożliwiając ruch na wymaganych portach, jednocześnie blokując nieautoryzowany dostęp.

Snort to narzędzie do wykrywania i zapobiegania włamaniom, które dostosowano do ochrony aplikacji REDUCE i jest kluczowym elementem jej zabezpieczenia. Proces rozpoczyna się od instalacji Snorta na systemie Ubuntu za pomocą poleceń widocznych na Listingu 4.37.

Następnie konfiguracja interfejsu sieciowego jest wymagana, aby Snort mógł nasłuchiwać na odpowiednim interfejsie. Edycja pliku `/etc/snort/snort.conf` i ustawienie `ipvar HOME_NET` wraz z adresem serwera pozwala na taką konfigurację. Kolejnym krokiem są reguły wykrywania będące „sercem” Snorta, definiując, jakie rodzaje ruchu są uważane za podejrzane. Dla aplikacji REDUCE, specjalne reguły zostały dodane do pliku reguł w `/etc/snort/rules`, odzwierciedlając jej unikalne wymagania bezpieczeństwa i są widoczne na Listingu 4.38.

Wybór tych konkretnych reguł dla aplikacji jest uzasadniony różnymi aspektami bezpieczeństwa, które są kluczowe dla ochrony i prawidłowego działania aplikacji:

- **Wykrywanie próby SQL Injection:** Choć aplikacja REDUCE nie posiada bazy danych, tym niemniej używa tablic NumPy oraz pandas które także są podatne na tego typu ataki [158-160]. Reguła ta zapobiega próbom nieautoryzowanego dostępu do tych danych poprzez wstrzyknięcie złośliwego kodu SQLi wykrywa próby wstrzyknięcia kodu SQL w żądaniach HTTP.
- **Wykrywanie próby ataku XSS:** Ataki XSS mogą pozwolić atakującemu na wykonywanie skryptów w kontekście przeglądarki ofiary [161-165]. Ta reguła chroni użytkowników przed takimi atakami, które mogą prowadzić do kradzieży danych. Wykrywa próby ataku typu cross-site scripting poprzez umieszczanie tagów `<script>` w żądaniach.
- **Wykrywanie próby przeciążenia serwera:** Ataki DDoS mogą spowodować przeciążenie serwera i uniemożliwić prawidłowe działanie aplikacji [166-170]. Ta reguła pomaga w wykrywaniu i zapobieganiu takim atakom, zapewniając ciągłość działania usługi. Wykrywa potencjalne próby ataku DDoS poprzez analizę liczby żądań od pojedynczego adresu IP w określonym czasie.

Uruchomienie Snorta odbywa się za pomocą polecenia widocznego na Listingu 4.39. Po jego uruchomieniu rozpoczyna monitorować ruch sieciowy, logując wszelkie podejrzane działania zgodnie z zdefiniowanymi regułami. Te logi mogą być następnie analizowane, aby

zrozumieć i zareagować na potencjalne zagrożenia, zapewniając kompleksową ochronę dla aplikacji REDUCE.

W ramach kompleksowego podejścia do zabezpieczeń aplikacji REDUCE, konfiguracja firewalla Cisco ASA 5505 stanowi uzupełniającą warstwę ochrony, która działa w tandemie z innymi mechanizmami obronnymi, opisanymi do tej pory. Podczas gdy ufw zapewnia prostą kontrolę dostępu i filtrowanie na poziomie hosta, a Snort działa jako system wykrywania i zapobiegania intruzjom, konfiguracja Cisco ASA 5505 dodaje dodatkową warstwę zabezpieczeń na poziomie brzegu sieci. Ta zintegrowana strategia zapewnia wielowarstwową ochronę, która może skutecznie zminimalizować ryzyko naruszenia i zapewnić ciągłą dostępność i integralność aplikacji REDUCE.

Pierwszym krokiem w konfiguracji urządzenia było określenie nazw i adresów IP dla interfejsów sieciowych widoczne na Listingu 4.43. Kolejnym punktem była konfiguracja reguł NAT i ACL, aby umożliwić ruch na portach 80 (HTTP), 443 (HTTPS), i 22 (SSH) – Listing 4.40. Skorzystano także z rozwiązań unikalnych dla firewalla Cisco, a więc inspekcji ruchu ICMP, przedstawionej na Listingu 4.41. Poleceniem write memory zapisano konfigurację, której status przedstawia wyciąg z terminala na Listingu 4.42.

Ta konfiguracja zapewnia podstawowe zabezpieczenia i dostęp do aplikacji REDUCE, z możliwością dalszego dostosowania zgodnie z indywidualnymi wymaganiami bezpieczeństwa i działania.

4.2.3.4 Konfiguracja dostępu administracyjnego

Konfiguracja dostępu administracyjnego do narzędzia aplikacji REDUCE za pomocą sieci VPN Politechniki Rzeszowskiej, opartej na technologii Forti i protokołu SSH, została zabezpieczona zestawem kluczy RSA, zapewniających dostęp do terminala systemu Ubuntu Server. Klucze RSA zostały wygenerowane i zainstalowane, z kluczem publicznym umieszczonym na serwerze Ubuntu, a kluczem prywatnym zabezpieczonym i używanym do uwierzytelniania.

Dostęp do serwera został ograniczony do połączeń pochodzących z sieci VPN Politechniki Rzeszowskiej, z dodatkową konfiguracją SSH na serwerze, aby umożliwić uwierzytelnianie za pomocą kluczy RSA i zezwolić na połączenia tylko z zaufanych adresów IP. Regularny monitoring i audyt logów SSH zostały również wdrożone, aby zapewnić wykrywanie i reagowanie na wszelkie podejrzane działania.

Ta staranna konfiguracja zapewnia silne uwierzytelnianie i ogranicza dostęp do serwera tylko do autoryzowanych użytkowników, co jest kluczowe dla ochrony wrażliwych danych i funkcji zarządzania aplikacją REDUCE.

4.3. Testy zgodności, bezpieczeństwa oraz wydajnościowe aplikacji

Testowanie aplikacji REDUCE stanowi kluczowy element procesu zapewnienia jej jakości, wydajności i bezpieczeństwa. Wybór różnych rodzajów testów odzwierciedla złożoność i różnorodność funkcji aplikacji oraz potencjalne zagrożenia, które mogą wystąpić w różnych scenariuszach użytkowania. Oto uzasadnienie wyboru poszczególnych rodzajów testów:

- **Testy na różnych urządzeniach i platformach:** Aplikacja REDUCE, aby być skuteczna i dostępna dla szerokiego grona użytkowników, musi działać sprawnie na różnych urządzeniach, systemach operacyjnych i przeglądarkach. Dzięki temu można zidentyfikować i rozwiązać problemy z kompatybilnością, które mogą wpływać na doświadczenia użytkowników [171-176]. W kontekście aplikacji REDUCE, gdzie różnorodność użytkowników i ich preferencji technologicznych może być znacząca, testy te są niezbędne do zapewnienia spójnego i pozytywnego doświadczenia dla wszystkich użytkowników, niezależnie od używanego sprzętu czy oprogramowania.
- **Testy bezpieczeństwa i firewalla:** Bezpieczeństwo jest ważnym aspektem aplikacji REDUCE, zwłaszcza że przetwarza ona dane pochodzące od użytkowników. Testy bezpieczeństwa i firewalla mają na celu ocenę, jak dobrze zabezpieczony jest system przed potencjalnymi atakami i próbami naruszenia [177-182].
- **Testy SSL:** Komunikacja między klientem a serwerem w aplikacji REDUCE musi być zabezpieczona, aby zapewnić prywatność i integralność danych. Testy SSL oceniają konfigurację i wydajność certyfikatów SSL, zapewniając, że połączenia są odpowiednio zabezpieczone i zgodne ze standardami branżowymi [183-185].
- **Testy wydajności ładowania:** Wydajność ładowania strony jest kluczowym elementem doświadczenia użytkownika w aplikacji REDUCE. Długie czasy ładowania mogą prowadzić do frustracji użytkowników i ostatecznie do opuszczenia strony. Testy wydajności ładowania (w przypadku tej aplikacji z wykorzystaniem narzędzia GTMetrix [311]) pozwalają na analizę czasu ładowania strony, optymalizację zasobów i identyfikację potencjalnych wąskich gardeł w systemie [186-192].

4.3.1. Testy zgodności na różnych urządzeniach i platformach

W dzisiejszym złożonym środowisku cyfrowym, gdzie aplikacje muszą działać na różnych platformach, testowanie kompatybilności staje się kluczowym elementem procesu

rozwoju. W tym kontekście, narzędzie BitBar [193] zostało wybrane jako przedmiot analizy ze względu na dwie istotne cechy, które miały wpływ na proces testowania.

- **Skalowalność testowania:** BitBar oferuje możliwość równoczesnego uruchamiania testów na różnych przeglądarkach i urządzeniach, zwiększyło pokrycie testów i skróciło czas ich wykonania.
- **Dostęp do rzeczywistych środowisk:** Narzędzie to umożliwia testowanie na prawdziwych urządzeniach i przeglądarkach, co może zwiększyć wiarygodność wyników testów, eliminując jednocześnie konieczność konfiguracji i utrzymania własnych środowisk testowych. Przeglądarki dostępne w ramach narzędzia odpowiadają recenzowanym badaniom najpopularniejszych wśród użytkowników przeglądarek internetowych w ostatnich latach [194, 195].

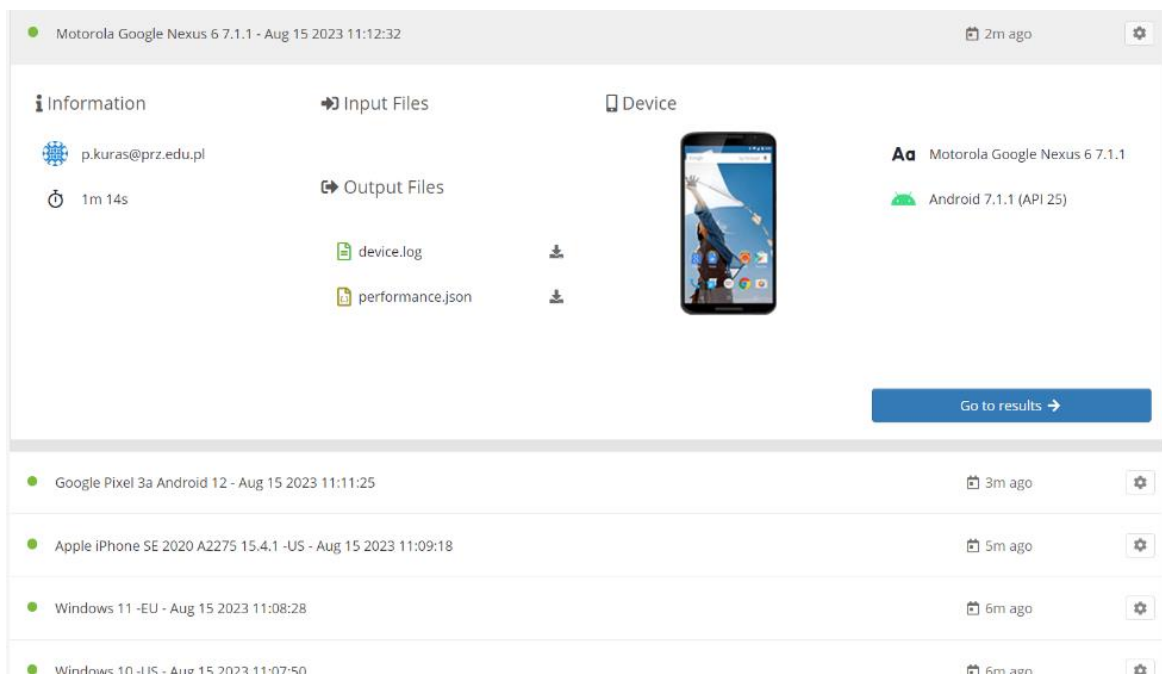
W ramach procesu testowania, przeprowadzono sesje na różnych urządzeniach i systemach operacyjnych. Testy obejmowały przeglądarki wbudowane w urządzenia Motorola Google Nexus 6 z systemem Android 7.1.1, Google Pixel 3a z Androidem 12, Apple iPhone SE 2020 z wersją iOS 15.4.1, Apple iPhone 7 z iOS 15.4.1, a także komputery (maszyny wirtualne) z systemami Windows 10 i 11 oraz macOS 12. Sesje testowe odbyły się w różnych lokalizacjach, w tym w Stanach Zjednoczonych i Europie, zaś przeglądarki jakie brano pod uwagę to:

- Windows 10:
 - Chrome 115, 114, 113, 112, 111, 110
 - Firefox 115, 114, 113, 112, 111, 110
 - Microsoft Edge 115, 114, 113, 112, 111, 110
 - Internet Explorer 11
- Windows 11:
 - Chrome 115, 114, 113, 112, 111, 110
 - Firefox 115, 114, 113, 112, 111, 110
 - Microsoft Edge 115, 114, 113, 112, 111, 110
- macOS 12:
 - Chrome 115, 114, 113, 112, 111, 110
 - Firefox 115, 114, 113, 112, 111, 110
 - Safari 15

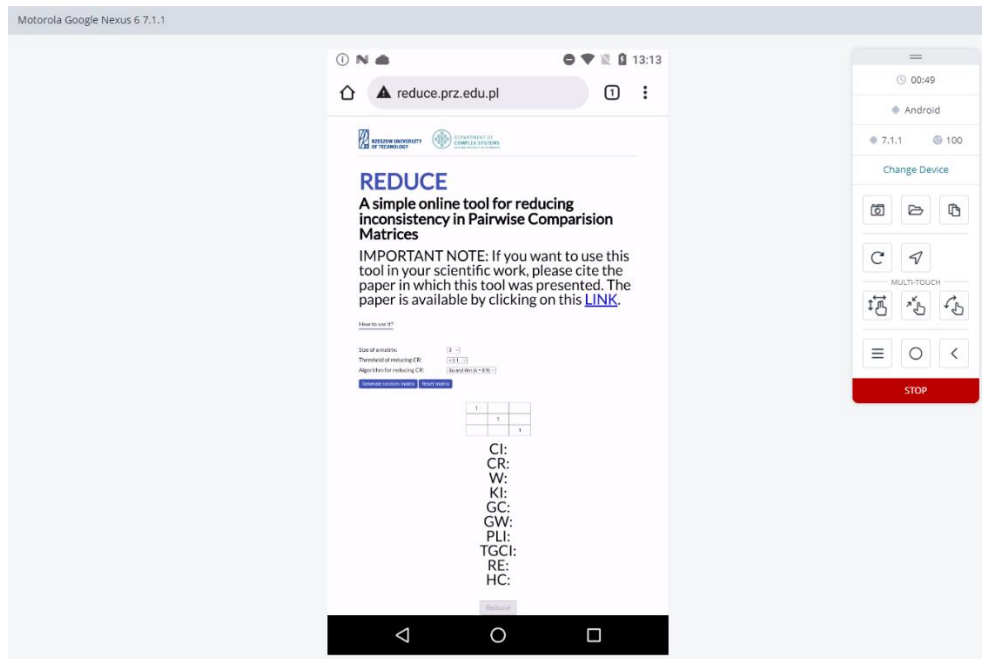
Proces testowania przeglądarek za pomocą BitBar Browser Live Testing rozpoczyna się od konfiguracji bezpiecznego tunelu dla lokalnych zasobów. Następnie uruchamiany jest test, przy wyborze systemu operacyjnego, wersji przeglądarki i rozdzielczości, a także

opcjonalnym wprowadzeniu bezpośredniego URL (Rys. 4.11). Po rozpoczęciu testu, możliwa jest interakcja z testowaną stroną, rejestrowanie działań, robienie zrzutów ekranu i przesyłanie plików, a test może być zakończony, zatrzymany lub wstrzymany w dowolnym momencie (widoczne na Rys. 4.12), a po zakończeniu BitBar dostarcza wyniki, w tym szczegóły sesji, artefakty i analizę wydajności (Rys. 4.13, 4.14). Możliwe jest również przeglądanie poprzednich sesji testowych.

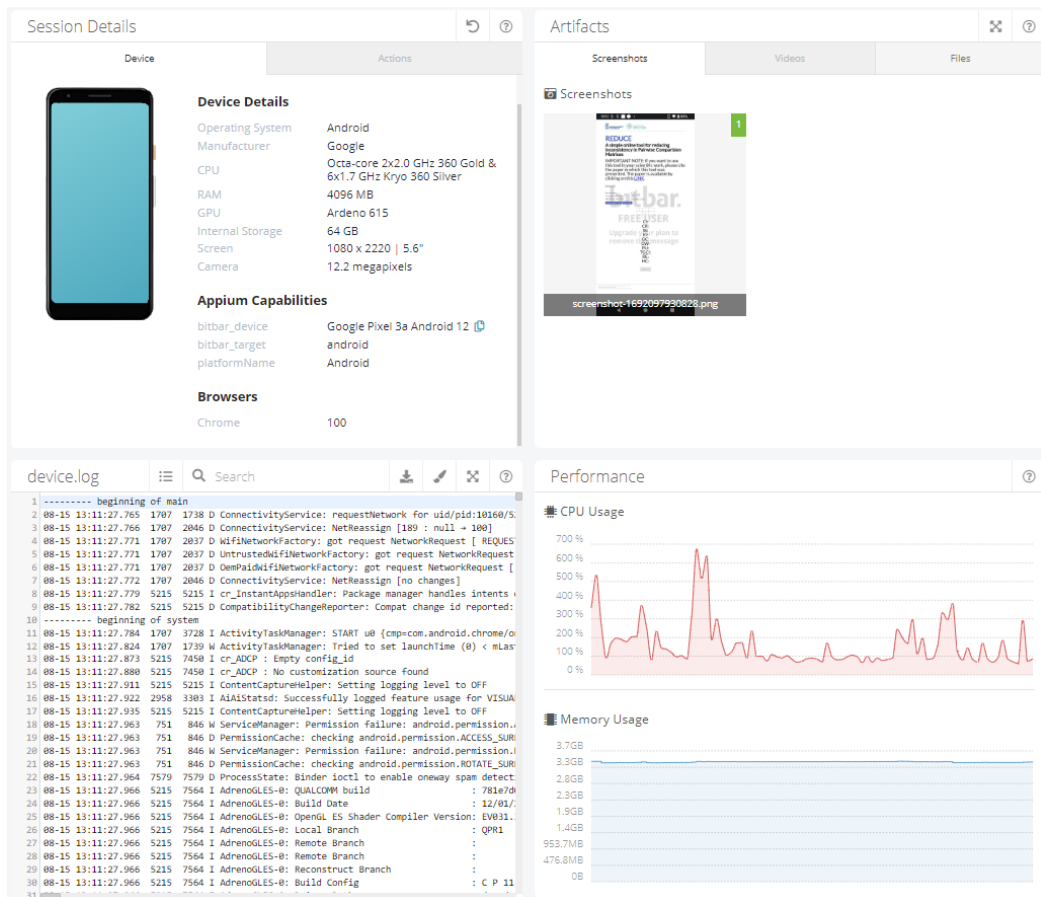
Cały proces oferuje kompleksowe narzędzie do interaktywnego testowania stron internetowych w różnych środowiskach przeglądania. W procesie testowania strony internetowej na różnych przeglądarkach i rozdzielczościach ekranu, dokonano szczegółowej analizy jakości wyświetlania i funkcjonalności. Dla wszystkich rozdzielczości powyżej 1152x900, w tym 2560x1920, 2560x1600, 1920x1200, 1920x1080, 1440x1050, 1400x900, 1366x768, 1280x1024, 1280x800, strona wyświetla się w 100% poprawnie i w całości na ekranie. Nie tylko estetyka i układ strony są zachowane, ale także wszystkie elementy funkcjonalne działają zgodnie z oczekiwaniami, zapewniając użytkownikom płynne i spójne doświadczenie, przykładowe wyświetlenie w przeglądarce Chrome 115 (1920x1080) na urządzeniu z Windows 10 znajduje się na Rys. 4.15, przeglądarki Safari 15 w rozdzielczości 1366x768 na urządzeniu macOS 12 na Rys. 4.17, zaś zaburzenia w wyświetlaniu przy rozdzielczości 1152x900 dla przeglądarki Firefox w systemie Windows 10 widoczne są na Rys. 4.16.



Rys. 4.11. Podgląd widoku testów w narzędziu BitBar Browser Live Testing




Rys. 4.12. Widok strony mobilnej REDUCE na prawdziwym urządzeniu z systemem Android korzystając z narzędzie BitBar Browser Live Testing



Rys. 4.13 Panel ze statystykami z odbytego testu na urządzeniu mobilnym Android

Session Details
↶ ?

Device
Actions

 Device

Device Details

Operating System	iOS
Manufacturer	Apple
CPU	Apple A13 Bionic
Instruction set	ARMv8.4-A
RAM	3072 MB
GPU	Apple GPU
Internal Storage	64 GB
Screen	4.7"
Camera	12 megapixels

Appium Capabilities

bitbar_device	Apple iPhone SE 2020 A2275 15.4.1 -US
bitbar_target	ios
platformName	iOS

Browsers

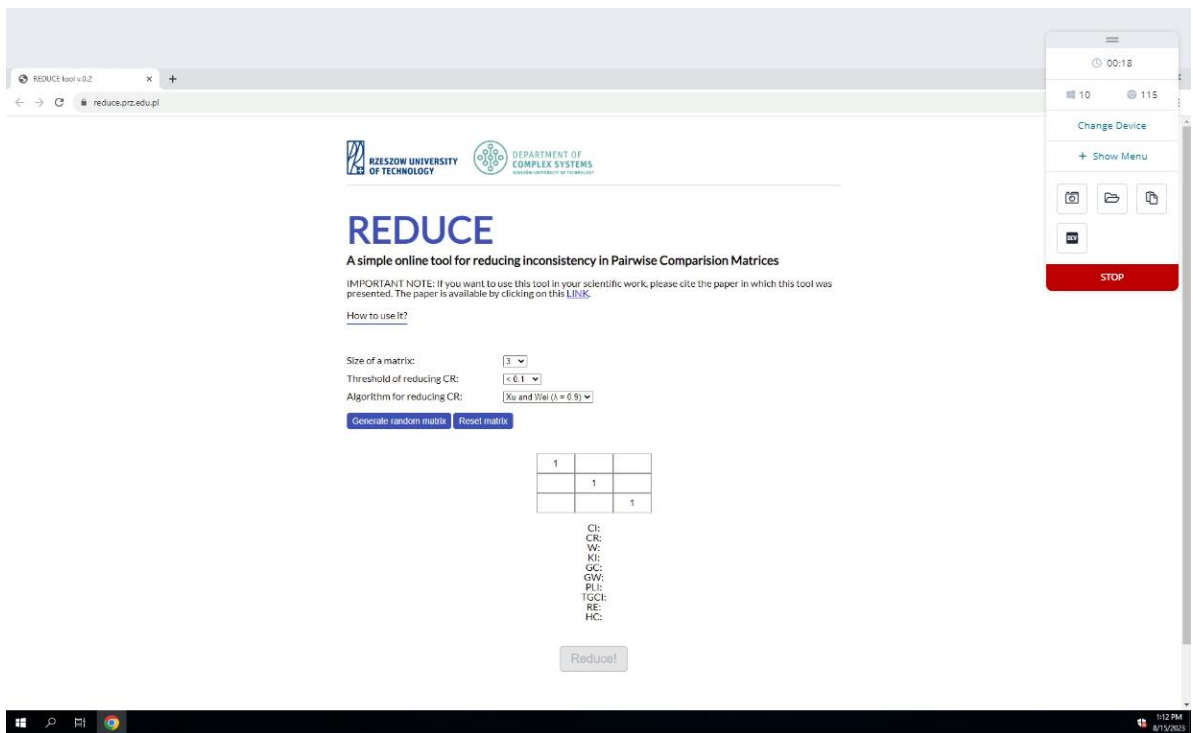
console.log
☰ 🔍 Search
⬇️ ✎ ✕ ?

```

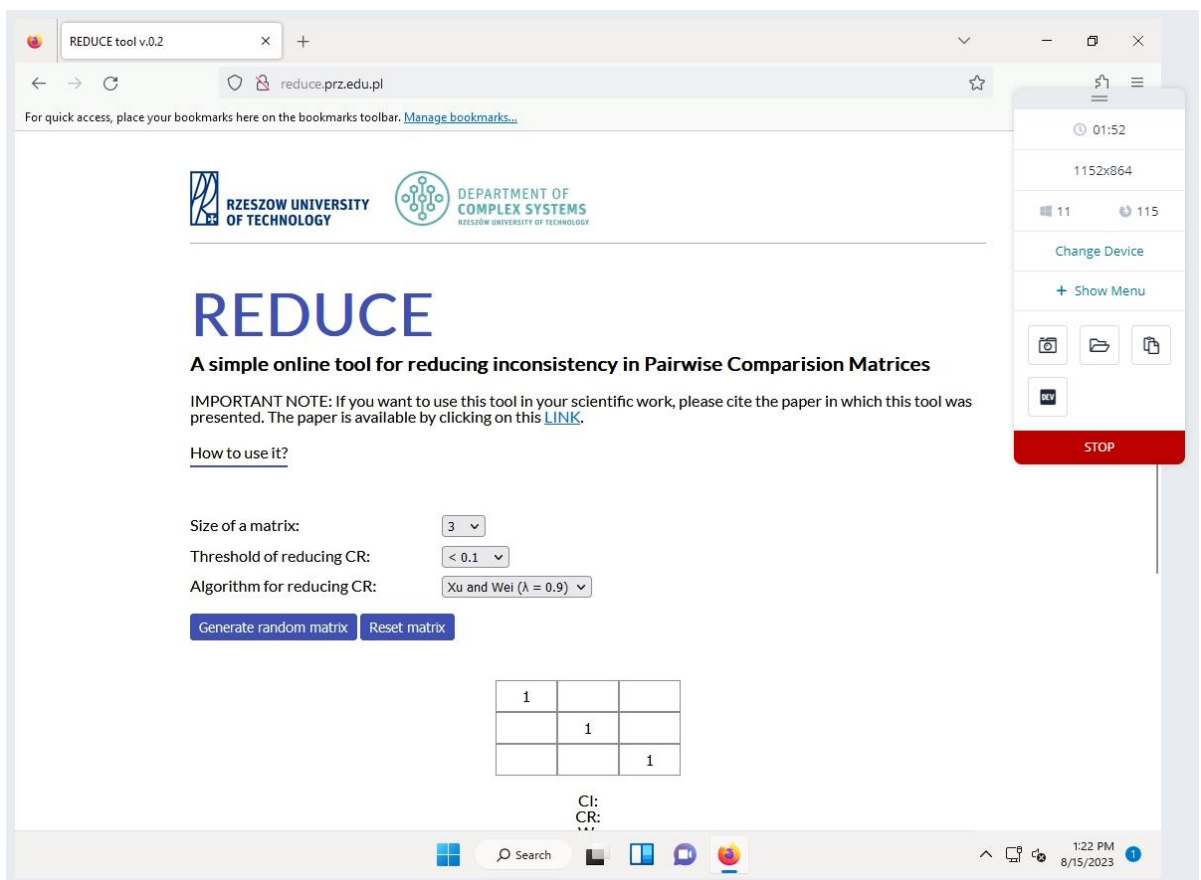
1 iostestios03.beta.wro.integration.hc.bitbar
2 uptime: /dev/ttys000: No such file or directory
3 4:09 up 45 mins, 1 user, load averages: 1.88 1.89 2.71
4 VM Version : 11.6-2 build : 2022-01-04 09:06:07 PST
5 Unlocking keychain...
6 en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
7   options=40b<RXCSUM, TXCSUM, VLAN_HWTAGGING, CHANNEL_IO>
8   ether 00:50:56:01:01:03
9   inet 192.168.195.130 netmask 0xffffff00 broadcast 192.168.195.255
10  media: autoselect (1000baseT <full-duplex>)
11  status: active
12 Setting authentication token...
13 Tue Aug 15 04:09:39 PDT 2023
14 Tue Aug 15 04:09:44 PDT 2023
15 Tue Aug 15 04:09:49 PDT 2023
16 Tue Aug 15 04:09:54 PDT 2023
17 Tue Aug 15 04:09:59 PDT 2023
18 Tue Aug 15 04:10:04 PDT 2023
19 Tue Aug 15 04:10:09 PDT 2023
20 Tue Aug 15 04:10:14 PDT 2023
21 Tue Aug 15 04:10:19 PDT 2023
22 Tue Aug 15 04:10:24 PDT 2023
23 Tue Aug 15 04:10:29 PDT 2023
24 Tue Aug 15 04:10:34 PDT 2023
25 Tue Aug 15 04:10:39 PDT 2023
26 Tue Aug 15 04:10:44 PDT 2023
27

```

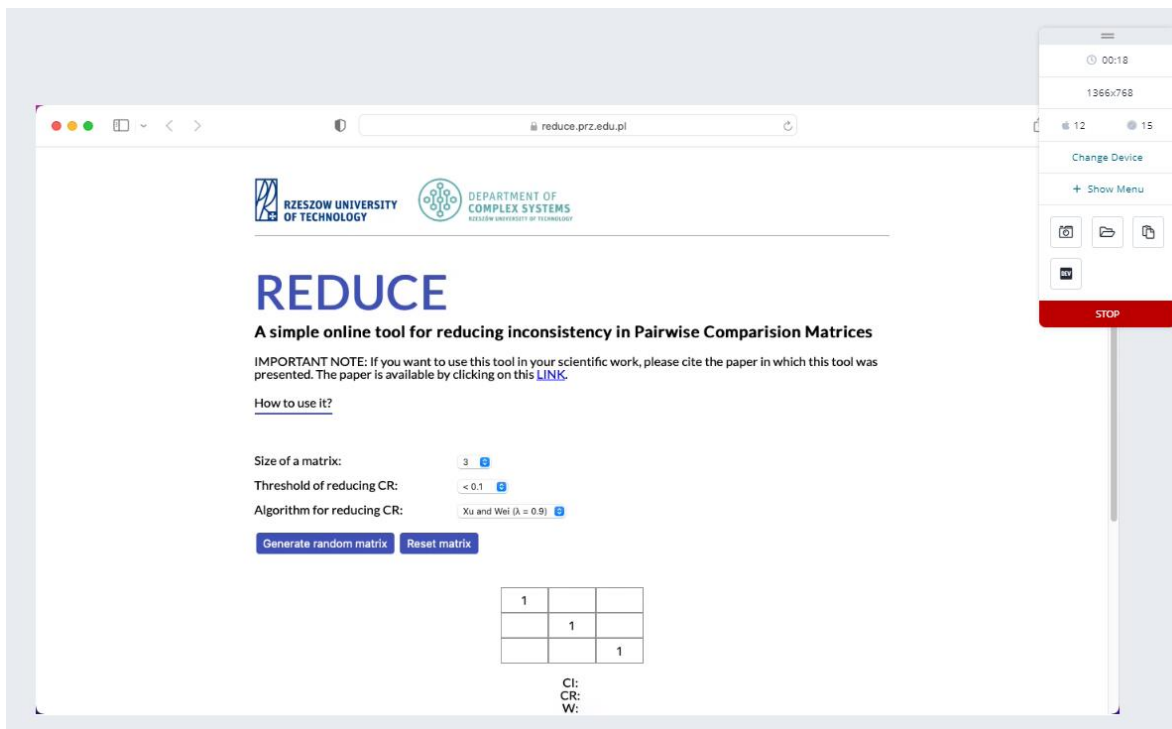
Rys. 4.14 Zawartość pliku console.log na urządzeniu iPhone



Rys. 4.15 Test aplikacji w przeglądarce Chrome z systemem Windows 10



Rys. 4.16 Test aplikacji w przeglądarce Firefox z systemem Windows 11



Rys. 4.17 Test aplikacji w przeglądarce Safari z systemem macOS 12

Jednakże, podczas testowania na urządzeniach mobilnych, zauważono pewne nieprawidłowości w wyglądzie strony (patrz na przykład Rys. 4.12, podobny schemat powtarzał się na innych urządzeniach branych pod uwagę). Choć strona działała poprawnie pod względem funkcjonalnym, wykryto drobne niezgodności wizualne, które mogą wpływać na estetykę i ogólne wrażenie użytkownika. Te nieprawidłowości zostały zidentyfikowane jako obszary do poprawy w kolejnych wersjach aplikacji. Ważne jest jednak podkreślenie, że te drobne kwestie wizualne nie wpływają na poprawną pracę z aplikacją i nie zakłócają kluczowych funkcji, co pozwala na dalsze korzystanie z aplikacji bez znaczących przeszkód.

Ważnym aspektem jest też analiza plików logów z różnych urządzeń, która dostarcza kluczowych informacji na temat działania aplikacji webowej oraz jej interakcji z systemem operacyjnym i sprzętem. W przypadku plików `console.log` i `device.log` pochodzących z urządzeń iPhone, te zapisy zawierają szczegółowe informacje na temat działania aplikacji w środowisku iOS.

- Plik **console.log** zawiera informacje na temat bieżących operacji wykonywanych przez aplikację, w tym komunikaty o błędach, ostrzeżeniach i innych zdarzeniach. Analiza tego pliku nie wykazała żadnych błędów w działaniu usługi, co świadczy o stabilności i niezawodności aplikacji na platformie Apple.
- Plik **device.log** dostarcza informacji na temat interakcji aplikacji z samym urządzeniem, takich jak zużycie baterii, połączenia sieciowe i inne zdarzenia na poziomie systemu.

Ponownie, analiza tego pliku nie wykazała żadnych nieprawidłowości ani ponadnormatywnego zużycia zasobów, co potwierdza optymalizację aplikacji pod kątem wydajności.

Co do urządzeń z systemem Android, analizowane pliki `performance.json` i `device.log` dostarczają podobnych, ale unikalnych dla platformy Android, informacji.

- Plik **`performance.json`** zawiera dane na temat wydajności aplikacji, takie jak czas odpowiedzi, zużycie pamięci i procesora. Analiza tego pliku nie wykazała żadnych problemów z wydajnością ani nadmiernego zużycia zasobów, co świadczy o efektywnym zarządzaniu zasobami przez aplikację.
- Plik **`device.log`** z Androida, podobnie jak jego odpowiednik na iPhone, zawiera informacje na temat interakcji aplikacji z urządzeniem. Analiza tego pliku również nie wykazała żadnych nieprawidłowości.

Podsumowując (pliki widoczne także na Rys 4.13 i 4.14), szczegółowa analiza tych plików logów z różnych platform i urządzeń dostarcza solidnych dowodów na to, że aplikacja webowa działa poprawnie i efektywnie zarządza zasobami. Nie wykryto żadnych błędów ani problemów wydajności, co świadczy o wysokiej jakości kodu i starannym procesie testowania i optymalizacji.

4.3.2. Testy bezpieczeństwa i firewalla

W dzisiejszym dynamicznie rozwijającym się świecie cyfrowym, bezpieczeństwo informacji stało się kluczowym elementem każdej organizacji. W miarę jak technologie stają się coraz bardziej złożone, tak samo rosną zagrożenia i potencjalne wektory ataku. Rozdział 4.3.2 koncentruje się na różnych technikach i narzędziach wykorzystywanych do testowania bezpieczeństwa, ze szczególnym naciskiem na analizę i ocenę firewalla oraz aplikacji webowej hostowanej na stronie reduce.prz.edu.pl. W tej sekcji zostaną szczegółowo omówione następujące aspekty:

- **Skanowanie portów z użyciem nmap:** Ta sekcja przedstawia, jak narzędzie `nmap` może być wykorzystane do skanowania portów, identyfikacji otwartych usług i oceny ogólnego stanu zabezpieczeń sieci. Skanowanie portów jest kluczowym elementem w rozpoznawaniu powierzchni ataku i pomaga w zrozumieniu, jakie wektory ataku mogą być dostępne.
- **Symulacja ataku XSS:** W tej części omówione zostanie, jak przeprowadzono symulowany atak XSS na stronie reduce.prz.edu.pl i jaka była reakcja systemów bezpieczeństwa. Atak XSS polega na wstrzyknięciu złośliwego kodu JavaScript do

strony internetowej, co może prowadzić do kradzieży danych i innych niebezpiecznych działań.

- **Symulacja ataku SQL Injection:** przeprowadzona także w celu sprawdzenia reakcji systemu detekcji intruzów Snort, który został wyposażony w taką regułę
- **Symulacja ataku DDoS:** Ostatnia część tego podrozdziału skoncentruje się na symulacji ataku DDoS (Distributed Denial of Service) na stronie reduce.prz.edu.pl. Ataki DDoS mogą sparaliżować stronę internetową poprzez zalew jej ogromną ilością ruchu, co prowadzi do niedostępności usługi.

4.3.2.1 Skanowanie portów z użyciem nmap

Wyniki skanowania portów aplikacji REDUCE za pomocą narzędzia nmap przedstawiają dokładny obraz stanu sieci i zabezpieczeń. W pierwszym skanie, porty takie jak FTP (21), SSH (22), Telnet (23), POP3 (110) i IMAP (143) są zablokowane, co wskazuje, że firewall blokuje te porty. Porty HTTP (80) i HTTPS (443) są otwarte, co jest standardem dla serwera hostującego stronę internetową.

W drugim skanie, inne porty, takie jak SMTP (25), Domain (53), MSRPC (135), NetBIOS-SSN (139), MySQL (3306) i różne inne, są zamknięte. Oznacza to, że te porty nie są aktywnie słuchane na serwerze, co jest dowodem na dobrą praktykę zabezpieczeń, gdzie tylko niezbędne porty są otwarte, a reszta jest zamknięta.

Ogólnie rzecz biorąc, wyniki te potwierdzają, że aplikacja REDUCE posiada dobrze skonfigurowany firewall, który zamyka lub filtruje niepotrzebne porty. Otwarte porty są związane z typowymi usługami internetowymi, co jest standardem dla hostowanej strony internetowej.

4.3.2.2 Symulacja ataku XSS i reakcji systemu Snort

Symulacja ataku XSS jest ważnym elementem testowania bezpieczeństwa aplikacji webowej. W tym przypadku skoncentrowaliśmy się na wykryciu ataku przez system Snort, wykorzystując rzeczywiste narzędzia i techniki stosowane w etycznym hakowaniu. Takie testy powinny być przeprowadzane tylko w kontrolowanym środowisku i zgodnie z obowiązującymi przepisami prawa oraz najlepszymi praktykami etycznego hakowania. Tworzenie złośliwego kodu jest kluczowym etapem w symulacji ataku XSS.

W tym przypadku symulowano lukę w walidacji, więc złośliwy kod musiał być starannie zaprojektowany, aby ominąć zabezpieczenia. Przykładowy prosty złośliwy kod JavaScript, który został stworzony: `<script>alert('XSS Attack!');</script>`. Ten prosty skrypt wykorzystuje funkcję `alert` do wyświetlenia okna dialogowego

z wiadomością "XSS Attack!". Jest to często stosowany sposób na demonstrację, że kod JavaScript może być wykonany na stronie. Następnie wykorzystano narzędzie Burp Suite [536], aby zmanipulować żądanie HTTP. Proces ten obejmuje kilka kroków:

- Za pomocą funkcji proxy w Burp Suite przechwytyje się żądanie HTTP wysyłane do strony `reduce.prz.edu.pl`.
- W przechwyconym żądaniu jest lokalizowane pole macierzy, w którym można wstrzyknąć złośliwy kod. Następnie jest wklejany złośliwy kod JavaScript w odpowiednie miejsce w polu macierzy.
- Po wstrzyknięciu złośliwego kodu, jest wysyłane zmodyfikowane żądanie z powrotem do serwera.
- Następuje obserwacja, jak serwer reaguje na zmodyfikowane żądanie, aby zrozumieć, czy atak się powiódł.

Struktura typowego wpisu w logach Snort obejmowała sygnaturę alertu, klasyfikację, priorytet, datę i czas, adresy IP i porty, szczegóły protokołu oraz link do większej ilości informacji. Podczas symulacji ataku XSS, Snort wygenerował wpis w logach widoczny na Listingu 4.46.

Analiza wygenerowanego wpisu wykazała, że sygnatura alertu wskazuje na unikalny identyfikator reguły, klasyfikacja "Web Application Attack" wskazuje na rodzaj ataku, a adresy IP i porty oraz szczegóły protokołu dostarczają dodatkowych informacji na temat wykrytego ruchu. W eksperymencie, logi Snort były kluczowym narzędziem do monitorowania i analizy ruchu sieciowego. Pomogły one w wykryciu i zrozumieniu próby ataku XSS oraz w podejmowaniu odpowiednich działań w celu zabezpieczenia systemu.

4.3.2.3 Symulacja ataku SQL Injection i reakcji systemu Snort

Atak SQL Injection może przebiegać analogicznie do symulowanego ataku XSS, z pewnymi różnicami związanymi z naturą ataku.

Podczas przeprowadzania symulowanego ataku SQL Injection na stronie testowej, wykorzystano narzędzie Burp Suite do przechwytywania i modyfikowania żądań HTTP. Po skonfigurowaniu Burp Suite jako proxy, przechwycono żądanie, w którym wstrzyknięto złośliwe zapytanie SQL. Następnie żądanie zostało przesłane do serwera, a odpowiedź została zanalizowana w celu oceny skutków ataku. Mimo precyzyjnej manipulacji żądaniem, atak nie powiódł się, ponieważ strona testowa nie była połączona z żadną bazą danych.

Wyciąg z logu narzędzia Snort (Listing 4.48) przedstawia wykrycie potencjalnego ataku SQL Injection.

4.3.2.4 Symulacja ataku DDoS

Należy zaznaczyć, że w trakcie symulacji ataku DDoS, mechanizmy anty-DDoS wbudowane w firewall Cisco zareagowały na niezwykle wzrost ruchu sieciowego. System zidentyfikował to jako potencjalny atak i automatycznie dodał używane w ataku adresy IP do listy permanentnie blokowanych adresów. Ta reakcja była zgodna z oczekiwaniami i pokazała, że system obrony przed DDoS działał prawidłowo, identyfikując i blokując atak. Warto podkreślić także, że opisane akcje, takie jak atak z wykorzystaniem narzędzia LOIC opisanego poniżej, są legalne wyłącznie wtedy, gdy są przeprowadzane na własnej witrynie lub w środowisku, do którego badacz ma odpowiednie uprawnienia. Przeprowadzanie takich ataków na obcych witrynach, bez wyraźnej zgody właściciela, jest nielegalne i może prowadzić do poważnych konsekwencji prawnych [201].

Dodatkowo, warto zauważyć, że narzędzie LOIC jest wykrywane przez systemy ochrony Windows jako oprogramowanie złośliwe (malware). Jest to spowodowane jego zdolnością do generowania ataków DoS/DDoS, co może być wykorzystywane w nielegalny sposób. Atak z wykorzystaniem narzędzia LOIC [196] (Low Orbit Ion Cannon) to popularna metoda przeprowadzania ataku typu Denial of Service (DoS) lub Distributed Denial of Service (DDoS). LOIC to otwartoźródłowe narzędzie, które może generować duże ilości ruchu sieciowego, skierowanego na określony cel. Jest łatwe w obsłudze i często wykorzystywane do testowania odporności systemów na ataki DoS/DDoS [197-200]. Oto szczegółowy opis ataku, który został przeprowadzony na stronie reduce.prz.edu.pl:

- Po uruchomieniu narzędzia LOIC, wprowadzono adres IP strony reduce.prz.edu.pl jako cel ataku. Wybrano typ ataku TCP i ustawiono parametry, takie jak prędkość i ilość równoczesnych połączeń.
- Atak został rozpoczęty przez kliknięcie przycisku "IMMA CHARGIN MAH LAZER". LOIC zaczął wysyłać żądania do strony reduce.prz.edu.pl z określoną prędkością i ilością połączeń.
- W trakcie ataku LOIC dostarczał informacje o stanie ataku, takie jak liczba wysłanych żądań, liczba aktywnych i zakończonych połączeń. Te dane były kluczowe dla zrozumienia wpływu ataku na system.

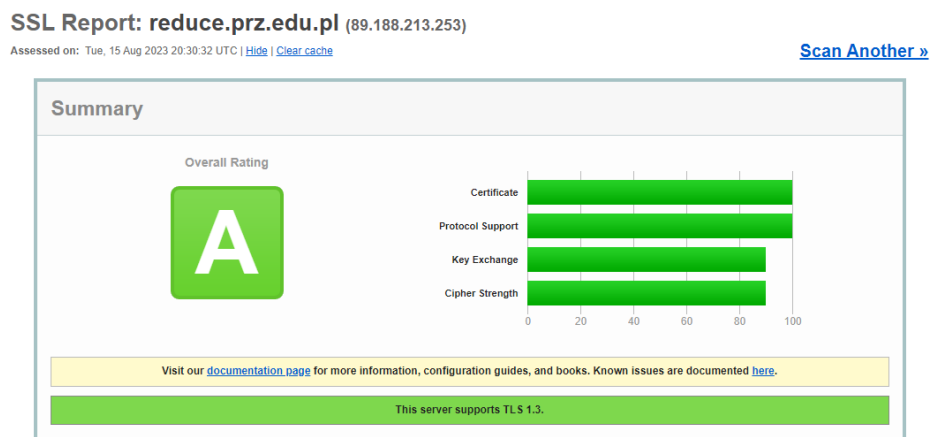
Dzięki blokadzie firewalla Cisco, atak na stronę się nie powiódł, blokując adres IP komputera dokonującego ataku. Atak został także ujęty w logach Snort, widocznych na Listingu 4.48.

Symulacja ataku DDoS i reakcja Snorta na niego dostarczyły cennych informacji na temat tego, jak system może reagować na rzeczywiste zagrożenia. Dzięki temu można

dostosować reguły Snorta i inne mechanizmy obronne, aby lepiej chronić system przed potencjalnymi atakami DDoS w przyszłości.

4.3.3 Testy TLS/SSL

Testy certyfikatu TLS/SSL służą do oceny i weryfikacji konfiguracji serwera, na którym działa strona internetowa, w celu zapewnienia, że komunikacja między serwerem a klientem jest szyfrowana i zabezpieczona.



Rys. 4.18 Sumaryczna ocena zastosowanego certyfikatu TLS/SSL

Server Key and Certificate #1

Subject	reduce.prz.edu.pl Fingerprint SHA256: 9c33def69d1e851e2f9d553d10d7bea4946ca16e87aec52df2ae74c70f803974 Pin SHA256: 3J07LiorKOGJxpG6J7aTwly0WsbHhbKU4ZHS70XkS6k=
Common names	reduce.prz.edu.pl
Alternative names	reduce.prz.edu.pl
Serial Number	046bdfd3873c7c710bc44dfc28233bb4ee01
Valid from	Sun, 13 Aug 2023 19:53:05 UTC
Valid until	Sat, 11 Nov 2023 19:53:04 UTC (expires in 2 months and 26 days)
Key	RSA 2048 bits (e 65537)
Weak key (Debian)	No
Issuer	R3 AIA: http://r3.o.leencr.org/
Signature algorithm	SHA256withRSA
Extended Validation	No
Certificate Transparency	Yes (certificate)
OCSF Must Staple	No
Revocation information	OCSF OCSF: http://r3.o.leencr.org
Revocation status	Good (not revoked)
DNS CAA	No (more info)
Trusted	Yes Mozilla Apple Android Java Windows

Rys. 4.19 Sekcja „Server Key and Certificate #2”

Additional Certificates (if supplied)	
Certificates provided	3 (3954 bytes)
Chain issues	None
#2	
Subject	R3 Fingerprint SHA256: 67add1166b020ae61b8f5fc96813c04c2aa589960796865572a3c7e737613dfd Pin SHA256: jQJTbIh0grw0/1TkHSumWb+F80Ggogr621gT3PvPKG0=
Valid until	Mon, 15 Sep 2025 16:00:00 UTC (expires in 2 years)
Key	RSA 2048 bits (e 65537)
Issuer	ISRG Root X1
Signature algorithm	SHA256withRSA
#3	
Subject	ISRG Root X1 Fingerprint SHA256: 6d99fb265eb1c5b3744765fbc648f3cd8e1bffa4dc4c2f99b9d47cf7f1c24f Pin SHA256: C5+lpZ7lcVvmwQIMcRIPbsQIWLABXhQzejna0wHFr8M=
Valid until	Mon, 30 Sep 2024 18:14:03 UTC (expires in 1 year and 1 month)
Key	RSA 4096 bits (e 65537)
Issuer	DST Root CA X3
Signature algorithm	SHA256withRSA

Rys. 4.20 Sekcja „Additional Certificates”

Certification Paths	
<div style="display: flex; border-bottom: 1px solid #ccc;"> <div style="background-color: #0070c0; color: white; padding: 2px 5px; margin-right: 5px;">Mozilla</div> <div style="padding: 2px 5px; margin-right: 5px;">Apple</div> <div style="padding: 2px 5px; margin-right: 5px;">Android</div> <div style="padding: 2px 5px; margin-right: 5px;">Java</div> <div style="padding: 2px 5px;">Windows</div> </div>	
Path #1: Trusted	
1	<div style="display: flex; justify-content: space-between;"> <div>Sent by server</div> <div>reduce.prz.edu.pl</div> </div> <p>Fingerprint SHA256: 9c33def69d1e851e2f9d553d10d7bea4946ca16e87aec52df2ae74c70f803974 Pin SHA256: 3J07LlofKOgJxpG6J7aTwly0WsbHhbKudZhS70XkS6k= RSA 2048 bits (e 65537) / SHA256withRSA</p>
2	<div style="display: flex; justify-content: space-between;"> <div>Sent by server</div> <div>R3</div> </div> <p>Fingerprint SHA256: 67add1166b020ae61b8f5fc96813c04c2aa589960796865572a3c7e737613dfd Pin SHA256: jQJTbIh0grw0/1TkHSumWb+F80Ggogr621gT3PvPKG0= RSA 2048 bits (e 65537) / SHA256withRSA</p>
3	<div style="display: flex; justify-content: space-between;"> <div>In trust store</div> <div>ISRG Root X1 Self-signed</div> </div> <p>Fingerprint SHA256: 96bcec06264976f37460779acf28c5a7cfe8a3c0aae11a8ffcee05c0bdddf08c6 Pin SHA256: C5+lpZ7lcVvmwQIMcRIPbsQIWLABXhQzejna0wHFr8M= RSA 4096 bits (e 65537) / SHA256withRSA</p>
Path #2: Not trusted (invalid certificate [Fingerprint SHA256: 0687260331a72403d909f105e69bcfd32e1bd2493ffc6d9206d11bcd6770739])	
1	<div style="display: flex; justify-content: space-between;"> <div>Sent by server</div> <div>reduce.prz.edu.pl</div> </div> <p>Fingerprint SHA256: 9c33def69d1e851e2f9d553d10d7bea4946ca16e87aec52df2ae74c70f803974 Pin SHA256: 3J07LlofKOgJxpG6J7aTwly0WsbHhbKudZhS70XkS6k= RSA 2048 bits (e 65537) / SHA256withRSA</p>
2	<div style="display: flex; justify-content: space-between;"> <div>Sent by server</div> <div>R3</div> </div> <p>Fingerprint SHA256: 67add1166b020ae61b8f5fc96813c04c2aa589960796865572a3c7e737613dfd Pin SHA256: jQJTbIh0grw0/1TkHSumWb+F80Ggogr621gT3PvPKG0= RSA 2048 bits (e 65537) / SHA256withRSA</p>
3	<div style="display: flex; justify-content: space-between;"> <div>Sent by server</div> <div>ISRG Root X1</div> </div> <p>Fingerprint SHA256: 6d99fb265eb1c5b3744765fbc648f3cd8e1bffa4dc4c2f99b9d47cf7f1c24f Pin SHA256: C5+lpZ7lcVvmwQIMcRIPbsQIWLABXhQzejna0wHFr8M= RSA 4096 bits (e 65537) / SHA256withRSA</p>
4	<div style="display: flex; justify-content: space-between;"> <div>In trust store</div> <div>DST Root CA X3 Self-signed</div> </div> <p>Fingerprint SHA256: 0687260331a72403d909f105e69bcfd32e1bd2493ffc6d9206d11bcd6770739 Pin SHA256: Vjs8r4z+80vNcr1YKepWQboSIRi63WsWXhIMN+eWYs= RSA 2048 bits (e 65537) / SHA1withRSA Valid until: Thu, 30 Sep 2021 14:01:15 UTC EXPIRED Weak or insecure signature, but no impact on root certificate</p>

Rys. 4.21 Sekcja „Certification Paths”

Protocols		
TLS 1.3		Yes
TLS 1.2		Yes
TLS 1.1		No
TLS 1.0		No
SSL 3		No
SSL 2		No

Cipher Suites		
# TLS 1.3 (server has no preference)		
TLS_AES_128_GCM_SHA256 (0x1301)	ECDH x25519 (eq. 3072 bits RSA) FS	128
TLS_AES_256_GCM_SHA384 (0x1302)	ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_CHACHA20_POLY1305_SHA256 (0x1303)	ECDH x25519 (eq. 3072 bits RSA) FS	256
# TLS 1.2 (server has no preference)		
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)	ECDH secp521r1 (eq. 15360 bits RSA) FS WEAK	128
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (0x9e)	DH 2048 bits FS	128
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)	ECDH secp521r1 (eq. 15360 bits RSA) FS	128
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (0x9f)	DH 2048 bits FS	256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)	ECDH secp521r1 (eq. 15360 bits RSA) FS	256
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xc0a8)	ECDH secp521r1 (eq. 15360 bits RSA) FS	256

Rys. 4.22 Sekcje „Protocols” oraz „Cipher Suites”

Przeprowadzanie takich testów ma na celu identyfikację potencjalnych luk i słabości w konfiguracji, które mogą zostać wykorzystane przez nieuprawnione osoby do uzyskania dostępu do wrażliwych danych. Obejmuje to sprawdzanie, czy serwer obsługuje najnowsze i najbezpieczniejsze protokoły, jak również czy nie ma żadnych znanych podatności, które mogą zostać wykorzystane do ataku.

Oprócz zwiększenia bezpieczeństwa, testy te są również ważne dla zgodności z różnymi regulacjami i standardami branżowymi. Mogą być wymagane przez różne organy regulacyjne lub ciała certyfikujące jako część procesu certyfikacji bezpieczeństwa. Wreszcie, testy te budują zaufanie użytkowników do strony internetowej lub usługi online. Widząc, że strona korzysta z certyfikatu SSL i że została ona odpowiednio przetestowana, użytkownicy mogą czuć się pewniej, korzystając z usług, i wiedzieć, że ich dane są chronione. W erze, w której obawy dotyczące prywatności są na porządku dziennym, takie testy są nie tylko mądre z punktu widzenia technicznego, ale także kluczowe dla utrzymania reputacji i zaufania klientów.

Do przeprowadzenia powyższego testu certyfikatu SSL wykorzystano narzędzie dostępne na stronie ssllabs.com. Jest to renomowane i powszechnie stosowane narzędzie, które oferuje szczegółową analizę konfiguracji SSL/TLS serwera. Dzięki niemu można nie tylko ocenić ogólną jakość konfiguracji, ale także zidentyfikować konkretne aspekty, które mogą wymagać poprawy.



Handshake Simulation

Android 4.4.2	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp521r1 FS
Android 5.0.0	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	ECDH secp521r1 FS
Android 6.0	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	ECDH secp256r1 FS
Android 7.0	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ECDH x25519 FS
Android 8.0	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ECDH x25519 FS
Android 8.1	-	TLS 1.3	TLS_CHACHA20_POLY1305_SHA256	ECDH x25519 FS
Android 9.0	-	TLS 1.3	TLS_CHACHA20_POLY1305_SHA256	ECDH x25519 FS
BingPreview Jan 2015	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp521r1 FS
Chrome 49 / XP SP3	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	ECDH secp256r1 FS
Chrome 69 / Win 7 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	ECDH x25519 FS
Chrome 70 / Win 10	-	TLS 1.3	TLS_AES_128_GCM_SHA256	ECDH x25519 FS
Chrome 80 / Win 10 R	-	TLS 1.3	TLS_AES_128_GCM_SHA256	ECDH x25519 FS
Firefox 31.3.0 ESR / Win 7	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	ECDH secp256r1 FS
Firefox 47 / Win 7 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	ECDH secp256r1 FS
Firefox 49 / XP SP3	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	ECDH secp256r1 FS
Firefox 62 / Win 7 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	ECDH x25519 FS
Firefox 73 / Win 10 R	-	TLS 1.3	TLS_AES_128_GCM_SHA256	ECDH x25519 FS
Googlebot Feb 2018	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	ECDH x25519 FS
IE 11 / Win 7 R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	ECDH secp256r1 FS
IE 11 / Win 8.1 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	ECDH secp256r1 FS
IE 11 / Win Phone 8.1 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	ECDH secp256r1 FS
IE 11 / Win Phone 8.1 Update R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	ECDH secp256r1 FS
IE 11 / Win 10 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Edge 15 / Win 10 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
Edge 16 / Win 10 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
Edge 18 / Win 10 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
Edge 13 / Win Phone 10 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Java 8u161	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	ECDH secp256r1 FS
Java 11.0.3	-	TLS 1.3	TLS_AES_128_GCM_SHA256	ECDH secp256r1 FS
Java 12.0.1	-	TLS 1.3	TLS_AES_128_GCM_SHA256	ECDH secp256r1 FS
OpenSSL 1.0.1j R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp521r1 FS
OpenSSL 1.0.2s R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
OpenSSL 1.1.0k R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
OpenSSL 1.1.1c R	-	TLS 1.3	TLS_AES_256_GCM_SHA384	ECDH x25519 FS
Safari 6 / iOS 6.0.1	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	ECDH secp256r1 FS
Safari 7 / iOS 7.1 R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	ECDH secp256r1 FS
Safari 7 / OS X 10.9 R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	ECDH secp256r1 FS
Safari 8 / iOS 8.4 R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	ECDH secp256r1 FS
Safari 8 / OS X 10.10 R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	ECDH secp256r1 FS
Safari 9 / iOS 9 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Safari 9 / OS X 10.11 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Safari 10 / iOS 10 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Safari 10 / OS X 10.12 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Safari 12.1.2 / MacOS 10.14.6 Beta R	-	TLS 1.3	TLS_CHACHA20_POLY1305_SHA256	ECDH x25519 FS

Rys. 4.23 Rezultaty symulacji protokołu handshake dla certyfikatu



Protocol Details

	Unable to perform this test due to an internal error. (1) For a better understanding of this test, please read this longer explanation (2) Key usage data kindly provided by the Censys network search engine; original DROWN website here (3) Censys data is only indicative of possible key and certificate reuse; possibly out-of-date and not complete INTERNAL ERROR: test.drownattack.com INTERNAL ERROR: test.drownattack.com
DROWN	
Secure Renegotiation	Supported
Secure Client-Initiated Renegotiation	No
Insecure Client-Initiated Renegotiation	No
BEAST attack	Mitigated server-side (more info)
POODLE (SSLv3)	No, SSL 3 not supported (more info)
POODLE (TLS)	No (more info)
Zombie POODLE	No (more info) TLS 1.2 : 0xc013
GOLDENDOODLE	No (more info) TLS 1.2 : 0xc013
OpenSSL 0-Length	No (more info) TLS 1.2 : 0xc013
Sleeping POODLE	No (more info) TLS 1.2 : 0xc013
Downgrade attack prevention	Yes, TLS_FALLBACK_SCSV supported (more info)
SSL/TLS compression	No
RC4	No
Heartbeat (extension)	No
Heartbleed (vulnerability)	No (more info)
Ticketbleed (vulnerability)	No (more info)
OpenSSL CCS vuln. (CVE-2014-0224)	No (more info)
OpenSSL Padding Oracle vuln. (CVE-2016-2107)	No (more info)
ROBOT (vulnerability)	No (more info)
Forward Secrecy	Yes (with most browsers) ROBUST (more info)
ALPN	Yes http/1.1
NPN	Yes http/1.1
Session resumption (caching)	Yes
Session resumption (tickets)	No
OCSP stapling	No
Strict Transport Security (HSTS)	No
HSTS Preloading	Not in: Chrome Edge Firefox IE
Public Key Pinning (HPKP)	No (more info)
Public Key Pinning Report-Only	No
Public Key Pinning (Static)	Unknown
Long handshake intolerance	No
TLS extension intolerance	No
TLS version intolerance	No
Incorrect SNI alerts	No
Uses common DH primes	No
DH public server param (Ys) reuse	No
ECDH public server param reuse	No
Supported Named Groups	secp256r1, secp384r1, secp521r1, x25519, x448 (Server has no preferences)
SSL 2 handshake compatibility	No
0-RTT enabled	No

Rys. 4.24 Szczegóły obsługiwanych protokołów

HTTP Requests	
1 https://reduce.prz.edu.pl/ (HTTP/1.1 200 OK)	
Server	nginx/1.18.0 (Ubuntu)
Date	Tue, 15 Aug 2023 20:29:24 GMT
1 Content-Type	text/html; charset=utf-8
Content-Length	30897
Connection	close
Miscellaneous	
Test date	Tue, 15 Aug 2023 20:29:10 UTC
Test duration	81.370 seconds
HTTP status code	200
HTTP server signature	nginx/1.18.0 (Ubuntu)
Server hostname	-

Rys. 4.25 Informacje dodatkowe na temat certyfikatu TLS/SSL

Narzędzie ssllabs.com [202] jest znane ze swojej zdolności do przeprowadzania kompleksowych testów, które obejmują różne aspekty konfiguracji, takie jak wsparcie protokołu, siła szyfrowania, szczegóły certyfikatu, konfiguracja, zabezpieczenia i wiele innych [203-206]. Dzięki temu jest to nieocenione narzędzie dla administratorów, deweloperów i profesjonalistów ds. bezpieczeństwa, którzy chcą upewnić się, że ich serwery są odpowiednio skonfigurowane i zabezpieczone.

Certyfikat otrzymał ocenę "A" (Rys. 4.18). Sekcja "Server Key and Certificate #1" (Rys. 4.19) zawiera szczegółowe informacje na temat klucza serwera i certyfikatu dla domeny reduce.prz.edu.pl. Oto podsumowanie tej sekcji:

- **Podmiot:** Domena, do której odnosi się certyfikat, to reduce.prz.edu.pl.
- **Odcisk palca SHA256:** Unikalny identyfikator certyfikatu, który może być używany do weryfikacji jego autentyczności [207].
- **Pin SHA256:** Jest to dodatkowy sposób identyfikacji certyfikatu, używany w niektórych zastosowaniach bezpieczeństwa [208].
- **Wspólne nazwy i alternatywne nazwy:** Odnoszą się do domeny, do której certyfikat jest wydany.
- **Numer seryjny:** Unikalny numer identyfikujący certyfikat.
- **Ważność:** Certyfikat jest ważny od 13 sierpnia 2023 r. do 11 listopada 2023 r. (w chwili oddania pracy do druku, tj. 15 września 2023 r. jest ważny)
- **Klucz:** Wykorzystuje RSA 2048 bitów, co jest standardową siłą klucza [209]. Nie jest to słaby klucz (np. związany z tzw. problemem Debian [210]).

- **Wydawca:** Wydawcą certyfikatu jest R3 [211], a ścieżka dostępu do AIA (Authority Information Access [212]) jest dostępna online.
- **Algorytm podpisu:** Wykorzystuje algorytm SHA256 z RSA [215], który jest powszechnie stosowany i uważany za bezpieczny [216].
- **Walidacja rozszerzona:** Brak, co oznacza, że certyfikat nie przeszedł dodatkowych kontroli tożsamości [218].
- **Przejrzystość certyfikatu:** Tak, co oznacza, że certyfikat jest zarejestrowany w publicznych rejestrach [217].
- **OCSP Must Staple:** Nie, co oznacza, że serwer nie musi dołączać informacji o statusie unieważnienia certyfikatu [219].
- **Informacje o unieważnieniu:** OCSP (Online Certificate Status Protocol) jest używany do sprawdzania statusu certyfikatu, który jest dobry (nie unieważniony) [220].
- **DNS CAA:** Brak, co wiąże się z brakiem dodatkowych zabezpieczeń związanych z autoryzacją certyfikatu [221].
- **Zaufanie:** Certyfikat jest zaufany przez główne platformy, takie jak Mozilla, Apple, Android, Java i Windows.

Sekcja "Additional Certificates" zawiera informacje o dodatkowych certyfikatach dostarczonych w łańcuchu, który składa się z trzech certyfikatów o łącznym rozmiarze 3954 bajtów (Rys. 4.20). Nie stwierdzono żadnych problemów z łańcuchem. Drugi certyfikat, wydany przez R3, ma ważność do 15 września 2025 r. i używa 2048-bitowego klucza RSA z algorytmem podpisu SHA256 z RSA. Trzeci certyfikat, wydany przez ISRG Root X1 [222], ma ważność do 30 września 2024 r. i używa 4096-bitowego klucza RSA z tym samym algorytmem podpisu. Obydwa certyfikaty są częścią zaufanego łańcucha, który kończy się certyfikatem DST Root CA X3 [222].

Sekcja "Certification Paths" (Rys. 4.21) przedstawia dwie ścieżki certyfikacji dla różnych platform, takich jak Mozilla, Apple, Android, Java i Windows (dla wszystkich platform uzyskane rezultaty testu są takie same). Ścieżka certyfikacji, znana również jako łańcuch certyfikacji, to sekwencja certyfikatów służących do łączenia danego certyfikatu z zaufanym certyfikatem głównym (root certificate [240]). Każdy certyfikat w ścieżce jest podpisany przez następny certyfikat, aż do osiągnięcia zaufanego certyfikatu głównego, który jest samopodpisany [223]. Ścieżki te przedstawiają się następująco:

- **Ścieżka #1: Trusted**
 - Ta ścieżka jest zaufana i składa się z trzech certyfikatów:

- Certyfikat serwera dla reduce.prz.edu.pl z kluczem RSA 2048 bitów i algorytmem SHA256-RSA.
 - Certyfikat R3 z kluczem RSA 2048 bitów i algorytmem SHA256-RSA.
 - Samopodpisany certyfikat ISRG Root X1 z kluczem RSA 4096 bitów i algorytmem SHA256-RSA.
- **Ścieżka #2: Not trusted**
 - Ta ścieżka nie jest zaufana z powodu nieważnego certyfikatu. Składa się z czterech certyfikatów:
 - Certyfikat serwera dla reduce.prz.edu.pl z kluczem RSA 2048 bitów i algorytmem SHA256-RSA.
 - Certyfikat R3 z kluczem RSA 2048 bitów i algorytmem SHA256-RSA.
 - Certyfikat ISRG Root X1 z kluczem RSA 4096 bitów i algorytmem SHA256-RSA.
 - Samopodpisany certyfikat DST Root CA X3 z kluczem RSA 2048 bitów i algorytmem SHA1-RSA, który wygaś 30 września 2021 r.

Ostatni certyfikat w ścieżce #2 jest oznaczony jako słaby lub niezabezpieczony, ale nie ma to wpływu na certyfikat główny. Sekcja "Konfiguracja" rozpoczyna się od szczegółowych informacji na temat protokołów i zestawów szyfrów obsługiwanych przez serwer (Rys. 4.22). Oto podsumowanie:

- **Protokoły**
 - **TLS 1.3:** Obsługiwany, najnowsza wersja protokołu TLS, zapewniająca najlepsze zabezpieczenia [213].
 - **TLS 1.2:** Obsługiwany, wcześniejsza wersja protokołu, nadal uważana za bezpieczną [214].
- **TLS 1.1, TLS 1.0, SSL 3, SSL 2:** Nie są obsługiwane, co jest pożądane, ponieważ te wersje są uważane za niebezpieczne i narażone na różne ataki [224-227].
- **Zestawy szyfrów**
 - **TLS 1.3:** Serwer obsługuje trzy zestawy szyfrów dla TLS 1.3, wszystkie z równoważnym poziomem bezpieczeństwa 3072-bitowego RSA. Zestawy te to [228-232]:
 - TLS_AES_128_GCM_SHA256 (128-bitowy)
 - TLS_AES_256_GCM_SHA384 (256-bitowy)

- TLS_CHACHA20_POLY1305_SHA256 (256-bitowy)
- **TLS 1.2:** Serwer obsługuje różne zestawy szyfrów dla TLS 1.2, w tym zarówno silne, jak i słabe (oznaczone jako "WEAK").
 - Niektóre z nich wykorzystują ECDH [239] z równoważnym poziomem bezpieczeństwa 15360-bitowego RSA, a inne DH [240] z 2048-bitowym kluczem.
 - Zestawy te obejmują różne algorytmy szyfrowania, takie jak AES [236] i CHACHA20 [237], z różnymi długościami klucza (128 lub 256 bitów).

Wyłączenie starszych i słabszych wersji protokołów i szyfrów jest zgodne z najlepszymi praktykami w zakresie bezpieczeństwa [233-235]. Sekcja "Handshake Simulation" konfiguracji zaś (Rys. 4.23) przedstawia, jak różne systemy klientów (takie jak różne wersje systemów operacyjnych, przeglądarek i botów) będą komunikować się z serwerem podczas nawiązywania połączenia TLS (Transport Layer Security). Symulacja ta jest kluczowa do zrozumienia, jakie protokoły i zestawy szyfrów będą używane przez różne aplikacje klienckie, co może mieć wpływ na bezpieczeństwo i wydajność połączenia [241-245]. Podsumowanie symulacji handshake dla certyfikatu reduce.prz.edu.pl przedstawia się następująco:

- **Android:** Różne wersje Androida obsługują różne wersje protokołu TLS, od TLS 1.2 do TLS 1.3, z różnymi zestawami szyfrów. Wersje od 8.1 wzwyż obsługują nowoczesny TLS 1.3.
- **BingPreview, Yahoo Slurp, YandexBot:** Boty wyszukiwarek korzystają z TLS 1.2 z różnymi zestawami szyfrów.
- **Chrome:** Różne wersje Chrome na różnych systemach operacyjnych obsługują zarówno TLS 1.2, jak i TLS 1.3.
- **Firefox:** Podobnie jak Chrome, różne wersje Firefox obsługują zarówno TLS 1.2, jak i TLS 1.3.
- **Internet Explorer (IE) i Edge:** Te przeglądarki na różnych wersjach systemu Windows obsługują głównie TLS 1.2, z różnymi zestawami szyfrów.
- **Java:** Różne wersje środowiska wykonawczego Java obsługują TLS 1.2 i TLS 1.3.
- **OpenSSL:** Różne wersje OpenSSL obsługują TLS 1.2 i TLS 1.3.
- **Safari:** Różne wersje Safari na iOS i macOS obsługują zarówno TLS 1.2, jak i TLS 1.3.
- **Nieudane symulacje:** Niektóre starsze klienty, takie jak Android 2.3.7, IE 6 i Java 6u45, nie są w stanie nawiązać połączenia z serwerem z powodu niekompatybilności protokołu.

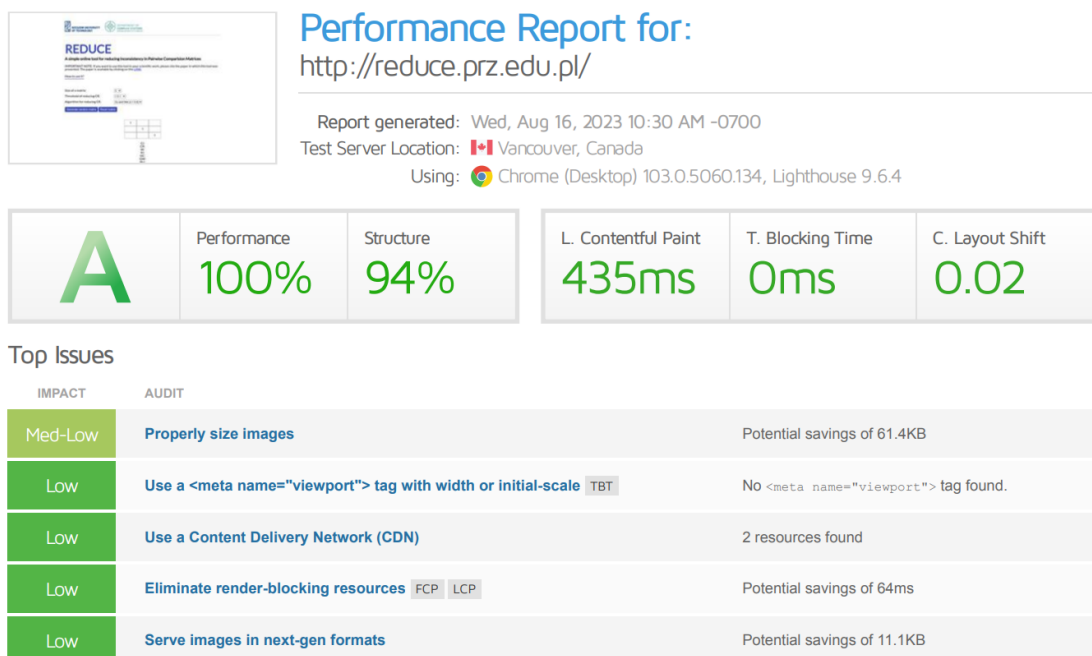
Wszystkie symulowane połączenia używają klucza RSA 2048 (SHA256) i obsługują Forward Secrecy [246-248], co zwiększa bezpieczeństwo połączenia. Ogólnie rzecz biorąc, ta sekcja pokazuje, że serwer jest skonfigurowany tak, aby obsługiwać nowoczesne i bezpieczne protokoły i zestawy szyfrów dla większości współczesnych klientów, podczas gdy starsze i mniej bezpieczne klienty mogą napotkać problemy z kompatybilnością.

Kolejna sekcja (Rys. 4.24) zawiera szczegółowe informacje na temat różnych protokołów i testów związanych z bezpieczeństwem połączenia TLS/SSL. Wyniki wskazują na to, że wiele współczesnych mechanizmów zabezpieczeń jest obsługiwanych, a znane luki i podatności zostały zminimalizowane. Oto podsumowanie kluczowych elementów:

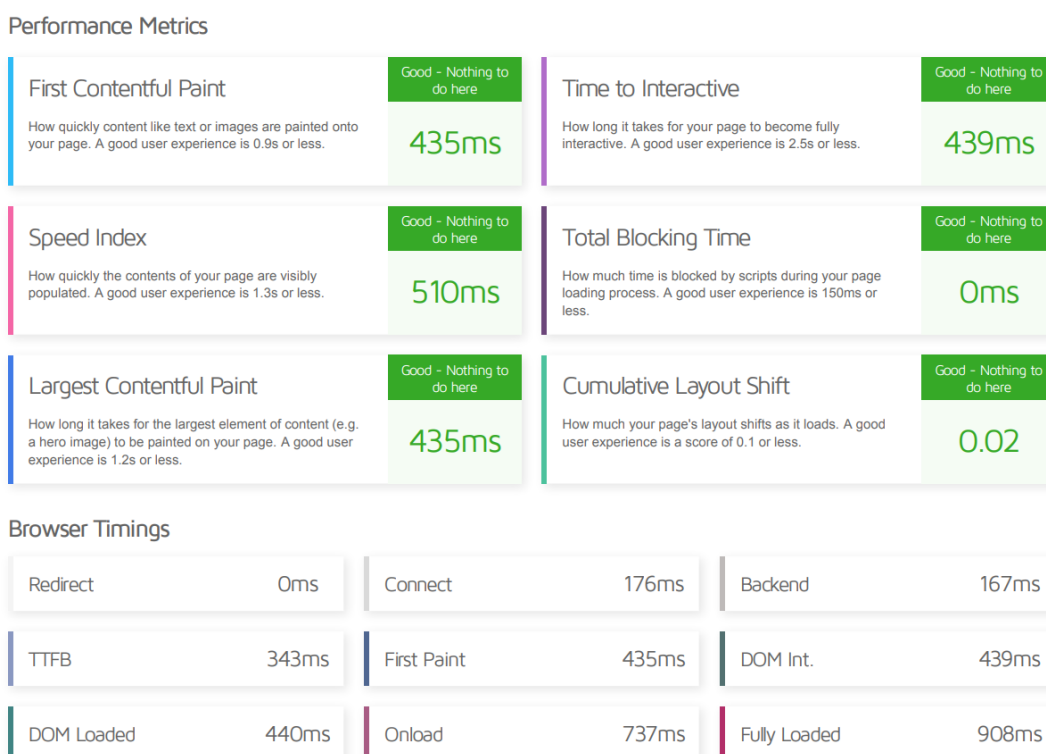
- **DROWN:** Test nie mógł zostać przeprowadzony z powodu wewnętrznego błędu. DROWN to atak, który może umożliwić złamanie szyfrowania TLS [249-250].
- **Secure Renegotiation:** Obsługiwane, co oznacza, że serwer wspiera bezpieczną renegocjację [251-253].
- **Insecure Client-Initiated Renegotiation:** Nie jest obsługiwane, co zwiększa bezpieczeństwo [251-253].
- **Atak BEAST:** Zabezpieczony po stronie serwera [254].
- **POODLE, Zombie POODLE, GOLDEN DOODLE, OpenSSL 0-Length, Sleeping POODLE:** Nie są [255-259] obsługiwane, co oznacza, że serwer nie jest podatny na te ataki.
- **Downgrade attack prevention:** Obsługiwane, co zapobiega próbom zmuszenia do korzystania z niższej wersji protokołu [260-262].
- **SSL/TLS compression, RC4, Heartbeat extension:** Nie są obsługiwane, co zwiększa bezpieczeństwo [263-272].
- **Heartbleed, Ticket bleed, OpenSSL CCS, OpenSSL Padding Oracle, ROBOT:** Nie są obecne [250, 273-285], co oznacza, że serwer nie jest podatny na te znane luki bezpieczeństwa .
- **ALPN i NPN:** Obsługiwane [286-289], co umożliwia negocjację protokołu na poziomie aplikacji.
- **Session resumption:** Obsługiwane dla buforowania, ale nie dla biletów [290-294].
- **OCSP, Strict Transport Security (HSTS), Public Key Pinning (HPKP):** Nie są obsługiwane, co może wpłynąć na bezpieczeństwo i wydajność [295-304].
- **Supported Named Groups:** Wykaz wspieranych grup nazwanych dla wymiany kluczy [305,306].
- **SSL 2 handshake, 0-RTT:** Nie są obsługiwane, co zwiększa bezpieczeństwo [307-310].

Ostatnie dwie sesje (Rys. 4.25) przedstawiają szczegóły żądania HTTP do strony reduce.prz.edu.pl, a także dodatkowe informacje na temat testu. Odpowiedź serwera była pozytywna (HTTP/1.1 200 OK), co oznacza, że żądanie zostało pomyślnie przetworzone.

4.3.4 Testy wydajności ładowania aplikacji



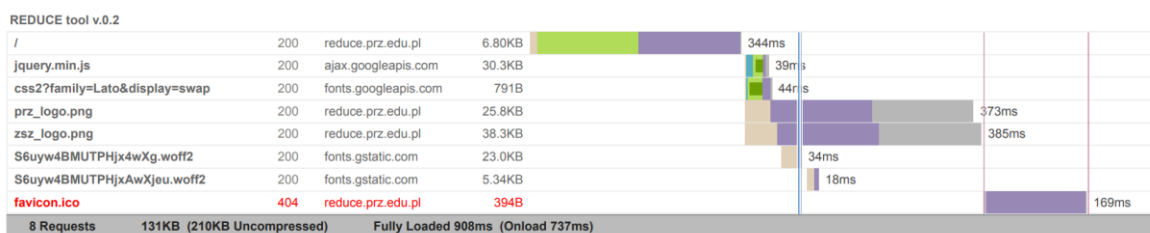
Rys. 4.26. Ocena ogólna wydajności witryny REDUCE.prz.edu.pl



Rys. 4.27. Metryki wydajności witryny REDUCE.prz.edu.pl

Summary		Performance	Structure	Waterfall	Video	History			
		Show Audits Relevant to			All	FCP	LCP	TBT	CLS
IMPACT	AUDIT								
Med-Low	Properly size images		Potential savings of 61.4KB					▼	
Low	Eliminate render-blocking resources <small>FCP LCP</small>		Potential savings of 102ms					▼	
Low	Use a <meta name="viewport"> tag with width or initial-scale <small>TBT</small>		No <meta name="viewport"> tag found.					▼	
Low	Use a Content Delivery Network (CDN)		2 resources found					▼	
Low	Serve images in next-gen formats		Potential savings of 11.1KB					▼	
Low	Reduce unused JavaScript <small>LCP</small>		Potential savings of 20.3KB					▼	
Low	Avoid an excessive DOM size <small>TBT</small>		303 elements					▼	
Low	Avoid enormous network payloads <small>LCP</small>		Total size was 132KB					▼	
Low	Reduce JavaScript execution time <small>TBT</small>		2ms spent executing JavaScript					▼	
Low	Reduce initial server response time <small>FCP LCP</small>		Root document took 168ms					▼	
Low	Avoid chaining critical requests <small>FCP LCP</small>		3 chains found					▼	
N/A	Largest Contentful Paint element <small>LCP</small>		1 element found					▼	
N/A	Avoid large layout shifts <small>CLS</small>		4 elements found					▼	
N/A	Minimize main-thread work <small>TBT</small>		Main-thread busy for 91ms					▼	
N/A	Reduce the impact of third-party code <small>TBT</small>		Total size was 60.7KB					▼	
N/A	User Timing marks and measures							▼	

Rys. 4.28. Struktura wpływu poszczególnych elementów na długość ładowania witryny



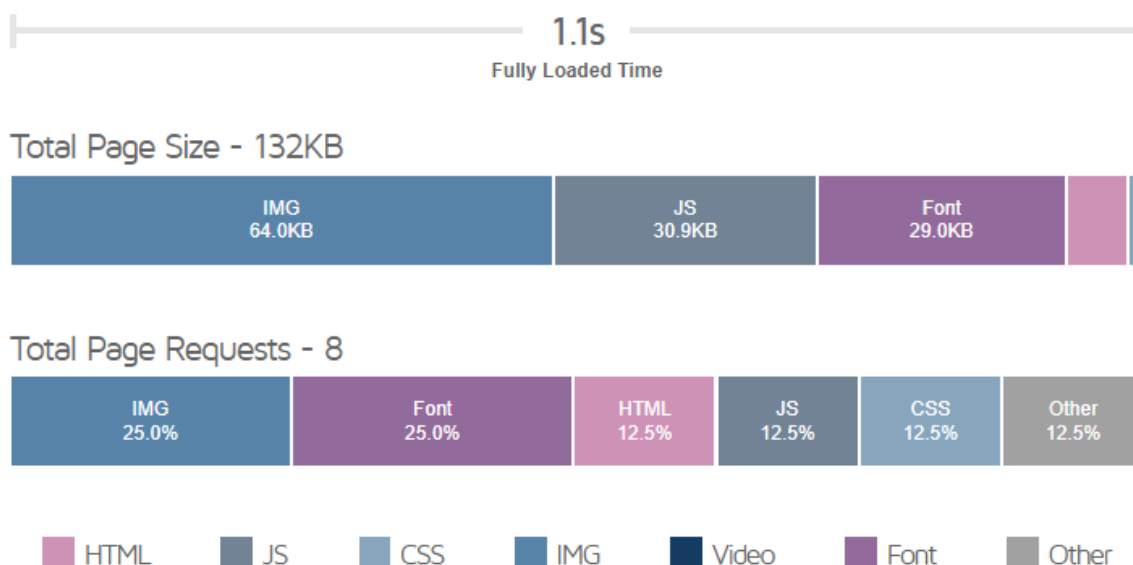
Rys. 4.29. Wykres wodospadu pokazujący czasy ładowania poszczególnych elementów

Raport ładowania witryny reduce.prz.edu.pl, przygotowany za pomocą narzędzia GTMetrix [311], dostarcza istotnych informacji na temat wydajności i struktury strony internetowej. Raport został wygenerowany we wtorek, 15 sierpnia 2023 r., o godzinie 17:24 czasu lokalnego, z serwera testowego zlokalizowanego w Vancouver w Kanadzie. Test przeprowadzono przy użyciu przeglądarki Chrome (Desktop) w wersji 103.0.5060.134 oraz narzędzia Lighthouse 9.6.4. Dlaczego wybrano GTMetrix? Został on wybrany jako narzędzie do analizy ze względu na jego wszechstronność i precyzję w ocenie wydajności

strony internetowej. Oferuje ono kompleksową analizę, łączącą różne metryki wydajności, takie jak czas ładowania, rozmiar strony, liczba żądań oraz inne kluczowe wskaźniki i jest bardzo często wybieranym narzędziem do pomiarów tego typu [312-318]. Narzędzie to wygenerowało obszerny raport [319], który zostanie opisany w dalszej części tej sekcji.

Page Details ?

Your page content is broken down into the following:



Rys. 4.30. Szczegółowe statystyki na temat użytych rodzajów elementów strony

Strona internetowa <http://reduce.prz.edu.pl/> osiągnęła imponującą ocenę A (Rys. 4.26) w teście wydajności, co jest wynikiem doskonałej optymalizacji i wysokiej jakości strony. Wyjątkowy wynik 100% w kategorii Performance świadczy o optymalnym działaniu strony, podczas gdy 94% w kategorii Structure wskazuje na dobrze zorganizowany i zoptymalizowany kod (jego elementy zostały podsumowane na Rys. 4.30). Dodatkowo, strona wykazała się natychmiastowym renderowaniem treści z Contentful Paint wynoszącym 435ms, brakiem opóźnień w Blocking Time (0ms), oraz stabilnością układu z Layout Shift wynoszącym 0.02. Te metryki razem składają się na doskonałą wydajność strony, zapewniając użytkownikom szybkie i płynne doświadczenie.

Oczywiście pod uwagę były brane także inne metryki, których wyniki dla sekcji Performance Metrics (Rys. 4.27), przedstawiono poniżej:

- **First Contentful Paint (435ms)** - First Contentful Paint (FCP) mierzy, jak szybko treść, taka jak tekst czy obrazy, jest renderowana na stronie. Jest to kluczowy wskaźnik, który odzwierciedla pierwsze wrażenie użytkownika na stronie. Dla

dobrej jakości użytkownika, FCP powinien wynosić 0,9s lub mniej [321-324]. Strona osiągnęła wynik 435 ms, co jest znakomitym rezultatem, wskazującym, że nie ma tu nic do poprawy. Użytkownicy widzą treść strony niemal natychmiast po jej wejściu.

- **Time to Interactive (439ms)** - Time to Interactive (TTI) określa, jak długo trwa, zanim strona staje się w pełni interaktywna i reaguje na działania użytkownika. Optymalny czas to 2,5 s lub mniej [325-327]. Strona osiągnęła wynik 439ms, co jest doskonałym wynikiem. Oznacza to, że użytkownicy mogą zacząć korzystać ze strony prawie natychmiast, bez żadnych opóźnień.
- **Speed Index (510ms)** - Speed Index mierzy szybkość, z jaką zawartość strony staje się widoczna dla użytkownika. Jest to kluczowy wskaźnik, który pokazuje, jak szybko użytkownik może zobaczyć pełną stronę. Optymalny *Speed Index* to 1,3 s lub mniej [328-330]. Strona osiągnęła wynik 510ms, co jest świetnym rezultatem, świadczącym o szybkim ładowaniu strony.
- **Total Blocking Time (0ms)** - Total Blocking Time (TBT) określa, ile czasu jest blokowane przez skrypty w trakcie ładowania strony. Dla dobrej jakości odczuć z użytkownika aplikacji, TBT powinien wynosić 150ms lub mniej [321, 331]. Strona osiągnęła wynik 0ms, co jest doskonałym wynikiem, wskazującym na brak opóźnień spowodowanych przez skrypty.
- **Largest Contentful Paint (435ms)** - Largest Contentful Paint (LCP) mierzy czas, jaki upływa, zanim największy element treści (np. główny obraz) jest renderowany na stronie. Dla dobrej jakości wrażeń ze strony użytkownika, LCP powinien wynosić 1,2s lub mniej [332-334]. Strona osiągnęła wynik 435ms, co jest znakomitym rezultatem, świadczącym o szybkim ładowaniu kluczowych elementów strony.
- **Cumulative Layout Shift (0.02)** - Cumulative Layout Shift (CLS) mierzy, jak bardzo układ strony przesuwa się w trakcie ładowania. Jest to ważny wskaźnik stabilności strony. CLS powinien wynosić 0,1 lub mniej [335-338]. Strona osiągnęła wynik 0.02, co jest doskonałym rezultatem, wskazującym na stabilność układu strony podczas ładowania.

Oprócz metryk wydajności, które skupiają się na tym, jak użytkownik doświadcza strony, istnieje również zestaw kluczowych wskaźników, które mierzą, jak przeglądarka internetowa przetwarza witrynę (także na Rys. 4.27). Te wskaźniki dają wgląd w różne etapy ładowania i renderowania strony, pozwalając na zrozumienie, gdzie mogą występować opóźnienia lub problemy. Poniżej przedstawiono szczegółowe opisy tych wskaźników, pogrupowane tematycznie:

- **Połączenie i przekierowanie:**

- **Redirect (0ms):** Czas, który upływa, gdy użytkownik jest przekierowywany z jednego URL-a na inny. W tym przypadku wynik 0ms oznacza, że nie ma przekierowań, co jest idealnym scenariuszem [339-340].
- **Connect (176ms):** Czas potrzebny na nawiązanie połączenia strony z serwerem. 176ms to akceptowalny wynik, ale może to być obszar do dalszej optymalizacji [341-343], zwłaszcza jeśli korzysta z zewnętrznych zasobów.
- **Odpowiedź serwera i przetwarzanie backend:**
 - **Backend (167ms):** Czas, który serwer potrzebuje na przygotowanie odpowiedzi na żądanie strony. 167ms to dobry wynik, świadczący o wydajnym przetwarzaniu na serwerze [348-349].
 - **TTFB (Time to First Byte) (343ms):** Czas, który upływa od żądania do otrzymania pierwszego bajtu odpowiedzi od serwera. 343ms to akceptowalny wynik, ale może to być obszar do dalszej optymalizacji [345-349].
- **Renderowanie i ładowanie strony**
 - **First Paint (435ms):** Czas, który upływa, zanim przeglądarka renderuje pierwszy piksel strony. 435ms to dobry wynik, świadczący o szybkim rozpoczęciu renderowania [350-352].
 - **DOM Interactive (DOM Int.) (439ms):** Czas, który upływa, zanim Document Object Model (DOM) strony jest w pełni interaktywny. 439ms jest dobrym czasem, który dowodzi o szybkim przetwarzaniu struktury strony [353-354].
 - **DOM Loaded (440ms):** Z wynikiem 440ms, DOM Loaded odnosi się do momentu, gdy struktura strony jest w pełni załadowana i gotowa do interakcji. Jest to pozytywny wynik, który wskazuje na sprawnie działającą stronę [355-356].
 - **Onload (737ms):** Czas ładowania strony, w tym wszystkich zasobów, takich jak obrazy i skrypty, wynosi 737ms. Jest to solidny wynik, który świadczy o tym, że strona jest dobrze zoptymalizowana pod względem szybkości ładowania [357-358].
 - **Fully Loaded (908ms):** Całkowity czas ładowania strony, w tym wszelkie dodatkowe zasoby ładowane po zdarzeniu Onload, wynosi 908ms. Ten bardzo dobry wynik odzwierciedla optymalizację całego procesu ładowania strony i świadczy o wysokiej jakości wydajności.

Te wskaźniki razem dają pełny obraz tego, jak strona jest przetwarzana przez przeglądarkę, od momentu żądania do pełnego załadowania (zwizualizowane jest to także

na diagramie wodospadu dostępnym na Rys. 4.29). Pozwalają one na zrozumienie, gdzie mogą występować wąskie gardła i gdzie można dokonać optymalizacji, aby zapewnić jeszcze szybsze i bardziej responsywne doświadczenie użytkownika.

W tym przypadku, strona wykazuje się ogólnie dobrymi wynikami, z kilkoma obszarami, które mogą wymagać dalszej analizy i optymalizacji.

Ostatnią istotną częścią raportu jest wykaz elementów bezpośrednio wpływających na długość ładowania (Rys. 4.28). Sekcja "Structure Audits" raportu analizuje strukturę strony internetowej, skupiając się na optymalizacji obrazów, tagach i elementach, czasie wykonania JavaScript, sieci i dostarczaniu zasobów oraz kluczowych wskaźnikach wydajności. Zawiera zarówno analizę mocnych stron, jak i potencjalnych obszarów do poprawy, takich jak brakujące tagi, możliwości oszczędności w rozmiarze zasobów, i rekomendacje dotyczące optymalizacji różnych aspektów struktury strony.

Jest to kluczowy element analizy wydajności ładowania witryny, który może prowadzić do znaczących usprawnień w doświadczeniu użytkownika i uwagi pochodzące z niej zostaną wdrożone w kolejnej wersji oprogramowania REDUCE – tak aby zapewnić jeszcze lepszą jakość obsługi narzędzia.

4.4. Walidacja poprawności budowy aplikacji w zgodzie z obowiązującymi standardami

Walidacja poprawności budowy aplikacji jest następnym ważnym etapem w procesie tworzenia oprogramowania, który zapewnia, że aplikacja jest zbudowana zgodnie z obowiązującymi standardami i wytycznymi. W przypadku projektu, który jest przedmiotem niniejszego rozdziału, zdecydowano się na trzy główne obszary walidacji: zgodność ze standardem W3C HTML, W3C CSS oraz audyt dostępności wg WCAG. Każde z tych zagadnień odgrywa istotną rolę w zapewnieniu jakości, wydajności i dostępności aplikacji.

- **Walidacja zgodności ze standardem W3C HTML:** HTML (HyperText Markup Language) jest podstawowym językiem wykorzystywanym do tworzenia stron internetowych. Standard W3C HTML zapewnia zbiór wytycznych i zasad, które gwarantują, że kod HTML jest dobrze sformułowany i zgodny z obowiązującymi praktykami [359-362]. Walidacja zgodności z tym standardem pomaga w identyfikacji i eliminacji błędów w kodzie, co prowadzi do lepszej kompatybilności z różnymi przeglądarkami i urządzeniami [363-367]. Ponadto, poprawnie napisany kod HTML jest łatwiejszy w utrzymaniu i rozbudowie [368-370].
- **Walidacja zgodności ze standardem W3C CSS:** CSS (Cascading Style Sheets) odpowiada za prezentację i stylizację stron internetowych. Standard W3C CSS

określa zasady i wytyczne, które zapewniają, że arkusze stylów są napisane w sposób spójny i zgodny z najlepszymi praktykami [370-374]. Walidacja CSS pozwala na wykrycie i naprawę błędów stylów, co prowadzi do lepszej wydajności i spójności wyglądu strony na różnych platformach [375-379].

- **Audyt dostępności wg. wytycznych WCAG 2.0.:** Dostępność stron internetowych jest kluczowym aspektem, który zapewnia, że treści są dostępne dla jak najszerszego grona użytkowników, w tym osób z różnymi niepełnosprawnościami [380-388]. Wytyczne dostępności treści internetowych (WCAG) są międzynarodowym standardem, który określa, jak uczynić treści internetowe dostępnymi dla wszystkich [389-395]. Audyt dostępności wg. WCAG obejmuje ocenę różnych aspektów strony, takich jak użyteczność, nawigacja, kontrast kolorów i dostępność dla technologii asystujących, takich jak czytniki ekranu [396-399].

W sumie, te trzy obszary walidacji łączą się, tworząc holistyczne podejście do zapewnienia, że aplikacja jest zbudowana zgodnie z obowiązującymi standardami i wytycznymi. Nie tylko zwiększa to jakość i wydajność aplikacji, ale także zapewnia, że jest ona dostępna i użyteczna dla szerokiego spektrum użytkowników, co jest kluczowe w dzisiejszym zglobalizowanym i zróżnicowanym świecie technologii.

4.4.1 Walidacja zgodności ze standardem W3C HTML

Walidacja zgodności ze standardem W3C HTML jest kluczowym etapem w procesie tworzenia i utrzymania stron internetowych. Jak wspomniano, zgodność z tym standardem zapewnia, że strona będzie poprawnie wyświetlana przez różne przeglądarki i urządzenia, a także będzie dostępna dla technologii wspomagających, takich jak czytniki ekranowe. W tym rozdziale skupiono się na procesie walidacji strony internetowej oraz zidentyfikowaniu potencjalnych błędów i niezgodności. Do walidacji wykorzystano narzędzie dostępne na stronie [400], które jest oficjalnym walidatorem World Wide Web Consortium (W3C). Do kluczowych cech tego narzędzia zalicza się fakt, iż:

- Jako produkt W3C, walidator ten jest uważany za autorytatywny i wiarygodny w dziedzinie walidacji HTML.
- Walidator obsługuje różne wersje HTML, w tym HTML5, XHTML i inne, umożliwiając walidację zgodnie z konkretnym standardem.
- Narzędzie to analizuje kod źródłowy strony, identyfikując błędy i ostrzeżenia, które mogą wpływać na jej działanie i dostępność.

- Jest to narzędzie online, dostępne dla każdego, kto chce przetestować swoją stronę pod kątem zgodności ze standardem. Prosty interfejs ułatwia korzystanie z walidatora zarówno dla profesjonalistów, jak i początkujących twórców stron.
- Poza identyfikacją błędów, walidator dostarcza także zaleceń i wskazówek, jak je naprawić, co jest niezwykle pomocne w procesie optymalizacji strony.



Rys. 4.31. Wyniki walidacji HTML, cz. 1

Ostrzeżenia i błędy, które wychwycił walidator znajdują się na Rys. 4.31, 4.32 oraz 4.33. Podczas procesu walidacji strony narzędzia REDUCE za pomocą walidatora W3C HTML, zidentyfikowano szereg błędów i ostrzeżeń, które wymagają uwagi. Oto podsumowanie znalezionych problemów:

- **Ostrzeżenia:**
 - W dwóch miejscach użyto atrybutu "border" w tagu obrazu. Zaleca się zastąpienie go odpowiednią regułą CSS.
- **Błędy:**
 - Atrybut "color" nie jest dozwolony w jednym z kontekstów.
 - Wiele błędów związanych z nieprawidłowym umieszczaniem tekstu i znaków w elemencie "ul". Element "ul" powinien zawierać tylko elementy "li" i elementy wspierające skrypty.
 - Kilka przypadków, w których prawdopodobnie brakuje zastąpienia "<" przez "<".
 - Wiele przypadków, w których "ul" został umieszczony wewnątrz "ol", co jest niezgodne ze standardem.

- Ostatni błąd wskazuje na nieprawidłowe umiejscowienie tagu "script".

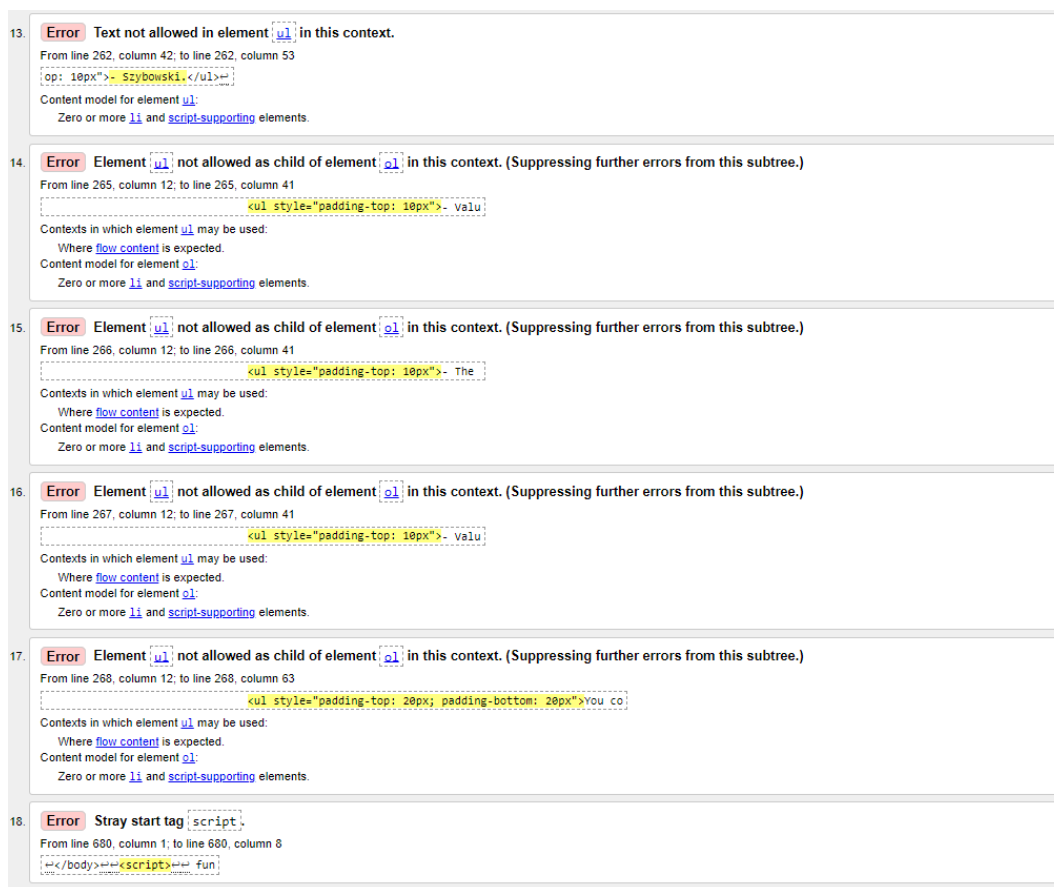
Wszystkie te błędy i ostrzeżenia są szczegółowo opisane, co umożliwi ich skuteczną naprawę. Jest to kluczowe dla zapewnienia zgodności strony z obowiązującymi standardami, co z kolei wpłynie na jej wydajność, dostępność i kompatybilność z różnymi przeglądarkami.

The image shows a screenshot of HTML validation results, specifically items 7 through 12. Each item is presented in a separate box with a red 'Error' label. The errors are as follows:

- Item 7:** Error: Text not allowed in element `ul` in this context. From line 257, column 46; to line 257, column 53. The code snippet is: `10px"> < 0.1 and < 0.05`. Content model for element `ul`: Zero or more `li` and `script-supporting` elements.
- Item 8:** Error: Bad character `<` after `<`. Probable cause: Unescaped `<`. Try escaping it as `<`. At line 257, column 55. The code snippet is: `"> < 0.1 and < 0.05 - CR belo`.
- Item 9:** Error: Text not allowed in element `ul` in this context. From line 257, column 54; to line 257, column 55. The code snippet is: `< 0.1 and < 0.05 -`. Content model for element `ul`: Zero or more `li` and `script-supporting` elements.
- Item 10:** Error: Text not allowed in element `ul` in this context. From line 257, column 56; to line 257, column 171. The code snippet is: `0.1 and < 0.05 - CR below these values is considered to be enough for achieving consistency for Pairwise Comparison Matrices,`. Content model for element `ul`: Zero or more `li` and `script-supporting` elements.
- Item 11:** Error: Text not allowed in element `ul` in this context. From line 258, column 42; to line 258, column 107. The code snippet is: `op: 10px"> 0 - which means that the result matrix will be fully consistent.`. Content model for element `ul`: Zero or more `li` and `script-supporting` elements.
- Item 12:** Error: Text not allowed in element `ul` in this context. From line 261, column 42; to line 261, column 101. The code snippet is: `op: 10px"> Xu and Wei - with two values of lambda parameter (0.9 and 0.5),`. Content model for element `ul`: Zero or more `li` and `script-supporting` elements.

Rys. 4.32. Wyniki walidacji HTML, cz. 2

Znalezione błędy i ostrzeżenia są typowymi problemami, które mogą wystąpić podczas tworzenia stron internetowych. W niektórych przypadkach są to drobne nieścisłości, ale w innych mogą prowadzić do poważniejszych problemów z wyświetlaniem strony. Dlatego też planowane jest ich naprawienie w kolejnej wersji narzędzia REDUCE. Proces walidacji i naprawy tych błędów jest niezbędny, aby zapewnić, że strona będzie działała zgodnie z oczekiwaniami użytkowników i zgodnie z międzynarodowymi standardami.



Rys. 4.33. Wyniki walidacji HTML, cz. 3

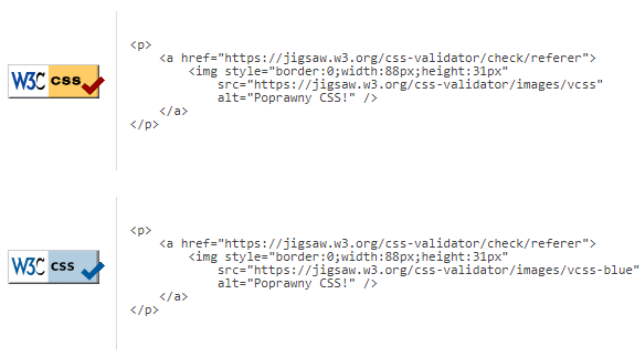
4.4.2. Walidacja zgodności ze standardem W3C CSS

Walidacja arkuszy stylów CSS jest istotna w procesie tworzenia i utrzymania stron internetowych. Pozwala na upewnienie się, że używane style są zgodne ze standardami i nie zawierają błędów, które mogą wpłynąć na wygląd i działanie strony w różnych przeglądarkach. Do przeprowadzenia walidacji CSS dla narzędzia REDUCE wykorzystano narzędzie dostępne pod adresem [401]. Jest to oficjalny walidator CSS stworzony przez World Wide Web Consortium (W3C), organizację odpowiedzialną za rozwijanie standardów internetowych. Narzędzie to analizuje arkusze stylów pod kątem zgodności z obowiązującymi standardami CSS, takimi jak CSS2, CSS2.1, CSS3 oraz innymi wersjami. Oferuje również różne opcje walidacji, takie jak sprawdzanie konkretnych profilów CSS, medium (np. ekran, druk) czy poziomu ostrzeżenia.

Wyniki Walidatora CSS W3C dla <https://reduce.prz.edu.pl> (CSS wersja 3 + SVG)**Gratulacje! Nie znaleziono żadnych błędów.**

Dokument ten jest poprawnie napisanym arkuszem [CSS wersja 3 + SVG](#) !

Aby pokazać czytelnikom swojej strony, że stworzyłeś stronę interoperacyjną, możesz umieścić ikonę na każdej stronie, która pomyślnie przeszła walidację. Oto kod HTML, który możesz dodać do swojej strony:



(zakończ znacznik obrazu z > zamiast /> jeśli używasz HTML <= 4.01)

Rys. 4.34 Wyniki walidacji CSS

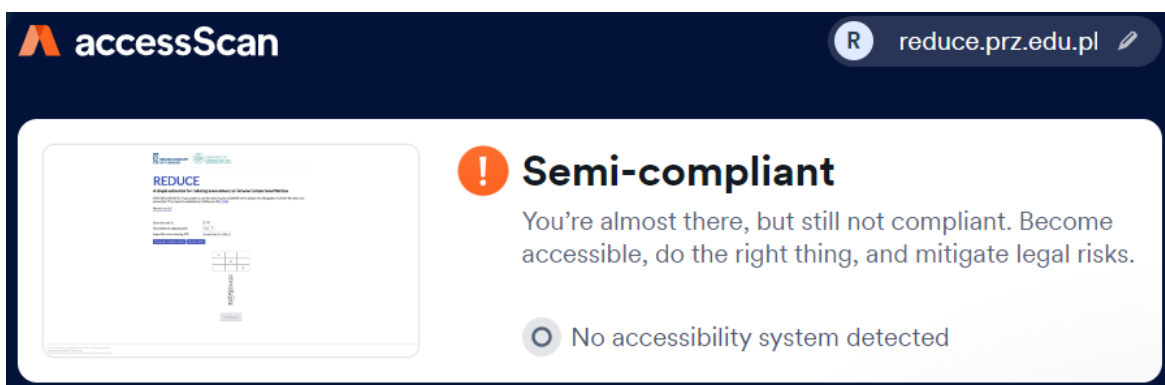
Proces walidacji arkuszy stylów CSS narzędzia REDUCE zakończył się pełnym sukcesem. Walidator nie wykrył żadnych błędów ani ostrzeżeń, co świadczy o wysokiej jakości kodu i jego zgodności ze standardami. Wynik ten jest widoczny na Rys. 4.34 i potwierdza, że style CSS zostały zaprojektowane i zaimplementowane zgodnie z najlepszymi praktykami oraz obowiązującymi wytycznymi. Taki wynik jest kluczowy dla zapewnienia spójnego i niezawodnego wyglądu strony na różnych urządzeniach i przeglądarkach.

4.4.3. Audyt dostępności wg. standardu WCAG 2.1

Audyt dostępności strony internetowej jest procesem, który ma na celu zapewnienie, że strona jest dostępna i użyteczna dla jak największej liczby osób, w tym osób z różnymi rodzajami niepełnosprawności. Standardy dostępności, takie jak WCAG (Web Content Accessibility Guidelines) 2.1, dostarczają wytyczne i kryteria, które pomagają twórcom stron internetowych w zapewnieniu, że ich treść jest dostępna dla wszystkich użytkowników. W ramach audytu dostępności narzędzia REDUCE, wykorzystano narzędzie dostępne na stronie [402]. Jest to zaawansowane narzędzie do skanowania dostępności, które automatycznie analizuje stronę internetową pod kątem zgodności z wytycznymi WCAG 2.1.

Narzędzie to oferuje szczegółowy raport, który zawiera informacje o potencjalnych problemach z dostępnością, a także sugestie, jak je naprawić.














Standard WCAG 2.1 składa się z serii wytycznych podzielonych na trzy poziomy zgodności: A (najniższy), AA (średni) i AAA (najwyższy). Każdy poziom ma swoje wymagania, które muszą być spełnione, aby strona była uważana za zgodną z danym poziomem. Audyt dostępności jest nie tylko kwestią etyczną, ale także prawną w wielu jurysdykcjach. Dostarcza on również korzyści biznesowych, ponieważ zwiększa zasięg i użyteczność strony, co może prowadzić do zwiększenia liczby użytkowników i klientów. W tej części przedstawimy szczegółowe wyniki audytu dostępności narzędzia REDUCE, w tym znalezione problemy i rekomendacje dotyczące ich rozwiązania.














Rys. 4.35 Wynik audytu WCAG 2.1

Audyt dostępności strony internetowej reduce.prz.edu.pl został przeprowadzony zgodnie z wytycznymi "Web Content Accessibility Guidelines" (WCAG) 2.1 na poziomie AA. Wyniki audytu skupiają się głównie na trzech kategoriach, które są kluczowe dla różnych grup użytkowników:













- **Dostosowania czytnika ekranowego (dla osób niewidomych):** Te testy oceniają, jak łatwo osoby niewidome mogą nawigować po stronie i zrozumieć jej treść za pomocą technologii asystujących, takich jak czytniki ekranowe.
- **Dostosowania nawigacji klawiaturą (dla osób z upośledzeniem ruchowym):** Te testy oceniają, jak łatwo osoby z ograniczoną zdolnością do korzystania z myszy lub innego urządzenia wskaźnikowego mogą nawigować po stronie za pomocą klawiatury.
- **Dostosowania interfejsu użytkownika, projektowania i czytelności (dla osób z upośledzeniem wzroku):** Te testy oceniają, jak łatwo osoby z różnymi rodzajami upośledzenia wzroku, takimi jak daltonizm czy niski kontrast wzroku, mogą korzystać ze strony.

Clickables		Score: 75
<input checked="" type="checkbox"/>	Elements with button functionality should be tagged for assistive technology	   
<input checked="" type="checkbox"/>	Buttons cannot be empty	  
<input checked="" type="checkbox"/>	Links cannot be empty	  
<input checked="" type="checkbox"/>	Links that open in new tabs/windows should be tagged for assistive technology	  

Rys. 4.36. Wyniki audytu w kategorii „Clickables”

Titles		Score: 100
<input checked="" type="checkbox"/>	Every page should include a single H1 title	  
<input checked="" type="checkbox"/>	Titles cannot be empty	 
<input type="checkbox"/>	Titles built as text tags should be labeled as headings for assistive technology	  
<input checked="" type="checkbox"/>	Titles should have a consistent hierarchy	  

Rys. 4.37. Wyniki audytu w kategorii „Titles”

Menus		Neutral
<input type="checkbox"/>	Menus should be tagged for assistive technology	   
<input type="checkbox"/>	Menu dropdowns should be tagged for assistive technology	   
<input type="checkbox"/>	Dropdown expanded/collapsed state should be represented in the code	   

Rys. 4.38. Wyniki audytu w kategorii „Menus”

Orientation		Score: 33
<input checked="" type="checkbox"/>	Interactive elements should be navigable using the keyboard	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input checked="" type="checkbox"/>	Keyboard focus should have a noticeable outline	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	Active popups should be tagged for assistive technology	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input checked="" type="checkbox"/>	Every page should include hidden links that allow skipping blocks	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	Manipulatively hidden interactive elements should be excluded from assistive technology	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	Manipulatively hidden interactive elements should be excluded from tabindex	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

Rys. 4.39. Wyniki audytu w kategorii „Orientation”

Wynik audytu określono jako "**półzgodny**", co oznacza, że strona spełnia niektóre, ale nie wszystkie wytyczne WCAG 2.1 na poziomie AA (Rys. 4.35). Jest to ważny wskaźnik tego, że choć strona ma pewne funkcje związane z dostępnością, istnieją obszary, które wymagają dalszej pracy i dostosowania.

Poniżej znajduje się interpretacja wyników testu w określonych kategoriach:

- **Elementy klikalne (Rys. 4.36):**

- Elementy, które zachowują się jak przyciski, ale są zbudowane przy użyciu innych znaczników, takich jak span, div, a lub innych, powinny zawierać atrybut "role" równy "button". Wymaganie to zostało spełnione z wynikiem 100%.
- Elementy interaktywne, takie jak linki, przyciski i pola formularzy, powinny być obsługiwane za pomocą klawiatury. To wymaganie nie zostało spełnione, a test zakończył się niepowodzeniem.
- Elementy interaktywne, które mogą być nawigowane za pomocą klawiatury, powinny być otoczone wizualnym konturem, gdy są skupione. Wymaganie to zostało spełnione z wynikiem 100%.

- **Tytuły (Rys. 4.37):**

- Tytuł H1 dostarcza niewidomym użytkownikom korzystającym z czytników ekranu informacji o głównym temacie strony. To wymaganie zostało spełnione z wynikiem 100%.
- Elementy tytułu HTML (H1-6) powinny zawierać tekst. To wymaganie zostało spełnione z wynikiem 100%.

- Elementy, które wizualnie wyglądają jak tytuły, ale są zakodowane za pomocą znacznika HTML innego niż nagłówek, powinny zawierać atrybut "role" równy "heading" lub mieć ustalone znaczniki. To wymaganie nie miało zastosowania w teście.
- Poziomy tytułów nie powinny być pomijane i powinny być zbudowane ze spójną hierarchią. Wymaganie to zostało spełnione z wynikiem 100%.
- **Menu (Rys. 4.38):**
 - Menu powinny być zbudowane przy użyciu elementu HTML5 "nav" lub zawierać atrybut "role" równy "menu" lub "navigation". To wymaganie nie zostało spełnione, a test zakończył się niepowodzeniem.
 - Elementy menu, które mają menu rozwijane, powinny zawierać określone atrybuty "aria". Wymagania te nie zostały spełnione, a test zakończył się niepowodzeniem.
- **Orientacja (Rys. 4.39):**
 - Punkty orientacyjne, takie jak treść główna i stopka, powinny być zbudowane przy użyciu odpowiedniego elementu HTML5 lub zawierać opis przy użyciu atrybutu "aria-label" i znacznika "role". To wymaganie zostało częściowo spełnione z wynikiem 67%.
- **Czytelność (Rys. 4.40):**
 - Rozmiar czcionki powinien wynosić co najmniej 11 pikseli, aby zachować czytelność. To wymaganie nie zostało spełnione, a test zakończył się niepowodzeniem.
 - Aby zachować czytelność, odstęp między literami nie powinny być mniejsze niż -1px. To wymaganie zostało spełnione z wynikiem 100%.
 - Elementy zawierające tekst powinny spełniać minimalny współczynnik kontrastu 4,5:1 między kolorem pierwszego planu a kolorem tła. To wymaganie zostało częściowo spełnione z wynikiem 33%.
- **Dokument (Rys. 4.41):**
 - Znacznik tytułu sekcji HEAD powinien istnieć i opisywać nazwę bieżącej strony internetowej dla użytkowników czytników ekranu. Wymaganie to zostało spełnione z wynikiem 100%.
 - Znacznik HTML powinien zawierać atrybut "lang". To wymaganie zostało spełnione z wynikiem 100%.
 - Znacznik meta viewport powinien pozwalać użytkownikom niedowidzącym na powiększenie obrazu co najmniej dwukrotnie w

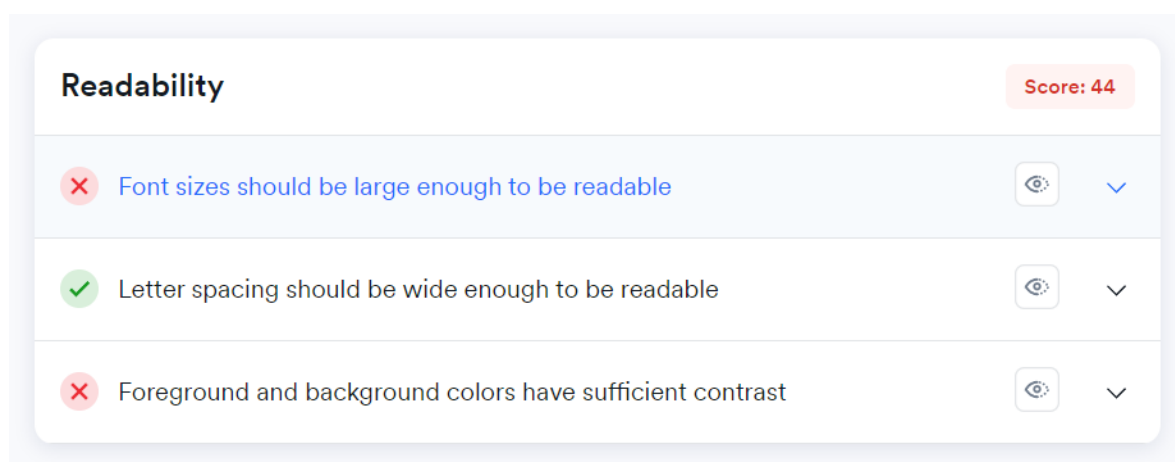
stosunku do standardowego skalowania. To wymaganie nie zostało spełnione, a test zakończył się niepowodzeniem.

- **Grafika (Rys. 4.42):**

- Obrazy powinny mieć alternatywny opis tekstowy. Wymaganie to zostało spełnione z wynikiem 100%.
- Różne wymagania związane z obrazami, obrazami tła i ikonami. Większość z tych wymagań nie została spełniona, a test zakończył się niepowodzeniem.

- **Formularze (Rys. 4.43):**

- Pola formularzy powinny zawierać atrybut "aria-label" lub połączony element LABEL opisujący wymagania pola. To wymaganie zostało częściowo spełnione z wynikiem 25%.
- Różne wymagania związane z polami formularzy, identyfikatorami, polami wymaganymi i przyciskami przesyłania. Większość z tych wymagań nie została spełniona, a test zakończył się niepowodzeniem.
- Formularze wyszukiwania powinny zawierać tag "role" równy "search". To wymaganie nie zostało spełnione, a test zakończył się niepowodzeniem.



Rys. 4.40 Wyniki audytu w kategorii „Readability”

Ogólnie rzecz biorąc, wyniki testów wskazują na mieszankę sukcesów i porażek w różnych kategoriach. Podczas gdy niektóre obszary, takie jak kliknięcia i tytuły, wykazują silną zgodność ze standardami dostępności, inne obszary, takie jak menu, czytelność, grafika i formularze, ujawniają znaczne pole do poprawy.

Document		Score: 89
<input checked="" type="checkbox"/>	The HEAD element should include a title element with the name of the page	
<input checked="" type="checkbox"/>	The HTML element should include a proper "lang" attribute	
<input type="radio"/>	Meta viewport allows display scaling of at least 200%	
<input type="checkbox"/>	Page landmarks should be tagged and described for assistive technology	

Rys. 4.41 Wyniki audytu w kategorii „Document”

Poprawki takie zostaną wdrożone w kolejnej wersji oprogramowania REDUCE, która planowana jest do wydania w kolejnym roku (2024), zaś potencjał komercyjny i dalsze plany rozwoju pakietu, zostaną opisane w kolejnym podrozdziale.

Graphics		Score: 50
<input checked="" type="checkbox"/>	Objects and embedded text of images should be described for assistive technology	
<input type="radio"/>	Background images behaving as standard images should be tagged and described	
<input checked="" type="checkbox"/>	Non-functional icons/spacers should be excluded from assistive technology	
<input type="radio"/>	Figure elements should have text or be excluded from assistive technology	
<input type="radio"/>	Image map areas should be described for assistive technology	
<input type="radio"/>	Tracker images (pixels) should be excluded from assistive technology	

Rys. 4.42 Wyniki audytu w kategorii „Graphics”

Forms		Score: 63
<input checked="" type="checkbox"/>	Form fields should be properly labeled	
<input type="checkbox"/>	Form fields should not have duplicate ID selectors	
<input type="checkbox"/>	Required form fields should be tagged for assistive technology	
<input type="checkbox"/>	Validation status of form fields should be represented in the code	
<input type="checkbox"/>	Incorrect use of aria-describedby/labeledby	
<input checked="" type="checkbox"/>	All forms have associated submission buttons	
<input type="checkbox"/>	Search forms should be tagged for assistive technology	

Rys. 4.43 Wyniki audytu w kategorii „Forms”

4.5. Ścieżki dalszego rozwoju oprogramowania i jego komercjalizacji

Ten rozdział należy rozpocząć od informacji, że w kolejnej wersji oprogramowania REDUCE, użytkownicy mogą spodziewać się następujących ulepszeń dot. funkcjonalności znanych z opisywanej wersji:

- **Naprawa błędów:** Wszystkie znane błędy zostaną zidentyfikowane i naprawione, co zapewni płynniejsze działanie.
- **Optymalizacja wydajności:** Zostaną wprowadzone ulepszenia w zakresie szybkości i responsywności systemu.
- **Ulepszone funkcje:** Istniejące funkcje zostaną udoskonalone, a także mogą zostać dodane nowe, aby lepiej spełnić potrzeby użytkowników.
- **Bezpieczeństwo:** Zostaną wdrożone najnowsze zabezpieczenia, aby zapewnić ochronę danych i prywatności użytkowników.
- **Wsparcie i dokumentacja:** Zostanie udostępniona aktualizowana dokumentacja oraz wsparcie techniczne, aby ułatwić korzystanie z oprogramowania.

Planowane poprawki są wynikiem ciągłego dialogu z użytkownikami i analizy ich potrzeb. Kolejna wersja oprogramowania REDUCE ma na celu nie tylko naprawę istniejących problemów, ale także wprowadzenie innowacji, które uczynią produkt jeszcze

bardziej atrakcyjnym i funkcjonalnym. Dalszy rozwój oprogramowania REDUCE nie jest jedynie kwestią wprowadzania nowych funkcji czy naprawy błędów. To holistyczne podejście, które obejmuje zrozumienie potrzeb użytkowników, analizę trendów rynkowych, ocenę konkurencji oraz wykorzystanie najnowszych technologii i metodologii. Jest to proces, który wymaga ciągłego dostosowywania się i uczenia, aby sprostać rosnącym wymaganiom i oczekiwaniom.

Ponieważ dzięki dotychczasowym badaniom planowane jest pozyskanie dodatkowych środków z programów grantowych na dalszy rozwój tego oprogramowania i jego rozwój, w tym podrozdziale omówiona zostanie ścieżka jego potencjalnej komercjalizacji.

4.5.1 Identyfikacja potrzeby rynkowej

Potrzeba rynkowa na narzędzie do podejmowania decyzji wynika z różnych wyzwań, przed którymi stają osoby, firmy i organizacje w dzisiejszym złożonym i szybko zmieniającym się środowisku. Do tych wyzwań należą:

- **Przeciążenie informacjami:** Przy rosnącej ilości dostępnych danych i informacji, osoby podejmujące decyzje często mają trudności z przetworzeniem i analizą wszystkich istotnych czynników [403-407]. Narzędzie decyzyjne może pomóc w syntezie i filtrowaniu krytycznych danych, umożliwiając użytkownikom dokonywanie bardziej świadomych wyborów [408].
- **Ograniczenia czasowe:** W dzisiejszym szybkim świecie osoby i organizacje muszą podejmować decyzje szybko i sprawnie [409-413]. Narzędzia decyzyjne mogą usprawnić ten proces, automatyzując powtarzające się zadania i dostarczając ustrukturyzowanych ram, aby zaoszczędzić czas i zasoby [414].
- **Zwiększona złożoność:** Współczesne problemy często dotyczą wielu interesariuszy, zmiennych i sprzecznych celów [415-417]. Narzędzia do podejmowania decyzji mogą pomóc użytkownikom w poruszaniu się po tych złożonościach, dzieląc problemy na łatwe do opanowania części i dostarczając narzędzi do porównywania i oceny opcji [418].
- **Potrzeba współpracy:** Podejmowanie decyzji często wymaga współpracy między członkami zespołu o różnej wiedzy i perspektywach [419-424]. Narzędzie decyzyjne może ułatwić komunikację, umożliwić dzielenie się danymi i spostrzeżeniami oraz wspierać wspólne podejmowanie decyzji [425-426].
- **Obiektywność:** Narzędzia decyzyjne mogą pomóc użytkownikom w bardziej obiektywnym podejściu do problemów, dostarczając ustrukturyzowanych ram

i metodologii. Może to pomóc w zmniejszeniu wpływu uprzedzeń poznawczych i czynników emocjonalnych, które mogą zaciemniać ocenę [427-444].

- **Zarządzanie ryzykiem:** Niewłaściwe podejmowanie decyzji może prowadzić do negatywnych rezultatów, strat finansowych i utraty reputacji [445-451]. Narzędzia decyzyjne mogą pomóc użytkownikom w ocenie potencjalnego ryzyka i niepewności, umożliwiając im dokonywanie bardziej świadomych wyborów i łagodzenie potencjalnych konsekwencji [452-454].

Rozbudowane o odpowiednie funkcje istniejące narzędzie REDUCE może konkurować skutecznie z dostępnymi na rynku narzędziami takimi jak:

- **Expert Choice** – kosztującego od 3400 zł miesięcznie (przy uwzględnieniu jednorazowej rocznej opłaty) [455-459]
- **QVISTORP** – kosztującego od 4000 zł miesięcznie (przy uwzględnieniu jednorazowej rocznej opłaty) [460]
- **Paramount Decisions** – kosztującego od 150 zł miesięcznie (przy uwzględnieniu jednorazowej rocznej opłaty), nie oferującego pakietu analiz opartego na AHP [461-462]

Głównym atutem aplikacji REDUCE (lub nowej nazwy aplikacji) będzie fakt, iż większość podstawowych funkcjonalności będzie dostępna dla użytkownika za darmo, zaś dopiero wybrane funkcjonalności dostępne będą za dodatkową, niewielką opłatą (tzw. model freemium [463-467]), a przychody od użytkowników niepłacących będą uzyskiwane za pomocą reklam umieszczonych w narzędziu internetowym (takich jak Google Ads i inne sieci reklamowe [468-470]), co pozwoli na nieustanne rozwijanie narzędzia.

W porównaniu do bardzo drogiego oprogramowania przedstawionych wcześniej, uważamy to za rewolucyjną zmianę na rynku, ponieważ małe i średnie przedsiębiorstwa zazwyczaj nie mogą sobie pozwolić na drogie oprogramowanie komercyjne [471] i REDUCE może rozwiązać ten problem w małych i średnich przedsiębiorstwach, torując w ten sposób drogę do lepszego i szybszego podejmowania decyzji.

4.5.2 Opis rynku docelowego

Rynkiem docelowym dla pakietu decyzyjnego opartego na AHP byłyby przedsiębiorstwa i organizacje wszystkich rozmiarów i branż, które wymagają ustrukturyzowanego podejścia do podejmowania decyzji. Mogą to być między innymi:

- **Małe i średnie przedsiębiorstwa (MŚP)**, które nie mają środków na zatrudnienie dedykowanych analityków lub konsultantów do pomocy w podejmowaniu decyzji [472-476].
- **Duże organizacje** o skomplikowanych procesach decyzyjnych, które wymagają standardowego podejścia [477-480].
- **Agencje rządowe i organizacje non-profit**, które wymagają przejrzystości i odpowiedzialności w swoich procesach decyzyjnych [481-495].
- **Instytucje akademickie i organizacje badawcze**, które wymagają rygorystycznego i systematycznego podejścia do podejmowania decyzji [496-503].

Segmenty klientów mogą obejmować kadrę kierowniczą, menedżerów, analityków i konsultantów, którzy są odpowiedzialni za podejmowanie strategicznych lub operacyjnych decyzji w swojej organizacji. Podmioty oferujące alternatywne sposoby zaspokojenia zdiagnozowanej potrzeby mogą obejmować:

- **Tradycyjne firmy konsultingowe**, które oferują usługi decyzyjne dostosowane do potrzeb klienta [504-507],
- **Dostawcy oprogramowania**, którzy oferują oprogramowanie do podejmowania decyzji ogólnego przeznaczenia, ale mogą nie oferować ustrukturyzowanego podejścia AHP [461],
- **Wewnętrzni analitycy lub konsultanci**, którzy stosują ręczne lub doraźne metody podejmowania decyzji [504-507].

Przy wprowadzaniu nowego produktu lub usługi do przedsiębiorstwa zawsze istnieją obszary ryzyka i bariery, które należy rozważyć. Do potencjalnych obszarów ryzyka i barier, które mogą się pojawić przy wprowadzaniu pakietu decyzyjnego do działalności przedsiębiorstwa należy zaliczyć:

- **Brak akceptacji przez użytkowników:** Jeżeli pracownicy nie chcą lub nie mogą przyjąć nowego pakietu decyzyjnego, nie będzie on skuteczny. Może się tak zdarzyć, jeżeli pracownicy są oporni na zmiany lub jeżeli pakiet jest zbyt trudny w użyciu. Aby zminimalizować to ryzyko, ważne jest, aby zaangażować pracowników w proces wdrażania oraz zapewnić im odpowiednie szkolenia i wsparcie [508-512].
- **Bezpieczeństwo i prywatność danych:** Pakiety decyzyjne często wykorzystują dane wrażliwe, dlatego ważne jest, aby pakiet posiadał odpowiednie środki bezpieczeństwa w celu ochrony przed naruszeniem danych lub nieautoryzowanym dostępem. Może to być wdrożenie protokołów szyfrowania, korzystanie z bezpiecznego przechowywania w chmurze lub ograniczenie dostępu do wrażliwych danych [513-525].

- **Kwestie techniczne:** Problemy techniczne, takie jak błędy w oprogramowaniu, awarie systemu lub problemy z kompatybilnością mogą zakłócić działalność biznesową i zmniejszyć efektywność pakietu decyzyjnego. Aby zminimalizować to ryzyko, ważne jest dokładne przetestowanie pakietu przed wdrożeniem oraz posiadanie solidnego systemu wsparcia IT [526-529].
- **Koszt:** Wdrożenie pakietu decyzyjnego może być kosztowne, szczególnie jeśli wymaga znacznego dostosowania lub integracji z istniejącymi systemami. Aby zminimalizować to ryzyko, należy dokładnie ocenić koszty i korzyści związane z pakietem, a w razie potrzeby rozważyć rozwiązania alternatywne [530].
- **Opór kulturowy:** W niektórych przypadkach wprowadzenie pakietu decyzyjnego może być postrzegane jako zagrożenie dla istniejącej kultury organizacyjnej lub dynamiki władzy [531]. Może to wywołać opór wobec pakietu i ograniczyć jego skuteczność [532]. Aby zapobiec temu ryzyku, ważne jest, aby zaangażować kluczowych interesariuszy w proces wdrażania oraz informować o korzyściach płynących z pakietu w sposób dostosowany do ich potrzeb i obaw [533].

4.5.3 Opis przygotowanej ścieżki komercjalizacji

Pakiet decyzyjny oparty na Analytic Hierarchy Process (AHP) to specjalistyczne oprogramowanie, które pomaga użytkownikom podejmować decyzje w oparciu o ustrukturyzowane podejście, polegające na rozbiciu złożonych problemów na mniejsze, łatwiejsze do opanowania elementy. Oto funkcjonalności, które powinien posiadać pakiet gotowy do komercjalizacji do podejmowania decyzji oparty na AHP – a więc jednej z metod wykorzystujących macierze porównań parami:

- **Tworzenie i zarządzanie hierarchią:** AHP polega na rozbiciu złożonych decyzji na hierarchię kryteriów i podkryteriów. Pakiet powinien umożliwiać użytkownikom tworzenie i zarządzanie tą hierarchią, w tym możliwość dodawania, edytowania i usuwania kryteriów i podkryteriów.
- **Porównywanie parami:** AHP opiera się na porównywaniu parami kryteriów i podkryteriów w celu określenia ich względnej ważności. Pakiet powinien zapewniać użytkownikom mechanizm porównywania kryteriów i podkryteriów w parach oraz przypisywania im wag lub priorytetów w zależności od ich ważności.
- **Sprawdzanie spójności:** AHP opiera się na założeniu, że oceny użytkownika są zgodne z jego preferencjami. Pakiet powinien zawierać mechanizm sprawdzania spójności ocen dokonywanych przez użytkownika i ostrzegania użytkownika

w przypadku wykrycia niespójności. Tutaj pomocne okażą się dotychczasowe prace, które pozwalają uwzględnić algorytmy redukcji niespójności macierzy i zaaplikować je do gotowej aplikacji.

- **Analiza wrażliwości:** AHP umożliwi użytkownikom przeprowadzenie analizy wrażliwości, aby ocenić, jak zmiany w wagach kryteriów lub podkryteriów wpływają na ogólną decyzję. Pakiet powinien zapewniać narzędzia umożliwiające użytkownikom przeprowadzenie analizy wrażliwości i ocenę solidności ich decyzji.
- **Raportowanie i wizualizacja:** Pakiet powinien dostarczać narzędzi umożliwiający użytkownikom wizualizację wyników analizy, takich jak wykresy słupkowe lub kołowe, oraz generowanie raportów podsumowujących wyniki w jasny i zwięzły sposób. Pomoże to użytkownikom w skutecznym informowaniu o swoich decyzjach interesariuszy i innych zainteresowanych stron.

Po przygotowaniu narzędzia należy również przygotować wersję mobilną tego rozwiązania, tak aby decyzje podejmowane przez przedsiębiorstwo mogłyby być podejmowane, np. podczas wyjazdu służbowego, zarówno w formie dedykowanej responsywnej witryny internetowej, jak i dedykowanej aplikacji mobilnej na system Android (najpopularniejszy mobilny system operacyjny) [535]. Uwzględnić należy również zaprojektowanie i wdrożenie odpowiednich dodatkowych systemów bezpieczeństwa danych, szczególnie że jest to narzędzie dedykowane przedsiębiorcom, którzy mogą chcieć podejmować decyzje w kluczowych dla nich sprawach.

Następnie pozostaje wdrożenie bibliotek programistycznych sieci reklamowych, aby program był gotowy do zarabiania w modelu freemium, a także opracowanie szczegółowych płatnych pakietów oprogramowania. Aplikacja internetowa REDUCE będzie działać w modelu Freemium, oferując różne pakiety dostosowane do potrzeb różnych użytkowników. Przewidywany model płatności będzie się składał z wersji bezpłatnej wspieranej przez reklamy oraz dwóch dodatkowych pakietów płatnych, zapewniających szereg funkcjonalności i usług:

- **Wersja bezpłatna (z reklamami):** Ten pakiet jest przeznaczony dla użytkowników, którzy chcą mieć dostęp do podstawowych funkcji aplikacji bez ponoszenia jakichkolwiek kosztów. Wersja bezpłatna zawiera:
 - Podstawowe narzędzia i szablony do podejmowania decyzji
 - Dostęp do ograniczonej liczby studiów przypadków
 - Integrację z platformami mediów społecznościowych w celu udostępniania

- Reklamy wyświetlane w całej aplikacji, zapewniające dochód w celu wsparcia bezpłatnego dostępu
- **Pakiet Pro (19zł rocznie):** Ten pakiet pośredni jest przeznaczony dla użytkowników, którzy poszukują rozszerzonego doświadczenia z dodatkowymi funkcjami i bez reklam. Pakiet Pro zawiera wszystkie funkcje wersji bezpłatnej plus:
 - Brak reklam
 - Dostęp do rozszerzonej biblioteki studiów przypadków i narzędzi decyzyjnych
 - Priorytetowe wsparcie klienta
 - Podstawowe możliwości analizy danych i raportowania
- **Pakiet Premium (49zł rocznie):** Ten najwyższy pakiet jest przeznaczony dla użytkowników, którzy wymagają zaawansowanych funkcjonalności i bardziej kompleksowego pakietu decyzyjnego. Pakiet Premium zawiera wszystkie funkcje Pakietu Pro, a także:
 - Zaawansowane możliwości analizy danych i raportowania
 - Możliwość dostosowania szablonów decyzyjnych
 - Priorytetowy dostęp do nowych funkcji i aktualizacji
 - Narzędzia współpracy do podejmowania decyzji w zespole

4.6. Wnioski końcowe

Rozdział stanowi kompleksową analizę oprogramowania REDUCE, narzędzia wspomagającego decyzje oraz redukcji niespójności w multiplikatywnych porównaniach parami. Oprogramowanie REDUCE zostało zaprojektowane z myślą o elastyczności i wydajności, co pozwala na skuteczne zarządzanie procesami decyzyjnymi.

Analiza warstwy front-end i back-end ukazuje złożoność i zaawansowanie technologiczne narzędzia. Wybór technologii został uzasadniony, a ich zastosowanie jest spójne z celami projektu. Przeprowadzone testy bezpieczeństwa, zgodności oraz wydajnościowe potwierdzają, że aplikacja spełnia wymagane standardy i jest odporna na różne rodzaje ataków. Aplikacja przeszła pomyślnie proces walidacji zgodności ze standardami W3C HTML, W3C CSS oraz standardem dostępności WCAG 2.1 Istnieje wyraźna ścieżka dalszego rozwoju oprogramowania oraz potencjał do komercjalizacji. Analiza potrzeb rynkowych i rynku docelowego wskazuje na możliwości ekspansji i adaptacji narzędzia do różnych zastosowań.

Podsumowując, oprogramowanie REDUCE stanowi ważny krok w dziedzinie wspomagania decyzji i redukcji niespójności. Jego złożona struktura, zaawansowane funkcje i potencjał do dalszego rozwoju i komercjalizacji czynią go narzędziem, które może znaleźć zastosowanie w różnych kontekstach i przyczynić się do rozwoju nauki i praktyki w tej dziedzinie.

5. Zbadanie potencjału użycia wybranych technik uczenia maszynowego do redukcji niespójności macierzy porównań parami

Słowa kluczowe rozdziału:

uczenie maszynowe, porównywanie parami, redukcja niespójności, hierarchiczny proces analityczny, ahp

Abstrakt rozdziału (za oryginalną publikacją):

W pracy przedstawiono nowe podejście do problemu, tzw. redukcji niespójności w macierzach porównań. Macierze porównań parami są wykorzystywane przede wszystkim w wielokryterialnych metodach podejmowania decyzji, takich jak AHP (Analytic Hierarchy Process), gdzie problem zapewnienia spójności oceny wielu różnych parametrów pojawia się również w tych technikach. W niniejszej pracy do redukcji niespójności wykorzystano wybrane algorytmy z rodziny uczenia maszynowego. Jako dane uczące i testowe dla algorytmów wykorzystano wyniki iteracyjnego algorytmu redukcji niespójności (Szybowski), którego skuteczność została już dostatecznie opisana i zbadana. Znaczenie algorytmów dla nowych danych oceniano również pod kątem następujących kryteriów: zachowania oryginalnego wektora priorytetów i tzw. rankingu obiektów, oryginalnych preferencji w postaci macierzy porównań parami oraz zachowania największej liczby niezmiennych pozycji oryginalnej macierzy porównań. Omówiono również zasadność stosowania takich technik dla przedstawionego problemu, a także porównano wyniki.

5.1. Wstęp

Jak było to już podkreślane w poprzednich rozdziałach rozprawy, w wielokryterialnych metodach podejmowania decyzji, takich jak AHP, macierze porównań parami są używane do oceny i nadawania rang dla różnych parametrów. Pojawia się jednak problem zapewnienia spójności w ocenach wielu kryteriów, co może prowadzić do niedokładnego podejmowania decyzji. Aby rozwiązać ten problem, w niniejszym rozdziale przedstawiono nowe podejście do redukcji niespójności w macierzach porównań, które wykorzystuje wybrane algorytmy z rodziny uczenia maszynowego do redukcji niespójności w macierzy porównań parami. Skuteczność iteracyjnych algorytmów redukcji niespójności (w tym algorytmu Szybowskiego zastosowanego w tym podejściu) została już zbadana, a wyniki jego działania zostały wykorzystane jako dane uczące i testowe dla algorytmów w badaniu omawianym w aktualnym rozdziale. Proponowane podejście zostało ocenione na podstawie zachowania oryginalnego wektora priorytetów, rankingu obiektów, macierzy porównań parami i niezmiennych pozycji oryginalnej macierzy.

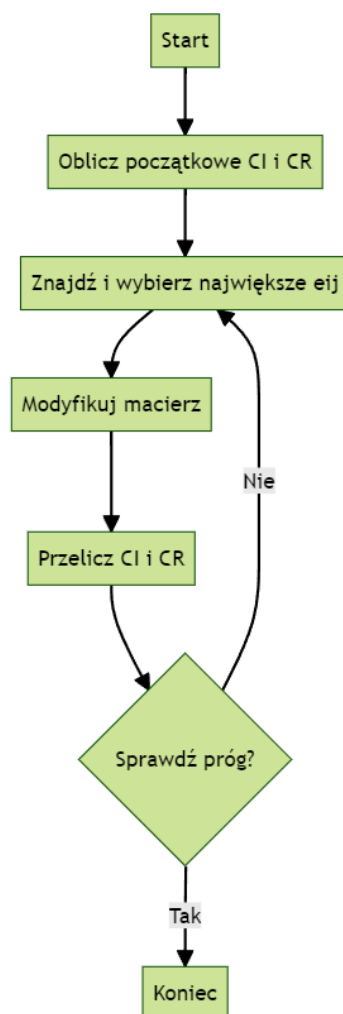
W chwili obecnej (2023 r.) nie zostały odnalezione w literaturze żadne inne próby rozwiązania tego problemu w macierzach porównań parami przy użyciu wskazanych technik. Niniejszy rozdział omawia także zasadność stosowania takich technik dla przedstawionego problemu i porównuje wyniki z istniejącymi metodami redukcji niespójności.

5.1.1. Uzasadnienie wyboru algorytmu będącego podstawą badań

Przypomnijmy, że spójność macierzy porównań parami jest kluczowym aspektem w procesie podejmowania decyzji. Saaty w 1977 roku wprowadził wskaźnik spójności (CI) oraz współczynnik spójności (CR) w celu oceny spójności macierzy [1]. Szczegółowy opis tego podejścia oraz matematyczne podstawy wskaźnika CI i współczynnika CR można znaleźć w Rozdziale 1.4. niniejszej rozprawy. Błędy w ludzkiej percepcji mogą prowadzić do niespójności w macierzach PC. Dokładne określenie lokalizacji tych błędów oraz ich wpływ na proces podejmowania decyzji zostało omówione w Rozdziale 1.4. oraz w publikacji Mazurka z 2023 roku [2].

Różne metody i algorytmy służące do identyfikacji i korygowania błędnych danych w macierzy A zostały opisane w literaturze. W Rozdziale 1.5. niniejszej rozprawy doktorskiej przedstawiono różnice między głównymi kategoriami algorytmów redukcji niespójności. Mazurek i in. [4] przeprowadzili badania nad algorytmami iteracyjnymi, w tym nad tymi zaproponowanymi przez Cao i in. [5], Szybowskiego [6] oraz Xu & Wei [7]. Warto zaznaczyć, że nie wszystkie algorytmy były skuteczne w praktyce, co zostało

również podkreślone w drugim rozdziale rozprawy doktorskiej, który oparty został na badaniach zespołu Mazurka i in. [4].



Rys. 5.1. Schemat działania algorytmu Szybowskiego (2018)

Według tych badań [4] algorytm Szybowskiego [6] utrzymywał najwięcej wpisów macierzy niezmienionych podczas redukcji niespójności. Z tego powodu został on również wybrany jako algorytm dostarczający danych uczących i testowych, co zostanie opisane w dalszej części rozdziału. Na poparcie tego wyboru wpłynęły następujące przesłanki, poparte analizą literatury dziedzinowej:

- Algorytm, który modyfikuje mniejszą liczbę elementów macierzy, zachowuje więcej oryginalnych danych. To może prowadzić do bardziej wiarygodnych wyników, ponieważ mniej informacji jest zniekształconych lub utraconych w procesie redukcji niespójności. Algorytm Szybowskiego, utrzymując większość wpisów macierzy niezmienionych, pozwala na zachowanie oryginalnej struktury danych, co może prowadzić do bardziej precyzyjnego modelowania [9-11].

- Modyfikacja mniejszej liczby elementów macierzy oznacza mniejsze ryzyko wprowadzenia szumu, który może zakłócić proces uczenia. Szum w danych może prowadzić do przeuczenia modelu, gdzie model staje się zbyt skomplikowany i traci zdolność do generalizacji na nowych danych [12-16].
- Model, który opiera się na danych mniej przetworzonych, jest zwykle łatwiejszy do zrozumienia i interpretacji. To może być kluczowe dla zastosowań, w których transparentność i zrozumienie procesu decyzyjnego modelu są ważne [17-18].

Jak już wcześniej wspomniano, algorytm Szybowskiemu opiera się na spostrzeżeniu, że jeśli macierz jest niespójna, to musi istnieć naruszenie własności przechodniości między co najmniej jedną parą elementów i ma on na celu systematyczne identyfikowanie i korygowanie tych naruszeń. Algorytm, zilustrowany na Rys. 5.1. został opisany w Rozdziale 1.5.2. niniejszej rozprawy doktorskiej.

5.1.2. Miary zachowania preferencji jako ostateczna metoda ewaluacji skuteczności nowego modelu

Celem każdego z algorytmów redukcji niespójności jest zminimalizowanie różnicy między oryginalną i zmodyfikowaną macierzą przy jednoczesnym zmniejszeniu niespójności. Algorytm redukcji niespójności jest definiowany jako algorytm, który przekształca oryginalną macierz porównań parami w zmodyfikowaną macierz z niższą miarą niespójności. W literaturze zaproponowano różne miary zachowania preferencji, w tym δ i σ przez Xu i Wei [7], całkowitą liczbę punktów odchylenia przez Pereirę i Costę [19] oraz miary kompromisu przez Abel i in. [19]. W niniejszym rozdziale zostaną wykorzystane następujące miary zachowania preferencji, jak wspomniano w Mazurek i in. (a także w drugim rozdziale niniejszej pracy doktorskiej) [4]:

- średnia zmiana wektora priorytetów (5.3),
- odległość tau Kendalla (5.4),
- odległość między dwiema macierzami (5.5), oraz
- liczba zmienionych elementów w macierzy po redukcji niespójności (5.6).

$$d(w, w') = \frac{1}{n} \sum_{i=1}^n |w_i - w'_i| \quad (5.3)$$

$$K_d(A_{ij}, A'_{ij}) = \sum_{\{i,j\} \in P, i < j} \bar{K}_{i,j}(\tau_1, \tau_2) \quad (5.4)$$

$$D(A, A') = \|A - A'\| = \sum_{i=1}^n \sum_{j=1}^n |a_{ij} - a'_{ij}| \quad (5.5)$$

$$N = |a_{ij}; a_{ij} \neq a'_{ij}, \forall i, j| \quad (5.6)$$

Niższe wartości tych miar wskazują na lepsze zachowanie oryginalnych preferencji.

5.1.3. Zasadność użycia technik machine learning do zagadnienia redukcji niespójności w macierzach porównania parami

Machine learning (ML), omówiony szerzej w kolejnym podrozdziale, stał się jednym z najbardziej popularnych terminów w dziedzinie technologii i nauki, biorąc pod uwagę statystyki z 2022 i 2023 roku [20-23]. Jego potencjał i możliwości są często przedstawiane jako rewolucyjne i wszechstronne [24, 25]. Jednakże, czy zastosowanie technik ML do każdej dziedziny jest zasadne? Czy nie jest to po prostu efekt nadmiernego rozdmuchania znaczenia i możliwości ML, znanego jako "overhype"? [26, 27].

Przecenianie możliwości ML, stało się powszechnym zjawiskiem w świecie technologii i nauki [26, 28]. ML jest często przedstawiany jako narzędzie, które może rozwiązać niemal każdy problem [28]. Od medycyny po finanse, ML jest promowany jako rozwiązanie na miarę naszych czasów [29, 30]. Media i działania marketingowe często przedstawiają ML jako technologię przyszłości, bez uwzględnienia jego ograniczeń i wyzwań [31-40]. Sensacyjne nagłówki i obietnice rewolucyjnych zmian mogą prowadzić do niewłaściwego zrozumienia możliwości ML.

Dla wielu osób, które nie są ekspertami w dziedzinie, ML może wydawać się magicznym rozwiązaniem [41, 42]. Brak zrozumienia technicznego może prowadzić do przeceniania tego, co ML może rzeczywiście osiągnąć [43-45]. Rzeczywiste sukcesy i przełomy w dziedzinie ML, takie jak zwycięstwa w grach planszowych [46] czy samochody autonomiczne [47], mogą prowadzić do przekonania, że ML jest odpowiedzią na wszystkie pytania. Firmy i naukowcy mogą odczuwać presję, aby zastosować ML w swoich projektach, nawet jeśli nie jest to najbardziej odpowiednie rozwiązanie, ze względu na oczekiwania rynku i inwestorów [48-50]. Oczywiście każda technologia często przechodzi przez cykl „hype'u”, gdzie początkowy entuzjazm prowadzi do nadmiernego optymizmu, a następnie do deziluzji, gdy technologia nie spełnia oczekiwań [51, 52]. ML, jak wskazują badacze, może być obecnie na szczycie tego cyklu [26, 27].

Kluczem jest zrozumienie, gdzie ML może być naprawdę skuteczny, a gdzie jest to jedynie efekt modnego trendu. W kontekście redukcji niespójności w macierzach porównania parami, badanie wykorzystania ML wydaje się być uzasadnione, choć zakładane będzie także nie podejmowanie dalszych działań, jeżeli rezultaty będą niezadawalające. Jednak już na początku prac panuje przypuszczenie, że np. drzewa decyzyjne mogą dobrze odwzorować strukturę typowego algorytmu wykorzystywanego do redukcji niespójności. Mogą one nauczyć się złożonych wzorców i zależności w danych. Ważne jest, aby podejść do ML z umiarkowanym entuzjazmem, rozumiejąc jego mocne i słabe strony, oraz zastosować go tam, gdzie naprawdę może przynieść wartość. W tym przypadku, analiza

i zrozumienie struktury problemu oraz odpowiednie zastosowanie technik ML mogą prowadzić do znaczących postępów w dziedzinie analizy wielokryterialnej i podejmowania decyzji.

5.2. Uczenie maszynowe w kontekście redukcji niespójności macierzy porównań parami

Uczenie maszynowe, do którego już nawiązano w poprzednim podrozdziale, będąc aplikacją sztucznej inteligencji, wykorzystuje algorytmy do analizy danych i uczenia się na ich podstawie, podejmując świadome decyzje [53]. W 1943 roku Walter Pitts i Warren McCulloch przedstawili pierwszy matematyczny model sieci neuronowych [54]. W 1949 roku Donald Hebb opublikował książkę "The Organization of Behavior", która stała się podstawowym dziełem w tej dziedzinie [55]. W 1950 roku Alan Turing stworzył Test Turinga, mający na celu określenie, czy komputer posiada prawdziwą inteligencję [56]. W 1952 roku Arthur Samuel napisał pierwszy program komputerowy uczący się gry w warcaby [57], a w 1957 roku Frank Rosenblatt zaprojektował perceptron, symulując ludzkie procesy myślowe [58]. W 1967 roku wprowadzono algorytm "najbliższego sąsiada" do podstawowego rozpoznawania wzorców [59]. W 1979 roku studenci ze Stanforda wynaleźli 'Stanford Cart' do nawigacji z przeszkodami [60]. W 1981 roku Gerald Dejong wprowadził Explanation Based Learning (EBL) [61].

Lata 90. przyniosły zmianę podejścia od opartego na wiedzy do opartego na danych [62]. W 1997 roku IBM's Deep Blue pokonał mistrza świata w szachach [63]. W 2006 roku Geoffrey Hinton wprowadził termin "deep learning" [64]. W 2010 roku Microsoft wprowadził technologię Kinect powiązaną z Xboxem 360 do śledzenia cech ludzkich [65]. W 2011 roku IBM's Watson wygrał w Jeopardy [66], a Google stworzyło Google Brain [67]. W 2014 roku Facebook wprowadził DeepFace do rozpoznawania osób na zdjęciach [68]. W 2015 roku Amazon wprowadził swoją platformę uczenia maszynowego [69], a Microsoft stworzył Distributed Machine Learning Toolkit [70]. W 2016 roku AlphaGo firmy Google wygrało w grze Go [71], a Waymo testowało autonomiczne samochody [72].

W 2020 roku OpenAI zaprezentowało światu GPT-3, przełomowy algorytm przetwarzania języka naturalnego [73]. Przyszłość uczenia maszynowego zapowiada się obiecująco, z naciskiem na usprawnienia w algorytmach uczenia nienadzorowanego, rozwój komputacji kwantowej i skupienie się na usługach poznawczych, takich jak rozpoznawanie wzrokowe i zrozumienie mowy [74].

5.2.1. Podstawowe rodzaje uczenia maszynowego

Jak wspomnieli Eberbach i Strzałka [75], algorytmy uczenia maszynowego umożliwiają systemom uczenie się i podejmowanie niezbędnych decyzji i są wykorzystywane w szeregu zadań obliczeniowych, w których projektowanie i programowanie algorytmów nie jest możliwe. Istnieją trzy główne kategorie metod i zadań uczenia maszynowego: uczenie nadzorowane, uczenie bez nadzoru i uczenie ze wzmocnieniem. Uczenie nadzorowane polega na mapowaniu obiektów wejściowych na pożądane wartości wyjściowe, podczas gdy uczenie nienadzorowane polega na zdefiniowaniu funkcji opisującej ukrytą strukturę na podstawie nieoznakowanych danych. Uczenie ze wzmocnieniem dotyczy tego, w jaki sposób agenci oprogramowania powinni podejmować działania w środowisku, aby zmaksymalizować skumulowaną nagrodę.

5.2.1.1. *Uczenie nadzorowane*

Uczenie nadzorowane jest jednym z głównych paradygmatów uczenia maszynowego, w którym model jest trenowany na podstawie etykietowanych danych uczących. Obejmuje to proces uczenia modelu na podstawie danych wejściowych (cech) oraz odpowiadających im wyjść (etykiet), tak aby mógł on przewidywać etykiety dla nowych, nieznanych danych [76]. Dwa główne typy zadań, które można rozwiązać za pomocą uczenia nadzorowanego, to klasyfikacja i regresja. Klasyfikacja polega na przyporządkowaniu obiektu do jednej z predefiniowanych klas, podczas gdy regresja ma na celu przewidzenie ciągłej wartości wyjściowej [77].

Istnieje wiele algorytmów uczenia nadzorowanego, takich jak maszyny wektorów nośnych (SVM), regresja logistyczna, drzewa decyzyjne i sieci neuronowe. Wybór odpowiedniego algorytmu zależy od natury danych i problemu, który ma zostać rozwiązany [78]. Podczas procesu uczenia, model jest optymalizowany poprzez minimalizację funkcji kosztu. Funkcja ta mierzy różnicę między rzeczywistymi etykietami a etykietami przewidzianymi przez model. Popularne funkcje kosztu to błąd średniokwadratowy dla regresji i entropia krzyżowa dla klasyfikacji [79]. Aby ocenić wydajność modelu, często dzieli się dane na zbiory treningowe, walidacyjne i testowe. Zbiór treningowy służy do trenowania modelu, zbiór walidacyjny do dostrajania hiper-parametrów, a zbiór testowy do ostatecznej oceny modelu [80]. Istnieje wiele metryk oceny, takich jak dokładność, precyzja, czułość i F1 dla klasyfikacji oraz błąd średniokwadratowy dla regresji, które omówione zostaną jeszcze w dalszej części rozdziału [81].

Uczenie nadzorowane ma szerokie zastosowanie w różnych dziedzinach, takich jak rozpoznawanie obrazów, analiza sentymentu, prognozowanie cen i wiele innych. Jest to

podstawowa technika w dziedzinie sztucznej inteligencji i uczenia maszynowego [82]. Poprzez odpowiedni dobór algorytmów, funkcji kosztu i metryk oceny, można stworzyć skuteczne modele do rozwiązywania różnorodnych problemów.

5.2.1.2. *Uczenie nienadzorowane*

Uczenie nienadzorowane to paradygmat w dziedzinie uczenia maszynowego, gdzie algorytmy uczą się wzorców wyłącznie z nieoznaczonych danych, w przeciwieństwie do uczenia nadzorowanego i ze wzmocnieniem [83]. Jest to kluczowy aspekt w dziedzinie analizy danych i statystyki, gdzie głównym celem jest wyodrębnienie i wyjaśnienie cech danych. Jednym z klasycznych przykładów uczenia nienadzorowanego w badaniu sieci neuronowych jest zasada Donalda Hebb'a, znana jako "neurony, które razem się aktywują, razem się łączą" [55]. Hebbian Learning, jak również metody takie jak Self-Organizing Map (SOM) i Adaptive Resonance Theory (ART), są powszechnie stosowane w dziedzinie uczenia nienadzorowanego [84-86], do popularnych metod należą także autoenkodery [107], oraz generatywne sieci adwersarialne (GAN) [108]. Metod uczenia nienadzorowanego używa się też często jako wstępny trening dla głębokich sieci neuronowych [109].

W dziedzinie statystyki, jednym z głównych zastosowań uczenia nienadzorowanego jest estymacja gęstości [87]. Metody takie jak analiza składowych głównych i analiza skupień są często stosowane w celu grupowania danych o wspólnych atrybutach [88]. Analiza skupień, na przykład, identyfikuje wspólne cechy w danych i reaguje na obecność lub brak takich wspólnych cech w każdym nowym fragmencie danych. W uczeniu nienadzorowanym stosuje się także różne podejścia do nauki modeli zmiennych ukrytych, takie jak algorytm oczekiwania-maksymalizacji [101], metoda momentów [102] i techniki rozdzielania sygnałów ślepych (na przykład analiza składowych głównych [103], analiza składowych niezależnych [104], faktoryzacja macierzy nieujemnej [105] czy rozkład wartości osobliwych [106]). Metoda momentów, na przykład, jest skuteczna w uczeniu parametrów modeli zmiennych ukrytych, takich jak modelowanie tematów [89].

Warto również zauważyć, że uczenie nienadzorowane odgrywa kluczową rolę w dziedzinie sieci neuronowych. Podczas fazy uczenia, nienadzorowana sieć próbuje naśladować dane, które otrzymuje, i korzysta z błędu w swoim naśladowanym wyjściu, aby się poprawić [90]. W przeciwieństwie do dominującego użycia wstecznego rozprzestrzeniania w metodach nadzorowanych [91], uczenie nienadzorowane wykorzystuje również inne metody, takie jak reguła uczenia Hopfielda [92], reguła uczenia Boltzmana [93], „wake-sleep” [94], wnioskowanie wariacyjne [95], prawdopodobieństwo maksymalne

[96], maksymalny a- posteriori [97], próbkowanie Gibbsa [98] i błędy rekonstrukcji wstecznej propagacji [99] lub reparametryzację stanu ukrytego [100].

5.2.1.3. Uczenie przez wzmacnianie

Uczenie przez wzmacnianie (z ang. „reinforcement learning”) to ostatni z trzech głównych nurtów uczenia maszynowego, skoncentrowany na interakcji ze środowiskiem w celu maksymalizacji nagrody. W odróżnieniu od uczenia nadzorowanego i nienadzorowanego, RL nie korzysta z przygotowanego zestawu danych, ale tworzy środowisko, z którego model zbiera dane [110]. Proces ten obejmuje trzy główne elementy: środowisko, agenta i bufor. Środowisko to zadanie lub symulacja, z którym agent wchodzi w interakcję, a bufor przechowuje zebrane informacje. Agent, który może być reprezentowany przez sieć neuronową lub funkcję losową, dąży do maksymalizacji nagrody poprzez interakcję ze środowiskiem. Proces uczenia się składa się z dwóch etapów: wstępnego zbierania danych i pętli uczenia, która obejmuje zbieranie danych, uczenie i ewaluację [111].

W uczeniu przez wzmacnianie, polityka (ang. „policy”) odnosi się do strategii lub planu działania, który agent stosuje w danym stanie, aby podejmować decyzje i jest używana przez dużą grupę algorytmów omawianych w kolejnym akapicie. Jest to funkcja, która na wejście przyjmuje obserwację środowiska (stan) i zwraca akcję, którą agent powinien podjąć w tym stanie [112]. Polityka może być deterministyczna, gdzie dla danego stanu zawsze zwraca tę samą akcję [113], lub stochastyczna, gdzie zwraca rozkład prawdopodobieństwa nad dostępnymi akcjami [114]. Może być reprezentowana na różne sposoby, na przykład jako tabela, sieć neuronowa, czy prosta funkcja [115]. Optymalizacja polityki, czyli znalezienie najlepszej strategii działania w danym środowisku, jest głównym celem procesu uczenia przez wzmacnianie [116].

Metody uczenia przez wzmacnianie są złożone i różnorodne. Q-Learning to technika, która uczy się wartości akcji w danym stanie [117], wykorzystując równanie Bellmana [118]. Deep Q Network (DQN) rozwija Q-Learning, stosując głębokie sieci neuronowe do aproksymacji funkcji wartości [119]. Policy Gradients [120] i REINFORCE [121] to metody, które bezpośrednio optymalizują politykę, zamiast uczyć się wartości akcji. Actor-Critic Methods łączą cechy metod wartości i gradientów polityki, próbując zrównoważyć eksplorację i eksploatację [122]. Proximal Policy Optimization (PPO) [123] i Trust Region Policy Optimization (TRPO) [124] wprowadzają ograniczenia na zmiany polityki dla stabilności. SARSA [125] to algorytm, który uczy się wartości akcji w bieżącej polityce. Temporal Difference Learning [126] łączy różnice czasowe i nagrody, aby oszacować

wartości stanów. Rainbow [127, 128] łączy różne techniki DQN, takie jak priorytetowe próbkowanie [129], podwójne DQN [130] i inne [128]. Soft Actor-Critic (SAC) to algorytm pozbawiony polityki, który automatycznie dostosowuje temperaturę, kontrolując stopień eksploracji [131].

W kontekście uczenia przez wzmacnianie, temperatura jest parametrem w niektórych algorytmach, który kontroluje stopień eksploracji i eksploatacji w procesie wyboru akcji przez agenta [132]. Temperatura reguluje, jak bardzo prawdopodobieństwa wyboru różnych akcji różnią się od siebie. Gdy temperatura jest wysoka, prawdopodobieństwa są bardziej jednolite, co prowadzi do większej eksploracji, ponieważ agent jest bardziej skłonny do próbowania różnych akcji. Gdy temperatura jest niska, prawdopodobieństwa są bardziej skoncentrowane na najlepszych akcjach, co prowadzi do większej eksploatacji, ponieważ agent skupia się na wykorzystaniu swojej obecnej wiedzy [133].

C51 modeluje rozkład prawdopodobieństwa nagród, zamiast oczekiwanej wartości [134]. Podejście *model-based RL* uczy się modelu przejścia środowiska, co może przyspieszyć uczenie się przez lepsze zrozumienie dynamiki środowiska [135]. Wybór odpowiedniej metody zależy od konkretnego problemu, wymagań dotyczących wydajności, złożoności i interpretowalności.

5.2.1.4. Wybór odpowiedniej grupy algorytmów do badań

Jak wspomniał Nasteski [136], nadzorowane uczenie maszynowe to proces, który trenuje model w celu przewidywania wyników na podstawie danych, przy jednoczesnym uwzględnieniu niepewności. Technika ta wymaga znanych danych wejściowych i wyjściowych, aby wytrenować model do przewidywania nowych danych. Jeśli dostępne są znane dane wyjściowe, uczenie nadzorowane jest odpowiednią metodą, tak jak w przypadku badanego przypadku. Cechy które sprzyjały uczeniu nadzorowanego, w kontekście redukcji niespójności zgrupowano w Tabeli 5.1.

Techniki klasyfikacji i regresji są wykorzystywane w uczeniu nadzorowanym do opracowywania modeli uczenia maszynowego. Techniki klasyfikacji są wykorzystywane od dłuższego czasu do przewidywania dyskretnych odpowiedzi, takich jak klasyfikowanie wiadomości e-mail jako spam [137], lub określanie, czy guz jest łagodny czy nowotworowy [138]. Z drugiej strony, techniki regresji są wykorzystywane do przewidywania odpowiedzi ciągłych, takich jak stan naładowania baterii [140], obciążenie sieci elektrycznej [141] lub cena aktywów finansowych [142] - jeśli zakres danych jest szeroki lub jeśli charakter odpowiedzi jest liczbą rzeczywistą, należy zastosować właśnie te techniki [139].

Tabela 5.1. Cechy różnych grup uczenia maszynowego w kontekście redukcji niespójności

Cecha	Uczenie nadzorowane	Uczenie nienadzorowane	Uczenie ze wzmocnieniem
Dostępność danych wejściowych i wyjściowych	Tak	Nie	Częściowo
Predykcja ciągłych odpowiedzi	Tak	Nie	Nie
Zgodność z charakterem problemu	Tak	Nie	Nie
Rozważanie niepewności	Tak	Nie	Częściowo
Modelowanie zależności i interpretowalność	Tak	Nie	Częściowo
Skalowalność działań	Tak	Nie	Częściowo
Złożoność obliczeniowa	Zmienna	Zmienna	Zwykle wysoka
Wrażliwość na szum w danych	Często tak	Często nie	Często nie

Przy podejmowaniu decyzji na temat wyboru odpowiedniej grupy algorytmów do badań (oraz decydując się na uczenie nadzorowane i algorytmy regresji), kierowano się następującymi powodami, popartymi dogłębną analizą literatury dziedzinowej oraz doświadczeniami eksperymentalnymi:

- **Dostępność danych wejściowych i wyjściowych:** Uczenie nadzorowane wymaga znanych danych wejściowych i wyjściowych do trenowania modelu [143]. W przypadku algorytmu iteracyjnego Szybowskiego, dane te są dostępne, co czyni uczenie nadzorowane odpowiednim wyborem.
- **Predykcja ciągłych odpowiedzi:** Algorytmy regresji są zaprojektowane do przewidywania ciągłych odpowiedzi, co oznacza, że mogą one modelować zależności, gdzie wynik jest liczbą rzeczywistą [143]. W kontekście redukcji niespójności macierzy porównań parami, odpowiedź jest ciągła, ponieważ niespójność może być mierzona jako wartość rzeczywista [144]. Na przykład, może to być różnica między rzeczywistą a przewidywaną wartością w macierzy. Algorytmy regresji są więc idealnie dopasowane do tego zadania, ponieważ mogą modelować te ciągłe zależności i dostarczyć precyzyjne prognozy [145].
- **Zgodność z charakterem problemu:** Algorytmy regresji są szczególnie odpowiednie, gdy zakres danych jest szeroki i natura odpowiedzi to liczba rzeczywista [146]. W przypadku macierzy porównań parami, zakres danych jest szeroki, ponieważ może obejmować różne parametry i ich wzajemne relacje. Ponadto, odpowiedź, czyli miara niespójności, jest liczbą rzeczywistą. Algorytmy regresji są więc w stanie obsłużyć te specyficzne właściwości problemu, dostarczając model, który jest zgodny z charakterem danych i naturą odpowiedzi [146].
- **Rozważanie niepewności:** Uczenie nadzorowane, a w szczególności techniki regresji, mogą uwzględniać niepewność w procesie prognozowania [147]. Niepewność może

odgrywać kluczową rolę w kontekście iteracyjnego algorytmu Szybowskiego, gdzie różne parametry mogą wpływać na siebie w określony sposób [6]. Algorytmy regresji mogą modelować te złożone zależności, uwzględniając niepewność w danych i strukturze modelu [148]. Na przykład, techniki takie jak regresja Bayesowska mogą jawnie modelować niepewność, dostarczając nie tylko punktową prognozę, ale także przedział ufności [149]. To rozważanie niepewności może prowadzić do bardziej zrównoważonych i wiarygodnych prognoz [150], co jest kluczowe dla skutecznego zarządzania niespójnością w macierzy porównań parami.

- **Modelowanie zależności i interpretowalność:** Regresja pozwala na modelowanie zależności między różnymi parametrami w macierzy, co może pomóc w zrozumieniu, jak te parametry wpływają na siebie nawzajem [151-153]. To zrozumienie może być kluczem do wykrycia i naprawy niespójności. Łatwa interpretowalność modeli regresji może dostarczyć solidnych podstaw do podejmowania decyzji dotyczących tego, jak naprawić niespójność w macierzy [154-158]. Zrozumienie, dlaczego model podejmuje określone decyzje, może pomóc w podejmowaniu świadomych i uzasadnionych decyzji dotyczących poprawek [159, 160]. Techniki regresji często można wizualizować za pomocą wykresów i diagramów [161-164]. Na przykład, można stworzyć wykres pokazujący, jak poszczególne parametry wpływają na niespójność. Wizualizacje te mogą ułatwić zrozumienie modelu i jego działania.
- **Skalowalność działań:** Techniki regresji mogą być zautomatyzowane i zastosowane do dużych macierzy porównań parami [165-169], co czyni je wydajnym narzędziem do analizy i poprawy spójności na dużą skalę.

Właśnie z tych powodów w niniejszym rozdziale wykorzystano cztery algorytmy regresji z tej gałęzi technik uczenia maszynowego, opisane przez Czajkowskiego i Kretowskiego [170]: regresję drzewa decyzyjnego, wielokrotną regresję liniową, regresję k-najbliższych sąsiadów i regresję perceptronu wielowarstwowego. Dlaczego zdecydowano się na te konkretne metody regresji?

Głównym celem tego badania jest zbadanie potencjału różnych technik regresji w kontekście redukcji niespójności macierzy porównań parami. Nie jest to próba wyczerpania wszystkich dostępnych metod, ale raczej skoncentrowane badanie kilku wybranych technik. Badania są czasochłonne i zasobożerne [171]. Wybór zbyt wielu technik mógłby znacznie wydłużyć czas badania i zwiększyć jego koszty [172]. Skoncentrowanie się na kilku wybranych technikach pozwoli na głębszą analizę i lepsze zrozumienie ich działania.

Drzewa decyzyjne są łatwe do zrozumienia i interpretacji [173], co może być kluczowe w zrozumieniu, jak niespójność jest modelowana [174]. Mogą dobrze radzić sobie z niejednorodnymi danymi [175], co może być istotne w kontekście macierzy porównań parami. Wielokrotna regresja liniowa jest z kolei idealna do modelowania liniowych zależności między zmiennymi [176], co może być odpowiednie w przypadku niektórych macierzy porównań parami. W dodatku działa efektywnie na dużych zestawach danych [177-180]. Regresja k-najbliższych sąsiadów nie zakłada konkretnej formy funkcji [181], co może być korzystne, gdy zależności są nieliniowe lub złożone [182, 183]. Oferuje też możliwość dostosowania przez wybór odpowiedniej liczby sąsiadów i metryki odległości [184-187]. MLP, jako forma sieci neuronowej, może modelować złożone, nieliniowe zależności [188-190], więc może być odpowiednie, gdy macierz porównań parami ma skomplikowaną strukturę [191] i zbadać potencjał do badań w tym kierunku (sieci neuronowych).

Wybrane techniki są powszechnie stosowane i istnieją narzędzia i biblioteki, które ułatwiają ich implementację i testowanie (zostaną opisane w dalszej części rozdziału). Odrzucono natomiast inne techniki regresji:

- **Regresje grzbietowa (L2) i Lasso (L1) [192]** - Te techniki są odmianami regresji liniowej, które wprowadzają regularyzację, aby zapobiec przeuczeniu [193]. Zostały odrzucone, ponieważ wielokrotna regresja liniowa już została uwzględniona, a te dodatkowe techniki nie wносиłyby znacząco nowych perspektyw w kontekście badanego problemu [211-213].
- **Regresje SVM [194]** – te techniki posiada bardzo wiele możliwości, ale jest trudna w optymalizacji i interpretacji [195-198], zwłaszcza w przypadku nieliniowych jąder [199]. W kontekście tego badania, uznano, że inne techniki lepiej pasują do charakteru danych i celów badania [203], zaś czas badań wywnioskowany z literatury [200-202], nie przystawał do znaczenia tego w odniesieniu do całej pracy
- **Regresja logistyczna [204-206]** - jest używana głównie do modelowania prawdopodobieństwa wyniku binarnego [207-209]. Ponieważ cel badania dotyczył ciągłej odpowiedzi, regresja logistyczna nie była odpowiednia [210].
- **Regresja wielomianowa [214]** - może modelować złożone, nieliniowe zależności, ale jest podatna na przeuczenie [215-217], zwłaszcza przy wyższych stopniach wielomianu [218-219]. Ze względu na te ryzyka i potencjalną trudność w optymalizacji [220], została ona odrzucona.
- **Regresja gamma [221]** - jest specyficzną formą regresji stosowaną do modelowania czasu do zdarzenia [222] lub innych zmiennych o rozkładzie gamma [223-226].

Została odrzucona z powodu niewłaściwej formy danych i braku dopasowania do charakteru problemu.

- **Regresja ElasticNet [227]** - łączy cechy regresji Ridge i Lasso [228-230]. Została odrzucona z powodów technicznych (problemy z dostosowaniem do konkretnego zestawu danych).
- **Regresja kwantylowa [231]** - ta technika może być używana do modelowania różnych kwantyli rozkładu odpowiedzi [232-235]. Została odrzucona, ponieważ cel badania nie wymagał analizy różnych kwantyli [236], a skupienie się na średniej odpowiedzi było bardziej odpowiednie.
- **Regresja Poissona [237]** - jest stosowana do modelowania liczby zdarzeń zachodzących w określonym czasie lub przestrzeni [238-241]. Została odrzucona, ponieważ nie pasowała do charakteru danych i problemu (nacisk na dane hierarchiczne [242]).
- **Regresja Cox'a (model proporcjonalnego hazardu) [243]** - ta technika jest specyficzna dla analizy przeżycia (survivalu) (244-246) i modelowania ryzyka zdarzenia w czasie (247-250). Została odrzucona, ponieważ nie przystawała do scenariusza redukcji niespójności macierzy.
- **Regresja z wykorzystaniem techniki Fused Lasso [251]** - Fused Lasso jest techniką, która może narzucać zarówno gładkość [255-257], jak i rzadkość na współczynniki regresji [258-260]. Została odrzucona z powodu złożoności [252-254] i braku bezpośredniego zastosowania do problemu.
- **Regresja odporna - [261, 265]** - jest odporna na odstające wartości, zależne jest to też od przyjętego estymatora [262-264, 266-268], ale została odrzucona, gdyż odstające wartości nie były znaczącym problemem w danych uczących (o czym szerzej w dalszej części rozdziału).

Tabela 5.2. Cechy pożądane do badania potencjału w technikach regresji

Technika Regresji	Łatwość Interpretacji	Skalowalność	Odporność na przeuczenie	Złożoność	Odporność na odstające wartości	Modelowanie zależności
Regresja drzewa decyzyjnego	Tak	Nie	Nie	Niska	Nie	Tak
Wielokrotna regresja liniowa	Tak	Tak	Nie	Niska	Nie	Tak
Regresja k-najbliższych sąsiadów	Nie	Nie	Nie	Średnia	Nie	Tak
Regresja perceptronu wielowarstwowego	Nie	Tak	Nie	Wysoka	Nie	Tak
Regresja grzbietowa (L2)	Nie	Nie	Tak	Średnia	Nie	Nie

Regresja Lasso (L1)	Nie	Nie	Tak	Średnia	Nie	Nie
Regresja SVM	Nie	Nie	Nie	Wysoka	Nie	Nie
Regresja logistyczna	Nie	Nie	Nie	Średnia	Nie	Nie
Regresja wielomianowa	Nie	Nie	Nie	Wysoka	Nie	Tak
Regresja gamma	Nie	Nie	Nie	Wysoka	Nie	Nie
Regresja ElasticNet	Nie	Nie	Tak	Średnia	Nie	Nie
Regresja kwantylowa	Nie	Nie	Nie	Średnia	Nie	Nie
Regresja Poissona	Nie	Nie	Nie	Średnia	Nie	Nie
Regresja Cox'a	Nie	Nie	Nie	Wysoka	Nie	Nie
Regresja z wykorzystaniem techniki Fused Lasso	Nie	Nie	Tak	Wysoka	Nie	Nie
Regresja odporna	Nie	Nie	Nie	Średnia	Tak	Nie
Model efektów losowych	Nie	Nie	Nie	Wysoka	Nie	Nie
Gradient boosting	Nie	Nie	Nie	Wysoka	Nie	Tak
Procesy Gaussowskie	Nie	Nie	Nie	Wysoka	Nie	Tak
Modele szeregów czasowych	Nie	Nie	Nie	Wysoka	Nie	Nie

- **Model efektów losowych [269]** - ta technika może być używana do modelowania danych hierarchicznych lub zagnieżdżonych [270-274]. Struktura danych nie wymagała takiego podejścia.
- **Gradient Boosting [275]** - jest techniką, która jest trudna w dostrojeniu [276-278] i optymalizacji [279-280] i została zarzucona z powodu tych wyzwań i potencjalnej trudności w interpretacji [281-283].
- **Procesy gaussowskie [284]** – są zbyt złożone i czasochłonne w obliczeniach dla dużych zbiorów danych [285-287].
- **Modele szeregów czasowych (np. ARIMA [288])** - dane nie miały struktury szeregów czasowych

Wybór technik regresji w badaniu był wynikiem starannego rozważenia różnych opcji i odrzucenia tych, które nie pasowały do charakteru problemu, danych, celów badania, oraz ograniczeń zasobów i czasu. Podsumowanie zostało zgromadzone także w Tabeli 5.2. Wybrane techniki były uważane za najbardziej odpowiednie i obiecujące w kontekście specyficznych wymagań i oczekiwań badania. Zostaną opisane w kolejnym rozdziale.

5.2.2. Wybrane do badań techniki regresji

Jak zostało wspomniane i uzasadnione w poprzednim podrozdziale, do badań zostały wybrane cztery techniki regresji – regresja drzew decyzyjnych, wielokrotna regresja

liniowa, regresja k-najbliższych sąsiadów oraz regresja perceptronu wielowarstwowego. Zostały one szeroko opisane w literaturze, gdzie można zapoznać się z szczegółową ich budową oraz zastosowaniem w konkretnych przypadkach, zaś poniżej opisano podstawowe założenia dotyczące ich mechanizmów działania:

- **Regresja drzew decyzyjnych:** drzewiasta struktura jest wykorzystywana przez drzewa decyzyjne do konstruowania modeli regresji lub klasyfikacji. Zbiór danych jest dzielony na coraz mniejsze podzbiory, przy jednoczesnym rozwijaniu powiązanego drzewa decyzyjnego. Wynikowe drzewo zawiera węzły decyzyjne i węzły liści. Węzeł decyzyjny ma wiele gałęzi, z których każda odpowiada wartościom testowanego atrybutu. Węzeł liścia oznacza decyzję dotyczącą celu liczbowego. Węzeł główny, który odpowiada najlepszemu predyktorowi, jest najwyższym węzłem decyzyjnym w drzewie. Drzewa decyzyjne mogą obsługiwać zarówno dane kategoryczne, jak i numeryczne [289-299].
- **Wielokrotna regresja liniowa:** jest to technika statystyczna służąca do opracowania modelu, który przedstawia liniowe połączenie między celem zależnym a jedną lub większą liczbą zmiennych niezależnych. Aby MLR była dokładna, musi być spełnionych kilka założeń, takich jak normalny rozkład błędów ze średnią równą zero i stałą wariancją. Bezstronne estymatory mają wartość oczekiwaną równą prawdziwej wartości parametru, efektywne estymatory mają mniejszą wariancję niż jakikolwiek inny estymator, a spójne estymatory mają odchylenia i wariancje, które zbliżają się do zera, gdy wielkość próby zbliża się do nieskończoności [300-309].
- **Regresja k-najbliższych sąsiadów:** algorytm K-najbliższych sąsiadów (KNN) jest prostą metodą, która zachowuje wszystkie dostępne przypadki i tworzy prognozy dla celów numerycznych, wykorzystując miarę podobieństwa, taką jak funkcje odległości. KNN jest stosowana jako technika nieparametryczna w estymacji statystycznej i rozpoznawaniu wzorców od wczesnych lat 70-tych. Aby wykonać regresję KNN, jedna prosta implementacja polega na obliczeniu średniej celu numerycznego dla K-najbliższych sąsiadów. Inne podejście polega na obliczeniu średniej ważonej odwrotnej odległości dla K najbliższych sąsiadów. Funkcje odległości stosowane w klasyfikacji KNN są również wykorzystywane w regresji KNN [310-319].
- **Regresja perceptronu wielowarstwowego:** Sztuczna sieć neuronowa z perceptronem wielowarstwowym (MLP) składa się z trzech lub więcej warstw perceptronów. Warstwy te składają się z pojedynczej warstwy wejściowej, jednej lub więcej warstw ukrytych i pojedynczej warstwy wyjściowej perceptronu. Dane w tej

sieci przesuwają się wyłącznie do przodu, zaczynając od warstwy wejściowej i przechodząc przez warstwy ukryte do warstwy wyjściowej. MLP wykorzystuje propagację wsteczną w celu zwiększenia dokładności przewidywania, co pociąga za sobą otrzymywanie informacji zwrotnych na temat wad w danych wyjściowych, a następnie odpowiednie modyfikowanie wag [320-329].

5.2.3. Wskaźniki dot. skuteczności wybranych technik regresji

W dziedzinie uczenia maszynowego, ocena i optymalizacja modeli stały się kluczowymi elementami procesu budowy efektywnych i niezawodnych systemów. Podrozdział ten skupia się na miarach i metodach, które są używane do oceny ich skuteczności.

Błąd średniokwadratowy (MSE) jest powszechnie stosowaną metryką w różnych technikach regresji [330]. W regresji drzew decyzyjnych może służyć jako kryterium podziału, ale może prowadzić do przeuczenia w przypadku zbyt złożonych drzew [331-333]. W wielokrotnej regresji liniowej i regresji perceptronu wielowarstwowego, jest często używany jako funkcja straty, z uwzględnieniem ryzyka niewłaściwego dopasowania lub utknięcia w lokalnych minimum [334-338]. W regresji k-NN, MSE może pomóc w wyborze optymalnej liczby sąsiadów, z równowagą między przeuczeniem a niedouczeniem [339-340]. Choć błąd średniokwadratowy (MSE) jest powszechnie stosowany, to jego specyficzne zastosowanie i interpretacja mogą być bardziej złożone:

- **W regresji drzew decyzyjnych:** MSE może służyć jako kryterium podziału, ale zbyt silne poleganie na nim może prowadzić do przeuczenia, zwłaszcza w przypadku zbyt złożonych drzew. Niuanssem jest tu równowaga między dopasowaniem a generalizacją, gdzie zbyt dokładne dopasowanie do danych treningowych może pogorszyć wydajność na nowych danych [331-333].
- **W wielokrotnej regresji liniowej:** MSE jest często używany jako funkcja straty, ale istnieje ryzyko niewłaściwego dopasowania, zwłaszcza gdy relacje między zmiennymi nie są liniowe [336-337]. Ponadto, MSE jest wrażliwe na wartości odstające, co może prowadzić do mylących wyników, jeśli takie wartości są obecne w danych [338].
- **W regresji perceptronu wielowarstwowego:** MSE jest również używany jako funkcja straty, ale może prowadzić do utknięcia w lokalnych minimum podczas procesu optymalizacji. To oznacza, że model może nie osiągnąć najlepszego możliwego dopasowania, jeśli proces optymalizacji zatrzyma się w suboptymalnym punkcie [334-335].

- **W regresji k-NN:** MSE może pomóc w wyborze optymalnej liczby sąsiadów, ale niuansiem jest tu znalezienie odpowiedniej równowagi między przeuczeniem a niedouczeniem. Zbyt mała liczba sąsiadów może prowadzić do modelu, który jest zbyt złożony i przeucza się na danych treningowych, podczas gdy zbyt duża liczba sąsiadów może prowadzić do modelu, który jest zbyt prosty i nie dopasowuje się wystarczająco dobrze do danych [339-340].

Średni błąd bezwzględny (MAE) to metryka, która mierzy średnią wartość bezwzględnych różnic między rzeczywistymi a przewidywanymi wartościami, stosowana w różnych technikach regresji [341]. W regresji drzew decyzyjnych, wielokrotnej regresji liniowej, regresji k-NN i regresji perceptronu wielowarstwowego, MAE może służyć jako bardziej wytrzymała na wartości odstające alternatywa dla MSE [342-345].

Dzięki temu, że nie podnosi różnic do kwadratu, jest mniej wrażliwy na ekstremalne wartości [346]. MAE może być używane do oceny ogólnej wydajności modelu, dostarczając intuicyjnego i bezpośredniego pomiaru błędu. Jego wybór może być szczególnie korzystny w sytuacjach, gdy wartości odstające są niepożądane lub nieistotne dla analizy [346-349]. Średni błąd bezwzględny (MAE) jest cenną metryką w różnych technikach regresji, ale jego zastosowanie i interpretacja mają swoje niuanse i złożoności:

- **W regresji drzew decyzyjnych:** MAE może być mniej podatny na przeuczenie w porównaniu z MSE, ponieważ nie karze modelu tak silnie za duże błędy [350]. Niuansiem jest tu wybór między MAE a MSE w zależności od tego, czy wartości odstające są istotne dla analizy [351].
- **W wielokrotnej regresji liniowej:** MAE jest bardziej odporny na wartości odstające niż MSE, co może prowadzić do bardziej stabilnych i wiarygodnych oszacowań w obecności takich wartości [352-354]. Niuansiem jest tu zrozumienie, jak wartości odstające wpływają na analizę i czy powinny być one zminimalizowane czy uwzględnione [353].
- **W regresji k-NN:** MAE może dostarczyć bardziej zrównoważony obraz błędu modelu, ponieważ nie nadmiernie karze duże błędy [355-357]. Niuansiem jest tu zrozumienie, jak różne wybory metryk wpływają na wybór sąsiadów i ostateczne przewidywania [355-357].
- **W regresji perceptronu wielowarstwowego:** MAE może być mniej wrażliwy na lokalne minimum w procesie optymalizacji, ponieważ nie podnosi błędów do kwadratu. Niuansiem jest tu zrozumienie, jak różne funkcje straty wpływają na proces uczenia i optymalizacji [358-360].

Współczynnik determinacji (R^2) jest metryką, która wyraża, jak dobrze przewidywania modelu dopasowują się do rzeczywistych danych, i jest stosowany w różnych technikach regresji, takich jak regresja drzew decyzyjnych, wielokrotna regresja liniowa, regresja k-NN i regresja perceptronu wielowarstwowego [362-367].

R^2 mierzy proporcję wariancji w zmiennej zależnej, która jest przewidywana z niezależnych zmiennych, przyjmując wartości od 0 do 1. Wartość R^2 równa 1 oznacza doskonałe dopasowanie, podczas gdy wartość 0 wskazuje, że model nie wyjaśnia żadnej wariancji zmiennej zależnej.

R^2 jest często używany do oceny, czy model jest odpowiedni dla danych, i może pomóc w porównywaniu różnych modeli [368]. Jednakże może być mylący, jeśli został zastosowany nieprawidłowo, zwłaszcza gdy liczba zmiennych objaśniających jest duża, co może prowadzić do zawyżonego R^2 , nawet jeśli model nie jest odpowiedni [369]. Współczynnik determinacji jest powszechnie stosowaną metryką w analizie regresji, ale jego zastosowanie i interpretacja mają swoje niuanse:

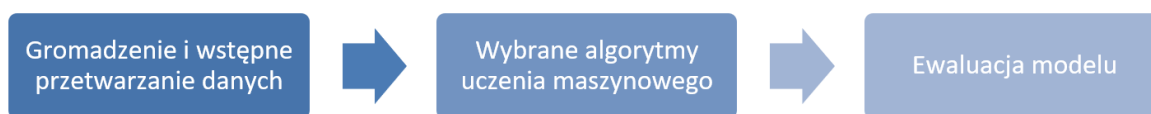
- **W regresji drzew decyzyjnych:** R^2 może być używany do oceny dopasowania drzewa, ale może być mylący, jeśli drzewo jest zbyt złożone i doskonale dopasowuje się do danych treningowych, prowadząc do przeuczenia [370-372].
- **W wielokrotnej regresji liniowej:** R^2 jest często używany do oceny dopasowania modelu, ale może być zawyżony, jeśli model zawiera zbyt wiele zmiennych objaśniających, [363] co może prowadzić do niewłaściwych wniosków o jego skuteczności.
- **W regresji k-NN:** R^2 może być używany do oceny, jak dobrze model generalizuje na nowe dane, ale niuansiem jest tu zrozumienie, że wysoki R^2 na danych treningowych niekoniecznie przekłada się na wysoką wydajność na danych testowych [373-374].
- **W regresji perceptronu wielowarstwowego:** R^2 może być używany do oceny dopasowania sieci neuronowej, ale niuansiem jest tu zrozumienie, że złożoność modelu i proces optymalizacji mogą wpływać na interpretację R^2 [375-377].

Ogólnie rzecz biorąc, R^2 jest użytecznym narzędziem do oceny, jak dobrze model przewiduje zmienną zależną, ale jego właściwe zastosowanie wymaga ostrożności. Wysoki R^2 nie zawsze oznacza dobry model, zwłaszcza jeśli model jest zbyt złożony lub zawiera zbyt wiele zmiennych objaśniających.

Ponadto, R^2 może być mniej użyteczny w przypadku nieliniowych relacji, gdzie inne metryki mogą dostarczyć bardziej oświecających informacji. Wybór i interpretacja R^2 powinny być oparte na solidnym zrozumieniu jego właściwości i ograniczeń, a także na

specyfice problemu i danych, z którymi się pracuje, podobnie zresztą jak we wszystkich omawianych miarach i technikach.

5.3. Metodyka badań



Rys. 5.2. Proces badań

Metodyka badań przedstawiona w tym podrozdziale to wszechstronne i systematyczne podejście do zrozumienia skuteczności stosowania technik uczenia maszynowego w celu zmniejszenia niespójności w macierzach porównań parami. Proces ten jest zobrazowany na Rys. 5.2 i obejmuje kilka kluczowych etapów, z których każdy przyczynia się do ogólnej integralności i solidności badań:

- **Zbieranie i przetwarzanie danych:**
 - Początkowa faza polega na generowaniu i przetwarzaniu 100 000 macierzy porównań parami 4x4. Wykorzystując narzędzie PC MATRICES GENERATOR (opisane w kolejnym rozdziale) wybrano macierze o określonych zakresach niespójności i przetworzono je za pomocą algorytmu Szybowskiego. Następnie dane zostały oczyszczone, podzielone i przygotowane do algorytmów uczenia maszynowego, zapewniając, że macierze były wystarczająco zgodne do dalszej analizy.
- **Zastosowanie technik uczenia maszynowego:**
 - Istota podejścia tkwi w zastosowaniu różnych modeli uczenia maszynowego do zmniejszenia niespójności w macierzach. Wykorzystując biblioteki Pythona, takie jak pandas, numpy i scikit-learn, badanie stosuje techniki, takie jak regresja liniowa, k-najbliższych sąsiadów, regresja drzewa decyzyjnego i regresja wielowarstwowego perceptronu. Modele są trenowane, testowane i oceniane, a wyniki są starannie dokumentowane i przechowywane.
- **Kryteria oceny:**
 - Ostatni etap metodologii skupia się na ocenie proponowanego podejścia. Najpierw obliczane są wskaźniki zaprezentowane w Rozdziale 5.2.3.2, a następnie poprawność wybranego typu regresji jest oceniana na podstawie wartości *CR*, a następnie przeprowadzana jest szczegółowa analiza przy

użyciu różnych metryk, takich jak odległość tau Kendalla i odległość między dwoma macierzami (opisanych już wcześniej). Wyniki są następnie porównywane z tymi uzyskanymi przez iteracyjne "klasyczne" algorytmy redukcji niespójności macierzy porównań parami.

5.3.1. Zbieranie i przetwarzanie danych

100 000 wygenerowanych macierzy porównań parami 4x4, pochodzących z narzędzia PC MATRICES GENERATOR (opisanego w kolejnym rozdziale), zostało użytych jako dane wejściowe do nauki i testowania algorytmów uczenia maszynowego przedstawionych we wcześniejszej części rozdziału. Narzędzie to zostało wybrane, ponieważ umożliwiała generowanie macierzy o wysokim stopniu niespójności, dzięki możliwości wyboru zakresów współczynnika CR. Wybrano macierze, dla których $0.30 < CR < 0.80$.

Aby uzyskać wynik, wszystkie 100 000 macierzy zostało ponownie przetworzone przy użyciu algorytmu Szybowskiego opisanego w rozdziale 1.5.2, redukując wskaźnik niezgodności za pomocą tego algorytmu do mniej niż 0.10, co uważa się za wystarczająco zgodne. Algorytm ten został zaimplementowany w ramach biblioteki i narzędzia REDUCE opracowanego przez Kowal i in. (zaprezentowanego w rozdziale 3). Ponieważ dane z tego narzędzia zawierały dodatkowe kolumny, zostały one oczyszczone z nadmiarowych kolumn, a ostateczne dane zawierały pola widoczne w Tabeli 5.3, będącej fragmentem pliku CSV przeznaczonego do trenowania algorytmów.

Tabela 5.3 Fragment pliku CSV przeznaczonego do treningu algorytmu

A11	A12	A13	A14	A21	A22	A23	...	A42	A43	A44	CR
1.0	0.5	0.25	0.12	2.0	1.0	1.0	...	5.0	7.0	1.0	0.051
1.0	4	9.0	0.2	0.2	1.0	5.0	...	9.0	22.35	1.0	0.05
1.0	0.33	6.0	0.33	3.0	1.0	9.0	...	3.0	8.0	1.0	0.0925
...
1.0	1.02	1.57	5.0	0.9	1.0	0.44	...	0.2	0.16	1.0	0.051

Dane zostały następnie podzielone na zbiory liczące po 20 000 macierzy każdy - odpowiednio zestawy uczące i testowe. Tylko takie dane mogły być przetworzone przez algorytmy uczenia maszynowego i tworzyć modele, których implementacja jest opisana szczegółowo w kolejnym podrozdziale. Po podziale na cztery używane algorytmy, pozostałe 20 000 macierzy zostało wykorzystanych do ostatecznej oceny algorytmu analizowanego w niniejszym artykule badawczym.

Proces ten łączy zaawansowane techniki generowania i przetwarzania danych z innowacyjnymi metodami uczenia maszynowego, aby skutecznie zredukować niespójności w macierzach porównań parami.

5.3.2. Zastosowanie technik uczenia maszynowego

Kod do wykonania zadań związanych z uczeniem maszynowym został napisany w języku Python i stosuje kilka bibliotek, takich jak pandas, numpy i scikit-learn, do przeprowadzenia szeregu operacji, w tym wczytywania danych, dzielenia danych, trenowania i testowania różnych modeli uczenia maszynowego oraz obliczania *CR*.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	A11	A12	A13	A14	A21	A22	A23	A24	A31	A32	A33	A34	A41	A42	A43	A44	CR	cr_real	
2	0.10	1.3289	7.0	6.0	0.7525	1.0	9.0	8.0	0.1429	0.1111	1.0	0.3333	0.1667	0.125	3.0	1.0	0.0593	0.059301299084693546	
3	1.10	3.0	5.0	0.25	0.3333	1.0	4.0	0.1429	0.2	0.25	1.0	0.0932	4.0	7.0	10.7293	1.0	0.0567	0.05671999392557785	
4	2.10	5.0	0.5	8.0	0.2	1.0	0.2367	0.8427	2.0	4.2243	1.0	6.0	0.125	1.1867	0.1667	1.0	0.0573	0.057291868028171315	
5	3.10	1.6147	0.125	0.25	0.6193	1.0	0.25	0.4597	8.0	4.0	1.0	1.0019	4.0	2.1755	0.9981	1.0	0.0822	0.08224385014179886	
6	4.10	0.3333	0.3333	3.993	3.0	1.0	0.3333	5.0	3.0	3.0	1.0	5.6499	0.2504	0.2	0.177	1.0	0.084	0.0840258152156074	
7	5.10	0.2	1.0	0.3333	5.0	1.0	3.0	1.0496	1.0	0.3333	1.0	0.971	3.0	0.9527	1.0299	1.0	0.064	0.06404620571016444	
8	6.10	0.3333	7.0	3.0	3.0	1.0	7.0	5.0	0.1429	0.1429	1.0	0.248	0.3333	0.2	4.0326	1.0	0.0717	0.07168744228206317	
9	7.10	3.5515	1.0852	1.0	0.2816	1.0	0.1429	0.125	0.9215	7.0	1.0	2.321	1.0	8.0	0.4308	1.0	0.0731	0.07313046448776156	
10	8.10	3.1321	3.0	7.0	0.3193	1.0	0.5	0.5	0.3333	2.0	1.0	3.0	0.1429	2.0	0.3333	1.0	0.0962	0.09617655926225537	
11	9.10	0.0892	0.2	0.1111	11.2142	1.0	7.0	2.0	5.0	0.1429	1.0	0.1667	9.0	0.5	6.0	1.0	0.0789	0.07896304926369001	
12	10.10	9.0	3.0	1.8744	0.1111	1.0	0.25	0.1429	0.3333	4.0	1.0	0.2	0.5335	7.0	5.0	1.0	0.0634	0.06342834543137425	
13	11.10	2.3523	4.0	0.2546	0.4251	1.0	4.0	0.2	0.25	0.25	1.0	0.1429	3.928	5.0	7.0	1.0	0.0689	0.06888474691211753	
14	12.10	10.7666	3.6071	4.0	0.0929	1.0	0.2	0.1667	0.2772	5.0	1.0	2.4924	0.25	6.0	0.4012	1.0	0.0665	0.06651946437210768	
15	13.10	1.0649	0.1429	1.0926	0.939	1.0	0.25	4.0	7.0	4.0	1.0	9.0	0.9153	0.25	0.1111	1.0	0.0632	0.0631918446969819	
16	14.10	8.0	4.0	9.1779	0.125	1.0	1.6118	3.0	0.25	0.6204	1.0	4.0661	0.109	0.3333	0.2459	1.0	0.0779	0.07797779991239674	
17	15.10	0.3333	4.4475	0.8049	3.0	1.0	9.0	7.0	0.2248	0.1111	1.0	0.5	1.2423	0.1429	2.0	1.0	0.0606	0.060582687783337995	
18	16.10	0.1111	7.0	1.0	9.0	1.0	17.8728	5.0	0.1429	0.056	1.0	0.25	1.0	0.2	4.0	1.0	0.0547	0.05485277462980698	
19	17.10	1.0	0.2146	0.2	1.0	1.0	0.2	0.5	4.6597	5.0	1.0	3.4944	5.0	2.0	0.2862	1.0	0.0657	0.06570545756617438	
20	18.10	0.2792	6.0	0.25	3.5813	1.0	8.0	0.4819	0.1667	0.125	1.0	0.091	4.0	2.0749	10.9924	1.0	0.0513	0.051329401232897094	
21	19.10	4.1994	2.0	3.0	0.2381	1.0	0.1429	0.25	0.5	7.0	1.0	3.6003	0.3333	4.0	0.2778	1.0	0.0928	0.09278212475371346	
22	20.10	6.0	1.0	0.1667	1.0	0.1667	0.1111	1.0	6.0	1.0	1.5197	2.0	9.0	0.658	1.0	0.0482	0.04819824233249391		



Rys. 5.3. Wyniki w plikach CSV

Skrypt najpierw wczytuje dwa zbiory danych (*X* i *y*) z plików CSV, gdzie *X* to dane wejściowe, a *y* to dane wyjściowe. Następnie dzieli zbiory danych na zestawy treningowe i testowe, używając funkcji `train_test_split()` z biblioteki `scikit-learn`. Skrypt tworzy listę modeli uczenia maszynowego do użycia, w tym regresję liniową, *k*-najbliższych sąsiadów, regresję drzewa decyzyjnego i regresję wielowarstwowego perceptronu. Następnie trenuje każdy model, używając zestawu treningowego, i przewiduje wynik, używając zestawu testowego.



Rys. 5.4. Schemat działania skryptu

Skrypt używa dzielonej walidacji do oszacowania wydajności każdego modelu, a wynikowe wyniki są przechowywane na liście `self.scores`. W końcu skrypt oblicza rzeczywisty *CR* dla każdego przewidzianego wyniku, używając funkcji `calc_cr()` z pliku `helpers.py`, i przechowuje wyniki w plikach CSV dla każdego modelu (Rys. 5.3). Schemat działania algorytmu pokazano na Rys. 5.4.

5.3.2.1. Użyte biblioteki

W procesie realizacji projektu użyto szereg specjalistycznych bibliotek, które znacząco ułatwiły implementację i wykonanie różnych zadań związanych z analizą danych, uczeniem maszynowym oraz obliczeniami matematycznymi.

Użycie tych bibliotek pozwoliło na skoncentrowanie się na kluczowych aspektach badania, eliminując potrzebę pisania wielu rutynowych i skomplikowanych fragmentów kodu od podstaw.

Poniżej przedstawiono listę użytych bibliotek wraz z ich wersjami oraz krótkim opisem, jak zostały wykorzystane w kontekście projektu:

- **numpy (1.22.4)**: do obsługi tablic wielowymiarowych i macierzy, kluczowa w operacjach matematycznych.
- **pandas (1.4.2)**: niezbędna do manipulacji i analizy danych, w tym wczytywania i zapisywania plików CSV.
- **scikit-learn (1.1.1)**: wykorzystana do implementacji różnych modeli uczenia maszynowego oraz technik walidacji krzyżowej.
- **scipy (1.8.1)**: używana do zaawansowanych obliczeń naukowych, takich jak obliczenia wartości własnych macierzy.
- **joblib (1.1.0)**, **mpmath (1.2.1)**, **python-dateutil (2.8.2)**, **pytz (2022.1)**, **six (1.16.0)**, **sympy (1.11.1)**, **threadpoolctl (3.1.0)**: pozostałe biblioteki, które są wykorzystywane w różnych częściach projektu do różnorodnych zadań, takich jak serializacja obiektów, obliczenia symboliczne czy obsługa dat i czasu.

W kolejnym podrozdziale zostaną przedstawione szczegółowe fragmenty kodu, w których te biblioteki zostały użyte, oraz wyjaśnione, jak przyczyniły się do realizacji konkretnych zadań w projekcie.

5.3.2.2. Analiza kodu

Projekt składa się z kilku kluczowych plików, które współpracują ze sobą, tworząc kompleksowy system analizy i przetwarzania macierzy. Plik `main.py` zawiera główną logikę, odpowiedzialną za wczytywanie danych, tworzenie, trenowanie i testowanie różnych

modeli uczenia maszynowego oraz obliczanie rzeczywistych wartości CR . Plik `helpers.py` dostarcza funkcję pomocniczą do obliczenia wartości CR dla danej macierzy. Plik `FileLoader.py` jest odpowiedzialny za wczytywanie danych z plików, wykorzystując bibliotekę `pandas` do odczytu plików CSV. Plik `CR_calc.py` zawiera klasę `CrCalc`, która wykonuje obliczenia na macierzach, takie jak wyznaczanie wartości własnych i obliczanie współczynnika zgodności (CR).

Wszystkie te pliki współpracują ze sobą, tworząc zintegrowany system, który wykorzystuje różne techniki uczenia maszynowego i obliczeń matematycznych do analizy macierzy porównawczych. Można zatem przedstawić szczegółowo działanie każdej z nich:

- **CR_calc.py** - jest kluczowym elementem projektu, zawierającym klasę `CrCalc`, która wykonuje różne obliczenia na macierzach porównawczych. Niektóre funkcje zawarte w tej klasie zostały zapożyczone z biblioteki `REDUCE`, opisanej w rozdziale trzecim tej rozprawy doktorskiej. Oto szczegółowy opis funkcji i metod zawartych w tej klasie (zaprezentowanej w całości na Listingu 5.1.):
 - **__init__(self, m: np.matrix)**: Konstruktor klasy, który przyjmuje macierz jako argument. Sprawdza, czy macierz jest kwadratowa (tj. ma taki sam wymiar wierszy i kolumn), a jeśli nie, zgłasza wyjątek `WrongMatrix Shape`. Inicjalizuje również atrybuty klasy, takie jak kształt macierzy i jej rozmiar.
 - **max_eigenvalue(self)**: Metoda ta oblicza i zwraca największą wartość własną macierzy. Wykorzystuje do tego funkcję `eig` z biblioteki `scipy.linalg`, która zwraca wartości i wektory własne macierzy.
 - **__select_ri(size)**: Metoda statyczna, która zwraca wartość RI na podstawie rozmiaru macierzy. Wartości RI są zdefiniowane dla różnych rozmiarów macierzy i są używane do obliczenia współczynnika zgodności (CR).
 - **calc(self)**: Główna metoda klasy, która oblicza współczynnik zgodności (CR) dla danej macierzy. Wykorzystuje wcześniej zdefiniowane metody do obliczenia wartości własnej i wartości RI , a następnie oblicza CI (Consistency Index) i CR według określonych wzorów (1.20, 1.21).

- **FileLoader.py** - zawiera definicję klasy `Reader`, która służy do wczytywania danych z plików CSV. Jest to prosty, ale kluczowy komponent w procesie przetwarzania danych, umożliwiającą łatwe i wydajne wczytywanie danych do analizy. Klasa `Reader` zawiera jedną statyczną metodę, która służy do wczytywania danych z plików CSV. Plik widoczny jest na Listingu 5.3.
 - **load(file, sep=';')**: Metoda ta wykorzystuje funkcję `read_csv` z biblioteki `pandas` do wczytania pliku CSV i zwrócenia go jako obiekt `DataFrame`. Obiekt `DataFrame` jest powszechnie używaną strukturą danych w `pandas`, która umożliwia łatwe manipulowanie i analizę danych w formie tabelarycznej. Zwraca obiekt `DataFrame`, który zawiera dane wczytane z określonego pliku CSV.
- **helpers.py** - zawiera definicję funkcji pomocniczej `calc_cr`, która jest kluczowym elementem w procesie obliczania współczynnika spójności (*CR*) dla danej macierzy. Oto szczegółowy opis tej funkcji (przedstawionej na Listingu 5.3):
 - **data**: parametr wejściowy, obiekt, który zawiera wartości elementów macierzy 4×4 . Wartości te są odczytywane z kluczy takich jak `A11`, `A12` itd., które odpowiadają poszczególnym elementom macierzy.
 - Następnie następuje zasadnicza funkcja:
 - Funkcja najpierw tworzy macierz 4×4 (*m*), korzystając z wartości zawartych w parametrze `data`. Macierz ta jest reprezentowana jako obiekt `np.matrix` z biblioteki `numpy`.
 - Następnie funkcja wywołuje metodę `max_eigenvalue()`, aby obliczyć największą wartość własną macierzy. Jest to kluczowy krok w obliczaniu współczynnika spójności.
 - Wartość *CI* jest obliczana jako $(\text{max_vals} - \text{self.SIZE}) / (\text{self.SIZE} - 1)$, gdzie `max_vals` to największa wartość własna, a `self.SIZE` to rozmiar macierzy (w tym przypadku 4).
 - Funkcja wywołuje metodę `__select_ri(self.SIZE)`, aby wybrać odpowiednią wartość *RI* na podstawie rozmiaru macierzy. Wartości *RI* są predefiniowane dla różnych rozmiarów macierzy.
 - Współczynnik spójności *CR* jest obliczany jako ci / ri , gdzie *ci* to wartość *CI*, a *ri* to wybrana wartość *RI*.

- **main.py** - jest kluczowym skryptem, który koordynuje cały proces analizy danych, uczenia maszynowego i obliczeń związanych z macierzami. Oto szczegółowy opis tego, co robi ten plik:
 - Skrypt importuje biblioteki (Listing 5.4) takie jak `pandas`, `numpy` i różne modele z `scikit-learn`, które są niezbędne do manipulacji danymi i modelowania. Importuje również własne moduły, takie jak `FileLoader`, `CrCalc` i `calc_cr`, które są odpowiedzialne za wczytywanie danych, obliczanie współczynnika spójności (CR) i inne zadania pomocnicze.
 - **`__init__`** - metoda inicjalizacyjna klasy `Calc` (Listing 5.5) wykonuje kilka kluczowych zadań:
 - Wykorzystuje klasę `FileLoader.Reader` do wczytania danych wejściowych i wyjściowych z plików CSV.
 - Dzieli dane na zestawy treningowe i testowe za pomocą funkcji `train_test_split`.
 - Tworzy puste listy na wyniki i predykcje.
 - Wywołuje metody do tworzenia listy klasyfikatorów, dopasowywania ich i obliczania rzeczywistego CR.
 - **`create_classifiers_list`** - Ta metoda jest odpowiedzialna za tworzenie listy instancji klasyfikatorów, które mają zostać użyte w procesie uczenia maszynowego. Lista zawiera różne modele regresji, które są używane do przewidywania wartości ciągłych na podstawie danych wejściowych. W tym przypadku lista zawiera `Linear Regression`, `KNeighborsRegressor`, `DecisionTreeRegressor` i `MLPRegressor`.
 - **`fit_classifiers_list`** - metoda ta wykonuje kilka kluczowych zadań związanych z procesem uczenia maszynowego:
 - Dla każdego klasyfikatora w liście model jest dopasowywany do danych treningowych. Oznacza to, że algorytm uczenia maszynowego próbuje znaleźć najlepsze parametry modelu, które minimalizują błąd na danych treningowych.
 - Po dopasowaniu modelu do danych treningowych, jest on używany do przewidywania wyników dla danych testowych. Jest to kluczowy krok w ocenie, jak dobrze model generalizuje na nowych, niewidzianych danych.

- Predykcje są zapisywane w słowniku `self.predictions`, gdzie kluczem jest nazwa klasyfikatora, a wartością są przewidywane wyniki.
- `calc_real_cr` - ta metoda wykonuje obliczenia związane z macierzami:
 - Do każdej predykowanej macierzy dodaje kolumnę, w której wylicza rzeczywiste *CR*. Wyliczenie *CR* jest wykonane za pomocą funkcji pomocniczej `calc_cr()`, która wykonuje odpowiednie obliczenia na macierzy.
 - Wyniki, w tym wyliczone wartości *CR*, są zapisywane w plikach CSV, aby można je było później analizować i porównywać.

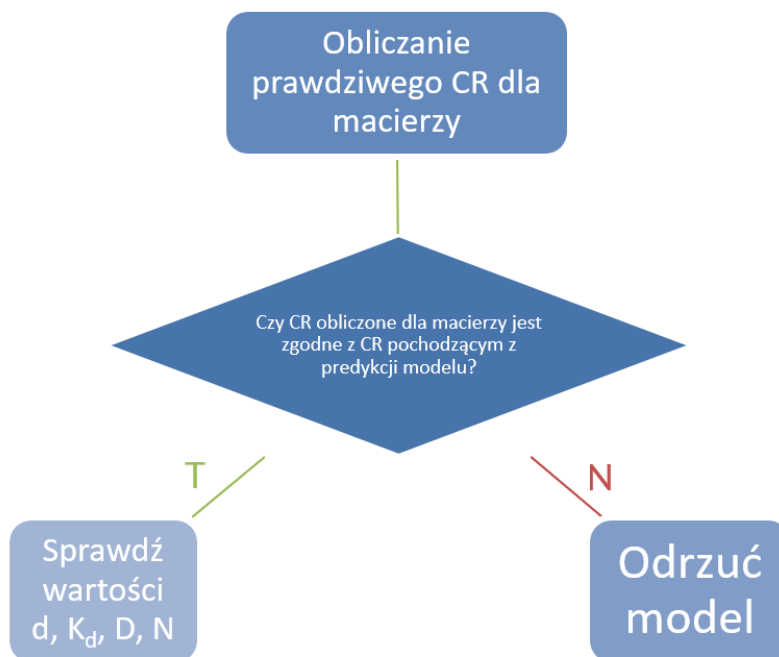
5.3.3. Kryteria oceny

Przyjęto, że pierwszym kryterium oceny będzie sprawdzenie wartości *MSE*, *MAE* oraz R^2 . Najlepsze wyniki osiągnięte przez algorytmy w tych wskaźnikach będą pierwszą sugestią dotyczącą wyboru algorytmu, który zostanie poddany szczegółowym ostatecznym pomiarom zbadającym jakość otrzymanych macierzy.

Drugim kryterium, zważywszy na czysto matematyczny charakter wskaźnika *CI* (oraz *CR*), oraz (jak widać na Rys. 5.3), wartość *CR* jest również zaangażowana w kształtowanie modelu uczenia maszynowego, fakt, że jest on poprawnie przewidywany, jest w tym przypadku kolejnym czynnikiem, który pozwala określić poprawność działania wybranego typu regresji na najbardziej podstawowym poziomie. Pozwoli to również na odrzucenie niektórych modeli do dalszej analizy, co uprości proces decyzyjny, ponieważ celem poniższego rozdziału jest analiza potencjału wykorzystania, zaś opracowanie konkretnych, skutecznych modeli mogłoby stanowić temat dla odrębnej pracy doktorskiej.

Następnie, na macierzach, które pozostaną wybrane, zostanie obliczona średnia zmiana wektora priorytetów, odległość tau Kendalla, odległość między dwiema macierzami oraz liczba zmienionych elementów w redukcji niespójności, które zostały opisane w rozdziale 2 niniejszej rozprawy.

Ostatnim krokiem będzie porównanie uzyskanych wyników z wynikami uzyskanymi przez iteracyjne "klasyczne" algorytmy redukcji niespójności macierzy porównań parami (wyniki pochodzące z rozdziału drugiego). Cały proces można zilustrować schematem przedstawionym na Rys. 5.4.



Rys. 5.4. Ostateczna ewaluacja modeli

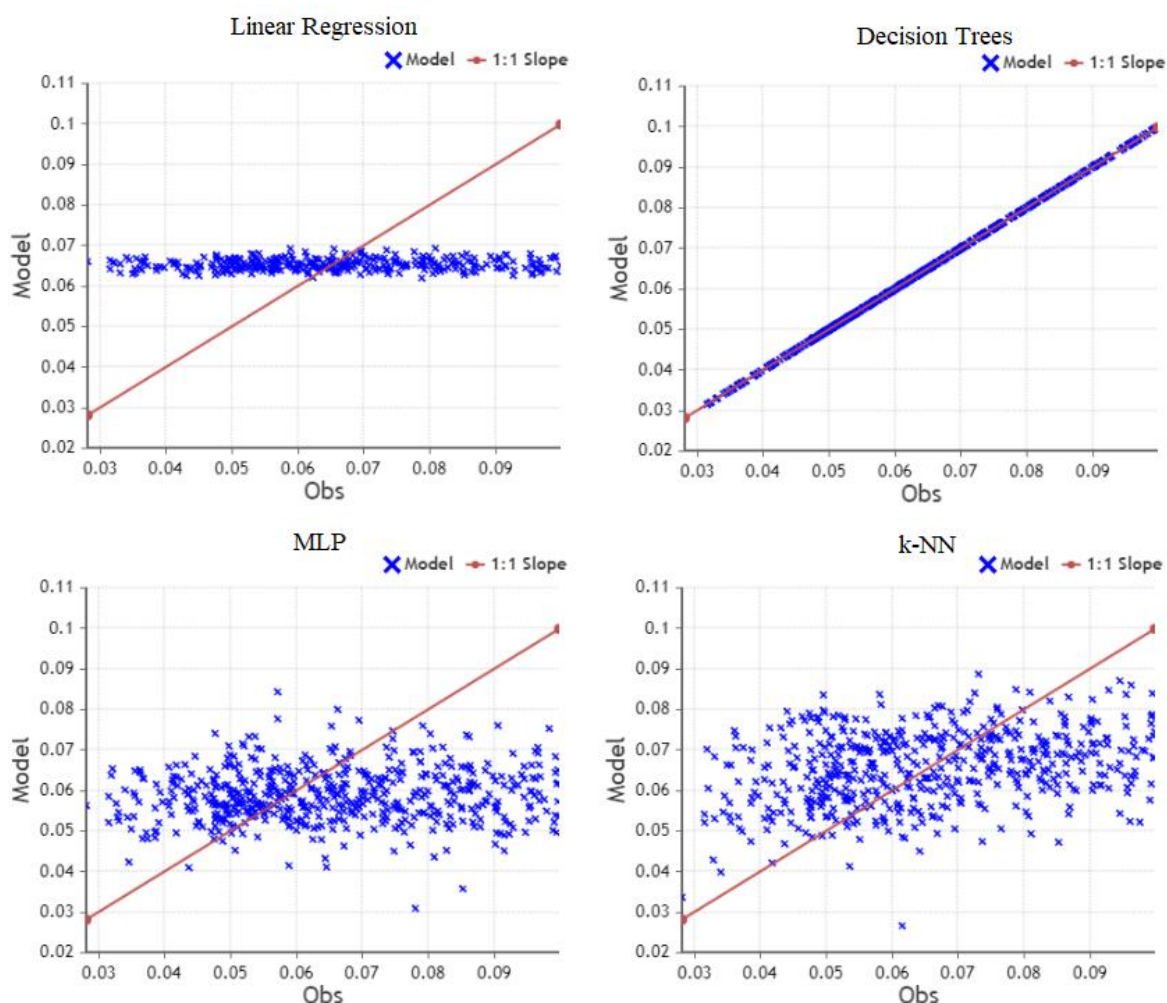
5.4. Wyniki i ich omówienie

W niniejszym podrozdziale przedstawione zostaną wyniki i analiza skuteczności proponowanego podejścia do redukcji niespójności w macierzach porównań parami. Szczególny nacisk położony zostanie na zrozumienie, dlaczego niektóre algorytmy okazały się skuteczniejsze od innych, oraz na zidentyfikowanie obszarów, w których proponowany model mógłby zostać dalej zoptymalizowany i zbadany.

Podrozdział ten ma na celu nie tylko przedstawienie surowych wyników, ale także zrozumienie ich znaczenia i kontekstu w dziedzinie redukcji niespójności w macierzach porównań parami. Przez połączenie analizy ilościowej z jakościową interpretacją, prowadzić będzie to do pełniejszego zrozumienia problemu i możliwości jego rozwiązania.

5.4.1. Wartości MSE , MAE oraz R^2

Zmierzono wartości MSE , MAE oraz R^2 (Rys.5.5) dla wyników CR uzyskanych przez metody wielokrotnej regresji liniowej, drzew decyzyjnych, algorytmu k-najbliższych sąsiadów oraz regresji perceptronu wielowarstwowego – dla danych testowych, które nie brały udziału w trenowaniu algorytmu. Wartości średnie dla wszystkich wskaźników umieszczono w Tabeli 5.3. Obliczeń dokonano za pomocą narzędzia [361].



Rys. 5.5. Dystrybucja wartości dla pomiarów błędów MSE , MAE oraz R^2 .

Tabela 5.3. Średnie wartości błędów MSE , MAE i R^2

	LR	DT	MLP	K-NN
MSE	0,00029	0	0,00029	0,00029
MAE	0,014	0	0,015	0,014
R^2	0,001	1	0,005	0,086

Analizując dane (Rys. 5.5) można dojść do następujących wniosków:

- **Drzewa decyzyjne (DT):** Wartości 0 dla MSE i MAE oraz 1 dla R^2 wskazują na idealne dopasowanie modelu do danych. Choć może to sugerować przetrenowanie, jest tu mowa o pomiarach dla danych testowych (a nie treningowych), więc w kontekście tego badania jest to interesujące, ponieważ oczekiwane jest badanie potencjału różnych metod, a niekoniecznie ich ostateczna skuteczność. Prawdopodobnie może to wynikać także z iteracyjnej oraz warunkowej natury algorytmu uczącego i podobnej do niego struktury drzew decyzyjnych.

- **Wielokrotna regresja liniowa (LR):** Wyniki dla regresji liniowej wskazują na brak znaczącego dopasowania, z wartościami skupionymi wokół wąskiego zakresu 0.06-0.07. To może sugerować, że model nie był w stanie nauczyć się złożonych wzorców w danych.
- **MLP i k-NN:** Wyniki dla perceptronu wielowarstwowego i metody k-najbliższych sąsiadów również nie są obiecujące, z wartościami, które wydają się być niewłaściwie obliczone. Jednak ważne jest, że użyto tutaj wyłącznie wartości domyślnych, które nadali tym metodom autorzy biblioteki sci-kit-learn, co oznacza, że te metody nie były dostosowane do konkretnego zadania i nigdy nie można wykluczyć, że mogą być brane pod uwagę.

Podsumowując, choć wyniki dla drzew decyzyjnych mogą wydawać się zbyt idealne, aby były prawdziwe, w kontekście tego badania są one najbardziej obiecujące, dlatego na podstawie tej części zostaną uwzględnione do pomiaru dalszych wskaźników (zostanie to jeszcze potwierdzone empirycznie w kolejnym podrozdziale). Pozostałe metody nie wykazały zdolności do skutecznego dopasowania się do wartości *CR*, co może wskazywać na konieczność dalszej optymalizacji i dostosowania do specyficznego zadania redukcji niespójności.

5.4.2. Zgodność obliczeń z prawdziwą wartością *CR* oraz dalsze wskaźniki

Kolejnym krokiem potwierdzającym było zbadanie, które algorytmy poprawnie nauczyły się przeprowadzać redukcję niespójności, o czym świadczy zgodność między wyuczonym *CR* a rzeczywistym. Jak widać na Rys. 5.6, niestety tylko jeden z algorytmów - regresja drzewa decyzyjnego - był w stanie poradzić sobie z tym testem, więc tylko on będzie brany pod uwagę w dalszych obliczeniach, co jest bezpośrednią pochodną poprzedniego podrozdziału.

Następnie, biorąc oryginalną, niezredukowaną macierz wraz z macierzą zredukowaną przez algorytm wykorzystujący regresję drzewa decyzyjnego, można policzyć wartości średniej zmiany wektora priorytetów, odległości tau Kendalla, odległości między dwiema macierzami oraz liczby zmienionych elementów w redukcji niespójności i wyznaczyć średnie wartości tych współczynników, co widać w Tabeli 5.4.

	A		B		C		D		E		F		G		H	
1	Decision Tree Regression		K-Neighbours Regression		Linear Regression		MLP Regression									
2	CR	cr_real	CR	cr_real	CR	cr_real	CR	cr_real	CR	cr_real	CR	cr_real	CR	cr_real	CR	cr_real
3	0,0593	0,0593	0,0593	0,06102	0,081721025	0,06668082481970666	-0,1815437522656399	0,05531259003746078	-0,6441869360562806	0,05644714728090024	0,10809365071506916	0,05644714728090024	0,10809365071506916	0,05644714728090024	0,10809365071506916	0,05644714728090024
4	0,0567	0,0567	0,0567	0,0658	0,083414797	0,06498310628713201	-0,3466652414701778	0,05869411736407466	0,2998302675352131	0,05869411736407466	0,2998302675352131	0,05869411736407466	0,2998302675352131	0,05869411736407466	0,2998302675352131	0,05869411736407466
5	0,0573	0,0573	0,0573	0,06338	0,125922859	0,06485094102104841	-0,4068496229021497	0,05318130745468309	0,3774515619750699	0,05318130745468309	0,3774515619750699	0,05318130745468309	0,3774515619750699	0,05318130745468309	0,3774515619750699	0,05318130745468309
6	0,0822	0,0822	0,0822	0,0733	0,150013554	0,06650971936339503	1,7733163372813425	0,05884960878718341	0,39846768947070915	0,05884960878718341	0,39846768947070915	0,05884960878718341	0,39846768947070915	0,05884960878718341	0,39846768947070915	0,05884960878718341
7	0,0840	0,0840	0,0840	0,0821	0,096944889	0,06536691996232516	-0,4659175667646214	0,0696566122432507	0,01749514266340363	0,0696566122432507	0,01749514266340363	0,0696566122432507	0,01749514266340363	0,0696566122432507	0,01749514266340363	0,0696566122432507
8	0,0640	0,0640	0,0640	0,05488	0,182202352	0,06446807537210893	0,3709091659464023	0,05192621005142665	0,2411207858175198	0,05192621005142665	0,2411207858175198	0,05192621005142665	0,2411207858175198	0,05192621005142665	0,2411207858175198	0,05192621005142665
9	0,0717	0,0717	0,0717	0,07076	0,100923195	0,06643511529577315	-0,2622324421477864	0,05687651286475145	-0,18801789501846342	0,05687651286475145	-0,18801789501846342	0,05687651286475145	-0,18801789501846342	0,05687651286475145	-0,18801789501846342	0,05687651286475145
10	0,0731	0,0731	0,0731	0,08876	0,269157551	0,06512591197201831	-0,22640845163799142	0,05256014903776385	0,39475787094573497	0,05256014903776385	0,39475787094573497	0,05256014903776385	0,39475787094573497	0,05256014903776385	0,39475787094573497	0,05256014903776385
11	0,0962	0,0962	0,0962	0,05242	0,243743367	0,0642571089673979	1,1750058375248462	0,06654492015461816	0,7987959026742121	0,06654492015461816	0,7987959026742121	0,06654492015461816	0,7987959026742121	0,06654492015461816	0,7987959026742121	0,06654492015461816
12	0,0789	0,0790	0,0790	0,06266	0,091078251	0,0665088340436724	-0,15511237204838307	0,06006381515895930	0,5795352277229816	0,06006381515895930	0,5795352277229816	0,06006381515895930	0,5795352277229816	0,06006381515895930	0,5795352277229816	0,06006381515895930
13	0,0634	0,0634	0,0634	0,05092	0,073636116	0,06460364231073536	-0,4895347548128841	0,04989692284575440	0,1143545682103039	0,04989692284575440	0,1143545682103039	0,04989692284575440	0,1143545682103039	0,04989692284575440	0,1143545682103039	0,04989692284575440
14	0,0689	0,0689	0,0689	0,07938	0,118899135	0,06579420557371973	-0,18273647232050333	0,0695654223297164	0,7216301519440642	0,0695654223297164	0,7216301519440642	0,0695654223297164	0,7216301519440642	0,0695654223297164	0,7216301519440642	0,0695654223297164
15	0,0665	0,0665	0,0665	0,0728	0,19883065	0,06616396771650308	-0,014059788311759252	0,0599059800447824	0,5541772669691913	0,0599059800447824	0,5541772669691913	0,0599059800447824	0,5541772669691913	0,0599059800447824	0,5541772669691913	0,0599059800447824
16	0,0632	0,0632	0,0632	0,05804	0,113480298	0,06660240818292772	-0,1538955467695304	0,05832734558550039	0,39938325407186753	0,05832734558550039	0,39938325407186753	0,05832734558550039	0,39938325407186753	0,05832734558550039	0,39938325407186753	0,05832734558550039
17	0,0779	0,0780	0,0780	0,06406	0,34270915	0,06518297211321623	-0,07465119307672279	0,0586999955459624	0,554045385312118	0,0586999955459624	0,554045385312118	0,0586999955459624	0,554045385312118	0,0586999955459624	0,554045385312118	0,0586999955459624
18	0,0606	0,0606	0,0606	0,06052	0,102749509	0,06563632284639982	-0,2260999295472858	0,05192621005142665	0,2411207858175198	0,05192621005142665	0,2411207858175198	0,05192621005142665	0,2411207858175198	0,05192621005142665	0,2411207858175198	0,05192621005142665
19	0,0547	0,0549	0,0549	0,05236	0,11563917	0,06484143161810733	-0,37432722402214474	0,05687651286475145	-0,18801789501846342	0,05687651286475145	-0,18801789501846342	0,05687651286475145	-0,18801789501846342	0,05687651286475145	-0,18801789501846342	0,05687651286475145
20	0,0657	0,0657	0,0657	0,07822	0,164282201	0,06321375850190067	0,44830024597323803	0,05256014903776385	0,39475787094573497	0,05256014903776385	0,39475787094573497	0,05256014903776385	0,39475787094573497	0,05256014903776385	0,39475787094573497	0,05256014903776385
21	0,0513	0,0513	0,0513	0,06624	0,181891312	0,06666214774714405	1,678606553498413	0,06654492015461816	0,7987959026742121	0,06654492015461816	0,7987959026742121	0,06654492015461816	0,7987959026742121	0,06654492015461816	0,7987959026742121	0,06654492015461816
22	0,0928	0,0928	0,0928	0,07138	0,14418268	0,0661493988855437	0,7531203119978424	0,06006381515895930	0,5795352277229816	0,06006381515895930	0,5795352277229816	0,06006381515895930	0,5795352277229816	0,06006381515895930	0,5795352277229816	0,06006381515895930
23	0,0482	0,0482	0,0482	0,06788	0,150423693	0,06522661537522256	-0,2536900846090026	0,04989692284575440	0,1143545682103039	0,04989692284575440	0,1143545682103039	0,04989692284575440	0,1143545682103039	0,04989692284575440	0,1143545682103039	0,04989692284575440
24	0,0778	0,0778	0,0778	0,06894	0,083098067	0,06849787444599985	0,36132883674587385	0,0695654223297164	0,7216301519440642	0,0695654223297164	0,7216301519440642	0,0695654223297164	0,7216301519440642	0,0695654223297164	0,7216301519440642	0,0695654223297164
25	0,0800	0,0801	0,0801	0,06784	0,099554128	0,06741970420636793	-0,18413801378810268	0,0599059800447824	0,5541772669691913	0,0599059800447824	0,5541772669691913	0,0599059800447824	0,5541772669691913	0,0599059800447824	0,5541772669691913	0,0599059800447824
26	0,0704	0,0704	0,0704	0,05724	0,094864616	0,06436433046733488	-0,48783571479342575	0,06481167459093655	0,4615159667562106	0,06481167459093655	0,4615159667562106	0,06481167459093655	0,4615159667562106	0,06481167459093655	0,4615159667562106	0,06481167459093655
27	0,0558	0,0558	0,0558	0,07394	0,145755823	0,06642909240112047	1,661315176012057	0,0565068352880550	-0,07671246624553281	0,0565068352880550	-0,07671246624553281	0,0565068352880550	-0,07671246624553281	0,0565068352880550	-0,07671246624553281	0,0565068352880550
28	0,0414	0,0414	0,0414	0,05356	0,063193113	0,06559907193172457	-0,30862680432573	0,06354175886849237	-0,8497009315418899	0,06354175886849237	-0,8497009315418899	0,06354175886849237	-0,8497009315418899	0,06354175886849237	-0,8497009315418899	0,06354175886849237

Rys. 5.6. Porównanie przewidywanej wartości CR z rzeczywistą wartością CR - pomarańczowe pola nie pasują, zielone są równe lub prawie równe.

Tabela 5.4. Średnie parametry zachowania preferencji po regresji drzewa decyzyjnego

Średnia zmiana wektora priorytetów (d)	Średnia odległość tau Kendalla (K_d)	Średnia odległość między dwiema macierzami (D)	Średnia liczba zmienionych elementów (N)
0.1946	0.3290	37.72	11.5

5.4.3. Porównanie proponowanego podejścia z istniejącymi metodami redukcji niespójności w macierzach porównań parami

W badaniu zaprezentowanym w rozdziale drugim rozprawy doktorskiej kilka iteracyjnych algorytmów redukcji niezgodności dla macierzy porównań parami zostało przetestowanych pod kątem miar zachowania preferencji w celu porównania ich skuteczności. Biorąc pod uwagę te badania, wyniki uzyskane w badaniu z niniejszego rozdziału zostały porównane z wynikami uzyskanymi we wspomnianym badaniu i pogrupowane w Tabeli 5.5.

Tabela 5.5. Porównanie wyników z iteracyjnymi metodami zmniejszania niespójności dla wartości początkowej $0.30 < CR < 0.80$. Zielone pola to najlepsza (najniższa) wartość, pomarańczowe - najgorsza (najwyższa).

	(d)	(K_d)	(D)	(N)
Algorytm drzew decyzyjnych	0.1946	0.3290	37.72	11.5
Cao et al. [5]	2.132	0.191	11.67	12
Cao et al. II [5]	1.552	0.147	14.44	12
Mazurek et al. [8]	5.341	0.595	8.74	4.62
Szybowski [6]	5.827	0.638	9.97	3.84
Xu and Wei [7]	0.567	0.039	14.47	12
Xu and Wei II [7]	0.723	0.049	12.12	12

5.4.4. Dyskusja na temat wyników

Model wykazuje duży potencjał - okazał się nawet najlepszy w zakresie jednego ze wskaźników - tj. średniej zmiany wektora priorytetów. Dodatkowo, na tle algorytmów iteracyjnych, wykazał się średnią odległością tau Kendalla na poziomie nieodbiegającym od innych oraz liczbą niezmiennych elementów bliską 12 (innymi słowy, zachowywane są tylko te na przekątnej) - co byłoby najgorszą wartością, ale jest też cechą innych algorytmów, np. Xu i Wei oraz Cao. Niestety, pod względem odległości między dwiema macierzami, wartość prezentowana przez model znacznie odbiega od innych wartości.

Co ciekawe, nie widać korelacji pomiędzy wynikami uzyskanymi za pomocą algorytmu Szybowskiego a modelem, na którym został on oparty. Daje to pole do dalszej analizy drzew decyzyjnych pod kątem redukcji niespójności, co z pewnością będzie przedmiotem przyszłych badań.

5.5. Wnioski

W rozdziale przedstawiono próbę użycia algorytmów uczenia maszynowego do rozwiązania problemu redukcji niespójności macierzy porównań parami. Na podstawie wyników i analizy można wysunąć następujące wnioski i obserwacje:

- Algorytmy drzew decyzyjnych uzyskały wyniki wskazujące na dalszy potencjał badawczy w tej dziedzinie. Nie tylko wykazały się zdolnością do redukcji niespójności, ale także wykazały pewne unikalne cechy, które mogą prowadzić do nowych odkryć i ulepszeń.
- Choć inne algorytmy uczenia maszynowego nie osiągnęły tak obiecujących wyników jak drzewa decyzyjne, nie należy ich definitywnie odrzucać. Mogą istnieć specyficzne ustawienia, dostosowania lub kombinacje, które mogą prowadzić do lepszych wyników.
- Wyniki te mogą zainicjować całkowicie nową gałąź badań na temat porównań parami i problemu redukcji niespójności. Wykorzystanie uczenia maszynowego do tego celu jest stosunkowo nowym podejściem i może prowadzić do innowacyjnych metod i technik.
- Dalsza analiza i eksperymenty są niezbędne, aby zrozumieć, dlaczego niektóre algorytmy, takie jak regresja liniowa, MLP i k-NN, nie osiągnęły oczekiwanych wyników. Zrozumienie tych ograniczeń może prowadzić do bardziej wyważonych i skutecznych metod redukcji niespójności.
- Porównanie z istniejącymi metodami redukcji niespójności, takimi jak te przedstawione przez Cao et al. i Xu i Wei, może prowadzić do synergii i integracji

różnych podejść, co może prowadzić do bardziej wszechstronnych i skutecznych rozwiązań.

- Ostatecznym celem jest nie tylko rozwój teoretyczny, ale także praktyczne zastosowanie tych metod w rzeczywistych systemach decyzyjnych i oceny. Dalsze badania powinny również skupić się na praktycznej implementacji i testowaniu w różnych kontekstach i zastosowaniach.

W świetle obecnych wyników, warto ponownie rozważyć niektóre techniki uczenia maszynowego, które mogły zostać wcześniej odrzucone ze względu na objętość badania, także te mniej konwencjonalne. Ponowne zbadanie tych technik może prowadzić do nowych odkryć i lepszego zrozumienia, jak różne algorytmy mogą być skutecznie wykorzystane w kontekście redukcji niespójności. W badaniu wykorzystano dane testowe generowane przez algorytm Szybowskiego. W przyszłości warto rozważyć pozyskanie danych testowych z różnorodnych źródeł, w tym rzeczywistych systemów decyzyjnych i oceny. Może to prowadzić do bardziej zróżnicowanego i reprezentatywnego zestawu danych, który lepiej odzwierciedla rzeczywiste wyzwania i potrzeby w dziedzinie redukcji niespójności. Różnorodność danych może również pomóc w zidentyfikowaniu unikalnych cech i wzorców, które mogą być kluczowe dla skutecznego wykorzystania algorytmów uczenia maszynowego w tym kontekście.

Podsumowując, wydają się istnieć obszary zastosowania algorytmów uczenia maszynowego w dziedzinie redukcji niespójności macierzy porównań parami. Wstępne wyniki są obiecujące, ale także wskazują na potrzebę dalszej analizy i eksperymentowanie, aby w pełni zrealizować potencjał tego podejścia.

6. Nowy algorytm generowania losowych macierzy porównań parami o zadanym zakresie współczynników niespójności - FAST-PCM oraz narzędzie online PC MATRICES GENERATOR do wydajnego generowania macierzy porównań parami

Słowa kluczowe rozdziału:

algorytm, porównanie parami, współczynnik niespójności, generowanie, nowe narzędzie online

Abstrakt rozdziału (za oryginalną publikacją):

Celem niniejszej publikacji jest przedstawienie nowej metody generowania losowych macierzy porównań parami z określonym przedziałem współczynnika niespójności (CR) przy użyciu algorytmów redukcji niespójności. Porównanie parami (PC) jest popularną techniką wielokryterialnego podejmowania decyzji, której celem jest przypisanie wag porównywanym podmiotom, a tym samym uszeregowanie ich od najlepszego do najgorszego. Przedstawiona metoda łączy tradycyjną losową generację macierzy porównań wspieraną przez algorytmy redukcji niespójności: algorytm "Xu i Wei" oraz algorytm "Szybowski". W artykule przedstawiono badania, które wykazały wzrost wydajności podczas generowania takich macierzy w stosunku do standardowej procedury losowego generowania macierzy porównań przy użyciu algorytmu "Szybowski". Inne algorytmy również poprawiają proces, ale w mniejszym stopniu, co sprawia, że algorytm wspomagający "Szybowski" jest preferowanym rozwiązaniem dla nowego procesu. W wyniku przeprowadzonych badań udostępniono również bezpłatne narzędzie online "PC MATRICES GENERATOR", które umożliwia wydajne generowanie dużej liczby macierzy porównań o zadanym zakresie współczynników CR, dowolnym rozmiarze macierzy i dowolnej liczbie macierzy, umożliwiając znacznie wydajniejsze i mniej czasochłonne badania w wielu dziedzinach wykorzystujących macierze porównań, takich jak analityczny proces hierarchiczny/sięciowy (AHP/ANP), ELECTREE, PAPRIKA, PROMETHE, VIKOR czy metoda Best-Worst (BWM).

6.1. Wstęp

Jak już wspomiano na łamach niniejszej pracy doktorskiej, ostatnie dziesięciolecie przyniosły rozwój wielu nowych wielokryterialnych metod podejmowania decyzji, głównie dzięki dogłębnym badaniom nad nimi. Główne z nich to techniki takie jak PROMETHEE, PAPRIKA, VIKOR, ELECTREE, a zwłaszcza BWM (Best-Worst Method) i AHP (Analytic Hierarchy Process, patrz np. Mazurek [1], Liang i in. [2], Saaty [3,4], Brans i in. [17,18], Hansen i in. [25], Opricovic i in. [26] i Alkiahari i in. [27]).

Wszystkie te metody stusują macierze porównań parami, które są używane również w wielu innych dziedzinach, patrz np. Koczkodaj i in. [20,21,22,23], Cavallo i in. [24]. Podczas realizacji badań, które je wykorzystują, zetknięto się również z problemem efektywnego generowania losowych macierzy porównań - niezbędnego np. do przeprowadzania symulacji Monte Carlo dla takich metod, które wymagają dużych ilości danych losowych, patrz np. Caflisch [5].

Problem ten był szczególnie widoczny w przypadku macierzy tzw. "dużych" (od 6x6 do 10x10) oraz gdy chciano jednocześnie uzyskać niski przedział CR - spójności (szczególnie poniżej wartości 0,1). Wówczas czas generacji kilku tysięcy macierzy losowych o zadanych parametrach mógł sięgać nawet kilku dni. Stanowiło to dużą przeszkodę w prowadzeniu efektywnych badań, gdyż konieczność przebadania dużej ilości danych z określonym przedziałem spójności może być potrzebna na takich frontach jak badanie konstruktywnych spójnych przybliżeń w porównaniach parami, patrz Smarzewski i in. [19]. Dlatego też, wykorzystując algorytmy redukcji niespójności, opracowano metodę, która znacznie usprawni ten proces i pomoże badaczom zaoszczędzić czas i zasoby sprzętowe.

Celem rozdziału jest przedstawienie i porównanie wyników pomiędzy normalną a ulepszoną metodą generowania losowych macierzy porównawczych o zadanych parametrach, a także wybór najlepszej metody redukcji niespójności, która wspiera algorytm generowania macierzy losowych oraz zaprezentowanie narzędzia on-line "PC MATRICES GENERATOR", które implementuje przedstawiony algorytm i jest udostępniane za darmo badaczom z całego świata, *(a także w przyszłości stanie się częścią pakietu REDUCE, opisanego w rozdziale 4, w ramach jego komercjalizacji – przyp. autorski)*.

6.2. Opis problemu i opracowanego algorytmu

W kontekście niniejszej pracy doktorskiej, przypomnijmy iż wskaźnik spójności CI (6.1) i wskaźnik spójności CR (6.2) zostały przedstawione przez Saaty'ego [3,4], gdzie

n w (6.1) jest rozmiarem macierzy porównań parami, a RI w (6.2) jest losowym wskaźnikiem spójności, określonym przez Saaty'ego (1980).

$$CI = \frac{\lambda_{max}}{n-1} \quad (6.1)$$

$$CR = \frac{CI}{RI} \quad (6.2)$$

Zwykle w obliczeniach dążymy do osiągnięcia wartości CR poniżej 0,10 - wtedy macierz jest uważana za spójną zgodnie z postulatami Saaty'ego [3]. W ciągu ostatnich lat opracowano wiele algorytmów redukujących taką niespójność i przedstawiono je w wielu pracach naukowych, oraz w pierwszym rozdziale niniejszej rozprawy doktorskiej patrz np. Xu i Wei [6], Cao i in. [7], Ergu i in. [8], Benítez i in. [9,10], Kułakowski i in. [11], Szybowski [12], Abel i in. [13], czy Mazurek i in. [14].

W opracowanej metodzie generowania macierzy zostaną wykorzystane algorytmy Szybowskiego [12] oraz algorytm Xu i Wei [6], ponieważ ich skuteczność została wykazana w niedawnej (2021) kompilacji różnych technik tego typu, patrz Mazurek i in. (również drugi rozdział tej rozprawy doktorskiej) [15], który również szczegółowo opisuje działanie tych algorytmów.

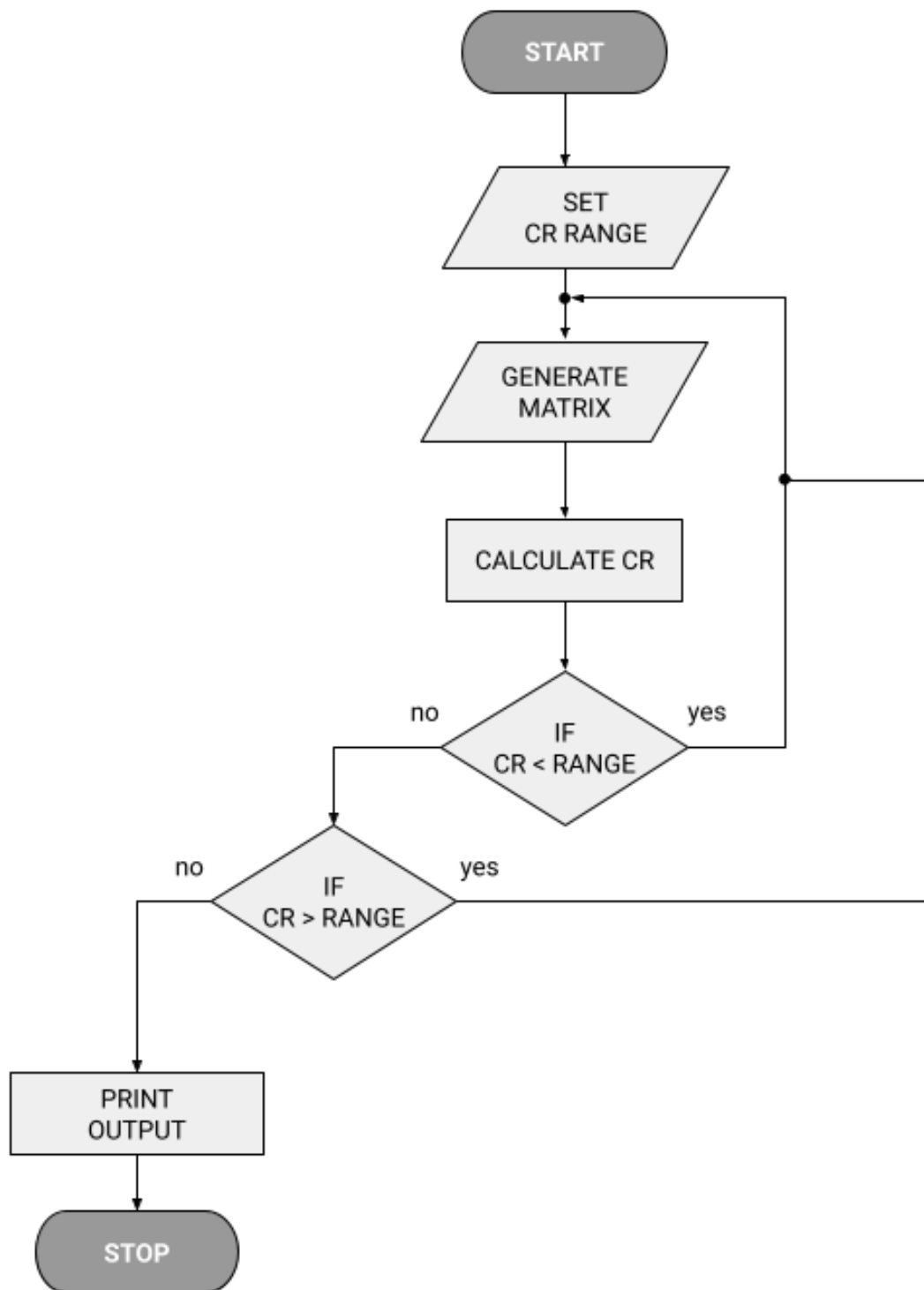
6.2.1. Standardowa metoda generowania macierzy porównań parami z żądanym współczynnikiem spójności

Schemat blokowy metody normalnej pokazano na Rys. 6.1 – rozpoczyna się od określenia zakresu współczynnika CR , następnie jest generowana losowa macierz porównań parami o zadanym rozmiarze (generując najpierw całą macierz pełną jedynek, uzupełniając ją losowymi wartościami od 1 do 9 na jednej przekątnej, a na drugiej - odwrotnościami tych wartości). Następnie jest obliczana wartość CR i jeśli mieści się ona w zadanym przedziale – jest dodawana do listy wygenerowanych macierzy, jeśli nie – generowane są kolejne macierze porównań parami, aż do uzyskania pożądanego wyniku.

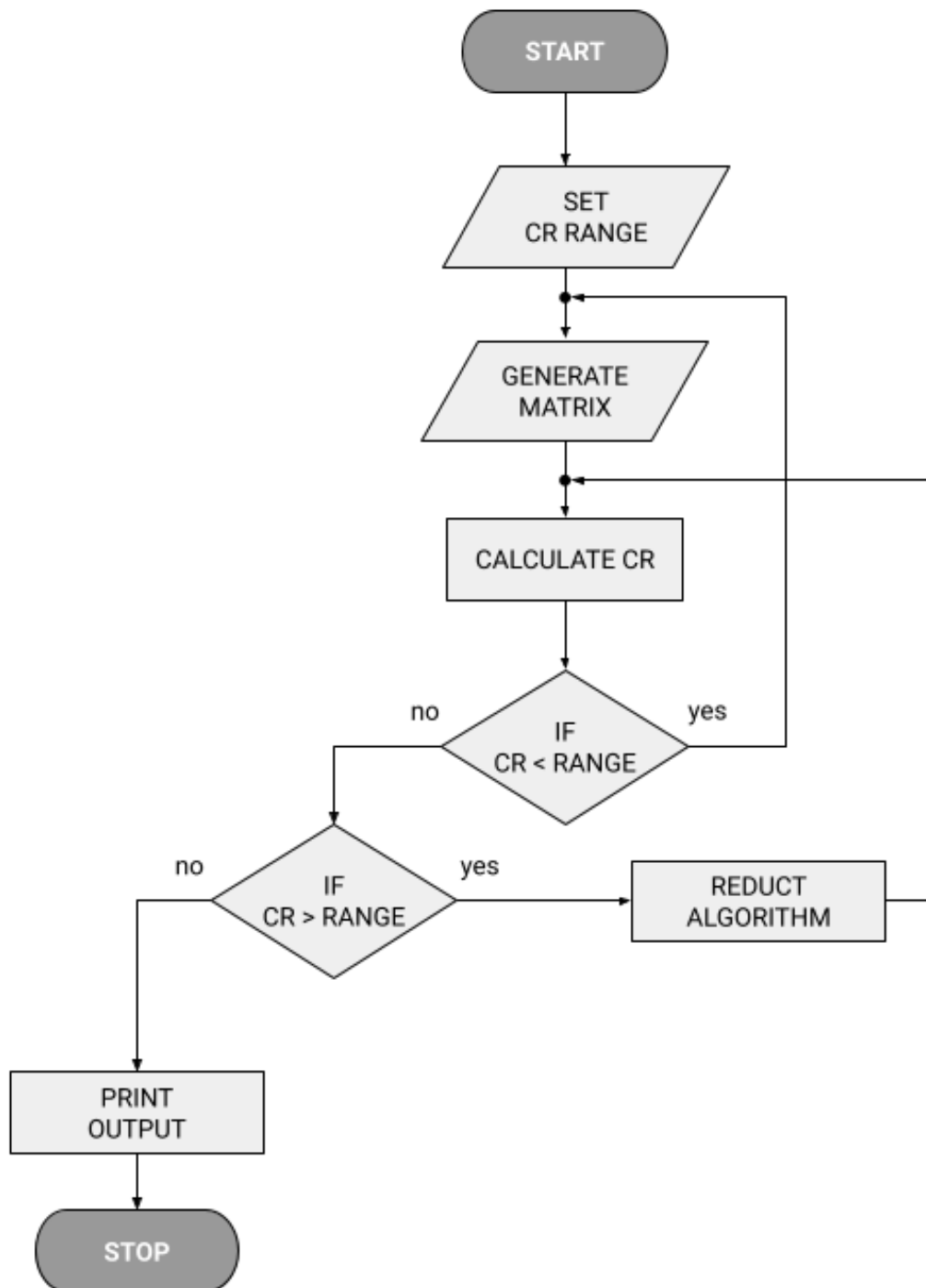
6.2.2. Ulepszona metoda generowania macierzy porównań parami z żądanym współczynnikiem spójności

Schemat blokowy ulepszonej metody pokazany jest na Rys.2 - w stosunku do zwykłej metody dodano punkt związany z redukcją niespójności, ale tylko w przypadku, gdy losowo wygenerowana macierz jest większa niż zadany zakres - w tej chwili nie ma algorytmów zwiększających niespójność macierzy porównawczej (można by było to rozważyć jako ciekawy przypadek badawczy, tym niemniej posiadać to będzie bliskie zeru zastosowanie praktyczne), więc należy odrzucić takie przypadki i wygenerować kolejną losową macierz.

Macierz jest zmniejszana przez algorytm do podanego zakresu i wtedy znajduje się już w podanym zakresie, więc może trafić bezpośrednio na wyjście algorytmu.



Rys. 6.1. Schemat blokowy standardowego algorytmu generowania losowych macierzy porównań z danym CR



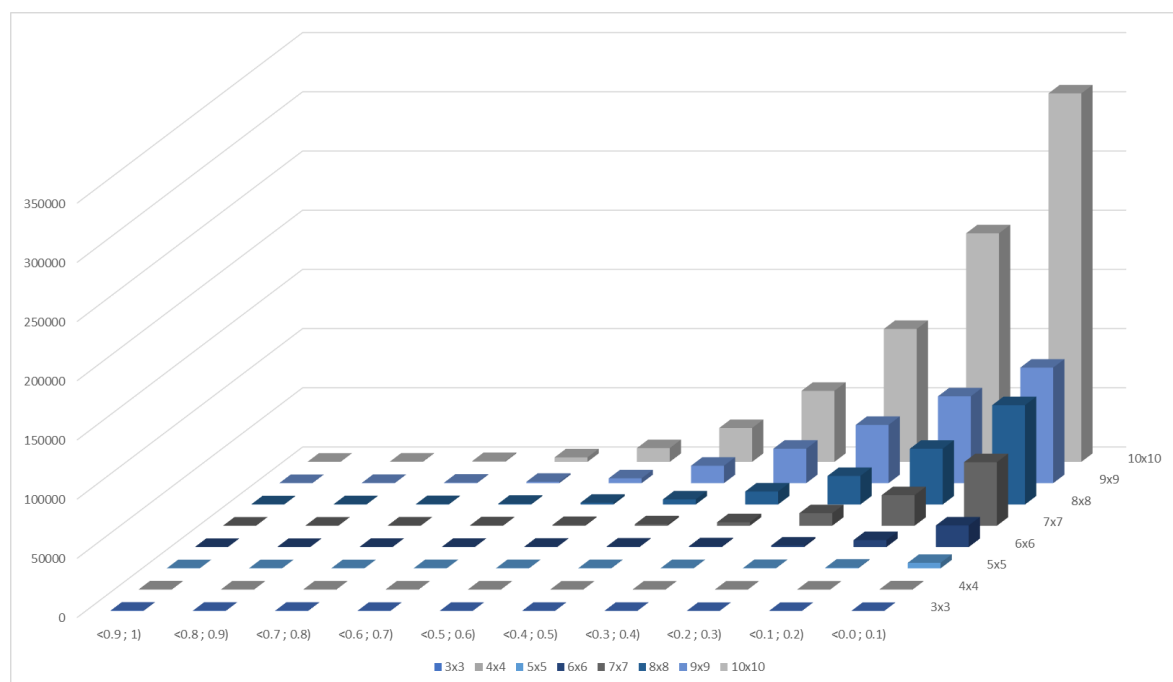
Rys. 6.2. Schemat blokowy ulepszony algorytmu generowania losowych macierzy porównań z danym CR

6.3. Wyniki i ich omówienie

6.3.1. Metodyka badań

Pomiary zostały zaimplementowane z wykorzystaniem środowiska Google Colab, w którym obie metody zostały zaimplementowane z wykorzystaniem środowiska Python 3.10.4 wykorzystującego - poza standardowymi bibliotekami dołączonymi do niego

- biblioteki NumPy 1.19.5, SciPy 1.6.0 i SymPy 1.7.1 (składowe biblioteki REDUCE opisanej w rozdziale 3, sformalizowanej po powstaniu tych badań), natomiast pomiary zostały zaimplementowane z wykorzystaniem biblioteki czasu języka Python.



Rys. 6.3. Czas generowania 10 000 losowych macierzy o różnych rozmiarach przy użyciu zwykłej metody w różnych zakresach CR (w sekundach)

Tabela 6.1. Czas generacji 10 000 macierzy losowych o różnych kształtach przy użyciu metody zwykłej w różnych zakresach CR

[s]	3x3	4x4	5x5	6x6	7x7	8x8	9x9	10x10
<0.9 ; 1)	272	204	146	132	151	156	157	154
<0.8 ; 0.9)	200	195	154	148	174	205	227	274
<0.7 ; 0.8)	211	182	167	180	234	337	456	688
<0.6 ; 0.7)	113	162	180	224	366	649	1198	3809
<0.5 ; 0.6)	101	139	190	296	643	1479	4109	11641
<0.4 ; 0.5)	73	117	200	415	1263	4292	14739	28617
<0.3 ; 0.4)	68	97	212	617	2954	10941	29110	60009
<0.2 ; 0.3)	50	95	273	1279	10632	24092	49217	112277
<0.1 ; 0.2)	49	112	584	5870	25837	47059	73377	193067
<0.0 ; 0.1)	25	246	4632	18245	53673	83849	97602	311210

Podane w tabelach czasy (w sekundach) są wynikiem uśrednienia 10 niezależnych pomiarów wykonanych na tym samym środowisku, z każdorazowym czyszczeniem zawartości pamięci cache i RAM środowiska.

6.3.2. Wyniki dla metody standardowej losowego generowania macierzy porównań parami

Wyniki pomiarów zostały przedstawione w Tabeli 6.1 i zilustrowane na wykresie (Rys. 6.3). Widać wyraźnie, że podczas gdy dla małych rozmiarów macierzy czasy są akceptowalne, wraz ze wzrostem rozmiaru macierzy i spadkiem CR , czasy osiągają ponad 3 dni dla największego rozmiaru.

6.3.3. Wyniki dla ulepszonej metody losowego generowania macierzy porównań parami

Dla ulepszonej metody przeprowadzono oddzielne pomiary zarówno dla algorytmu Xu i Wei [6], jak i algorytmu Szybowskiego [12]. W obu przypadkach (Tab. 6.2, Tab. 6.3) widać wyraźną poprawę i skrócenie o rząd wielkości czasu generowanych macierzy, ale zdecydowanym zwycięzcą okazał się algorytm Szybowskiego, co widać zarówno biorąc pod uwagę porównania "zwykłego" algorytmu z Xu i Wei (Rys. 6.4), jak i z algorytmem Szybowskiego (Rys. 6.5). Nie był to jednak "zwycięzca" każdego przedziału, ale w większości przypadków osiągnął najlepsze wyniki - pokazuje to Tab. 6.4.

Tabela 6.2 Czas generacji 10 000 macierzy losowych o różnych kształtach przy użyciu ulepszonego algorytmu z metodą redukcji Xu i Wei w różnych zakresach CR

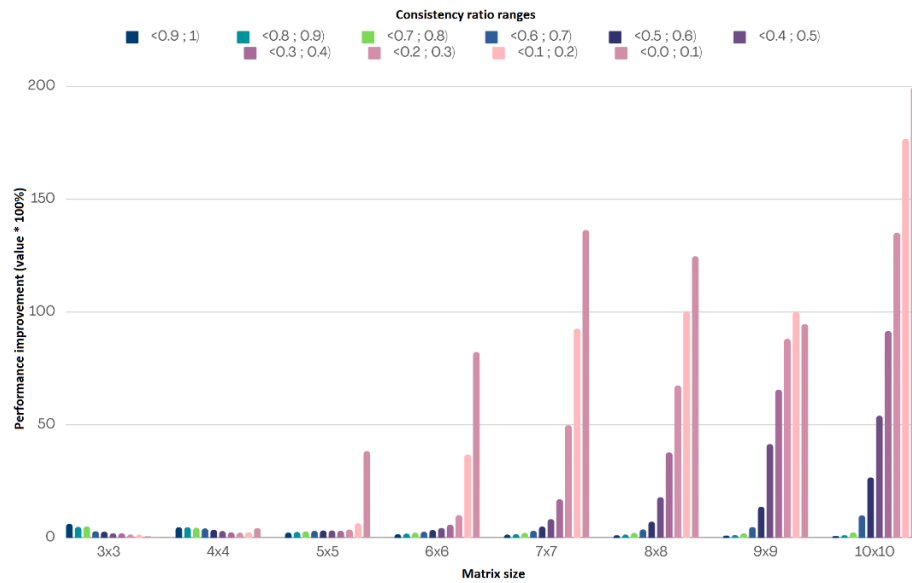
[s]	3x3	4x4	5x5	6x6	7x7	8x8	9x9	10x10
<0.9 ; 1)	148	132	129	133	104	105	117	133
<0.8 ; 0.9)	149	147	147	156	118	132	150	170
<0.7 ; 0.8)	158	161	170	191	150	179	201	257
<0.6 ; 0.7)	159	179	199	239	197	247	307	346
<0.5 ; 0.6)	162	197	237	299	263	338	423	533
<0.4 ; 0.5)	168	220	286	378	344	453	575	720
<0.3 ; 0.4)	172	248	347	489	461	607	774	951
<0.2 ; 0.3)	191	289	436	642	612	808	1049	1257
<0.1 ; 0.2)	219	365	593	872	834	1093	1409	1729
<0.0 ; 0.1)	259	533	876	1283	1207	1620	2071	2940

Tabela 6.3 Czas generacji 10 000 macierzy losowych o różnych kształtach przy użyciu ulepszonego algorytmu z metodą redukcji Szybowskiego w różnych zakresach CR

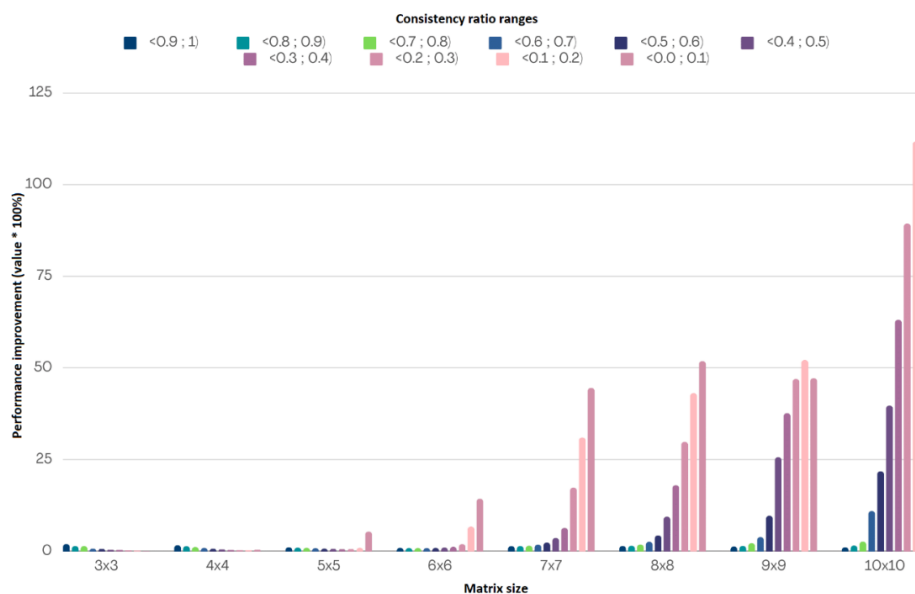
[s]	3x3	4x4	5x5	6x6	7x7	8x8	9x9	10x10
<0.9 ; 1)	45	45	61	76	99	126	150	175
<0.8 ; 0.9)	42	43	58	75	99	134	166	213
<0.7 ; 0.8)	43	42	57	76	103	148	216	273
<0.6 ; 0.7)	39	40	57	81	114	171	246	379
<0.5 ; 0.6)	37	40	57	84	127	204	298	434
<0.4 ; 0.5)	36	40	60	93	151	238	355	529
<0.3 ; 0.4)	33	40	66	105	172	289	444	655
<0.2 ; 0.3)	33	42	73	126	213	357	559	831
<0.1 ; 0.2)	34	46	90	159	279	469	733	1093
<0.0 ; 0.1)	34	58	121	222	394	673	1032	1560

Tabela 6.4 Najlepsza metoda dla każdego rozmiaru macierzy i zakresu CR stosowanego w opcji mieszanej (zalecanej) w narzędziu dostarczonym w rozdziale

[s]	3x3	4x4	5x5	6x6	7x7	8x8	9x9	10x10
<0.9 ; 1)	Sz.	Sz.	Sz.	Sz.	Sz.	Xu.	Xu.	Xu.
<0.8 ; 0.9)	Sz.	Sz.	Sz.	Sz.	Sz.	Xu.	Xu.	Xu.
<0.7 ; 0.8)	Sz.	Sz.	Sz.	Sz.	Sz.	Sz.	Xu.	Xu.
<0.6 ; 0.7)	Sz.	Sz.	Sz.	Sz.	Sz.	Sz.	Sz.	Xu.
<0.5 ; 0.6)	Sz.	Sz.	Sz.	Sz.	Sz.	Sz.	Sz.	Sz.
<0.4 ; 0.5)	Sz.	Sz.	Sz.	Sz.	Sz.	Sz.	Sz.	Sz.
<0.3 ; 0.4)	Sz.	Sz.	Sz.	Sz.	Sz.	Sz.	Sz.	Sz.
<0.2 ; 0.3)	Sz.	Sz.	Sz.	Sz.	Sz.	Sz.	Sz.	Sz.
<0.1 ; 0.2)	Sz.	Sz.	Sz.	Sz.	Sz.	Sz.	Sz.	Sz.
<0.0 ; 0.1)	Normal	Sz.	Sz.	Sz.	Sz.	Sz.	Sz.	Sz.



Rys.6.4 Porównanie wydajności ulepszonego algorytmu "Xu i Wei" ze zwykłym algorytmem - wartość na wykresie wskazuje wartość procentowej poprawy wydajności (wartość * 100%).



Rys. 6.5. Porównanie wydajności ulepszonego algorytmu "Szybowski" ze standardowym algorytmem - wartość na wykresie wskazuje wartość procentowej poprawy wydajności (wartość * 100%).

6.4. "PC MATRICES GENERATOR" – implementacja proponowanej nowej metody

Usługa online do generowania losowych macierzy porównań z danym współczynnikiem CR została uruchomiona pod adresem URL [16]. Jej interfejs pokazano na Rys. 6.6. i został on opracowany przy użyciu frameworka Flask - serwującego skrypty Pythona do środowiska aplikacji internetowej. Wykorzystuje on te same biblioteki, co środowisko używane do testowania czasów generowania macierzy (czyli między innymi bibliotekę Reduce.py opisaną w rozdziale 3), podczas gdy strona użytkownika wykorzystuje standardowe technologie - HTML, CSS i JavaScript (technologicznie wykorzystywane są tu te same mechanizmy, co w narzędziu REDUCE, ponieważ planowana jest integracja tego narzędzia do docelowego, komercyjnego pakietu REDUCE w nieodległej przyszłości).

PC MATRICES GENERATOR

A simple online tool for generating pairwise comparison matrices

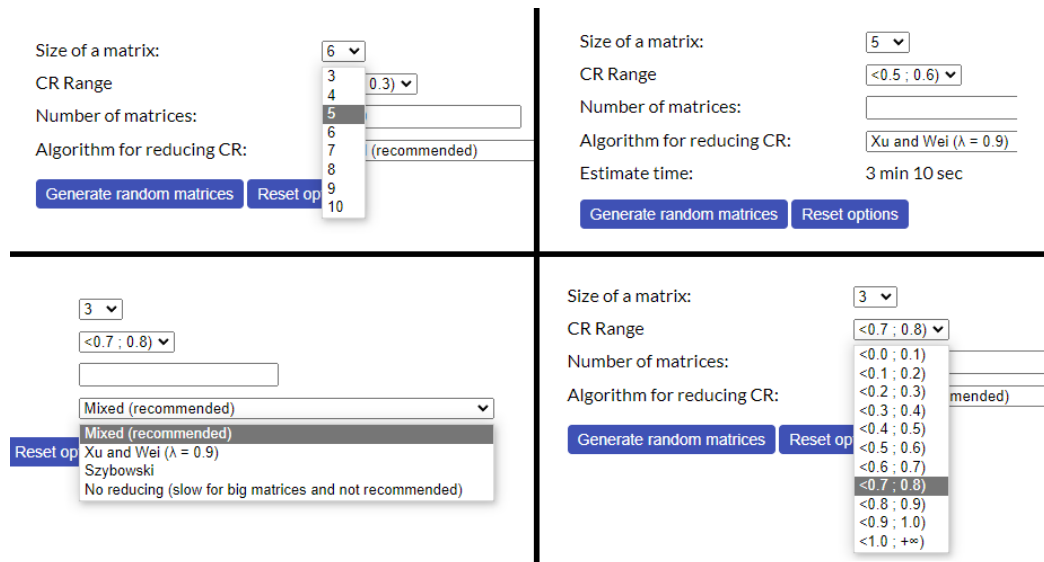
[How to use it?](#)

Size of a matrix:	<input type="text" value="6"/>
CR Range	<input type="text" value="<0.2 ; 0.3"/>
Number of matrices:	<input type="text" value="10000"/>
Algorithm for reducing CR:	<input type="text" value="Mixed (recommended)"/>

Rys. 6.6 Interfejs aplikacji "PC MATRICES GENERATOR"

Użytkownik może określić rozmiar macierzy, ustawić żądany interwał CR i zdefiniować liczbę macierzy oraz wybrać żądany algorytm (Rys. 6.7). Zalecana opcja - mieszana - została zaczerpnięta z Tab. 6.4 z tego artykułu i używa różne algorytmy w celu uzyskania maksymalnej wydajności. Po ustawieniu tych wartości program zwróci oczekiwany czas generowania macierzy i można użyć przycisku "Generate random matrices", aby uruchomić skrypt generujący.

Po zakończeniu przeglądarka użytkownika rozpocznie pobieranie pliku w formacie CSV zawierającego gotowe do użycia macierze wraz z wektorami priorytetów i wartościami CR .



Rys. 6.7 Dostępne opcje w aplikacji "PC MATRICES GENERATOR"

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	A11	A12	A13	A14	A21	A22	A23	A24	A31	A32	A33	A34	A41	A42	A43	A44	W1	W2	W3	W4	CR
2	1.0	0.1667	2.0	1.0	6.0	1.0	9.0	6.0	0.5	0.1111	1.0	0.5	1.0	0.1667	2.0	1.0	0.1229	0.6879	0.0662	0.1229	0.0039
3	1.0	0.125	0.1667	0.5	8.0	1.0	1.0	3.0	6.0	1.0	1.0	3.0	2.0	0.3333	0.3333	1.0	0.0606	0.4199	0.3897	0.1299	0.0039
4	1.0	6.0	6.0	1.0	0.1667	1.0	1.0	0.125	0.1667	1.0	1.0	0.125	1.0	8.0	8.0	1.0	0.4059	0.0628	0.0628	0.4685	0.0039
5	1.0	1.0	9.0	7.0	1.0	1.0	7.0	6.0	0.1111	0.1429	1.0	1.0	0.1429	0.1667	1.0	1.0	0.4613	0.4163	0.0581	0.0642	0.0041
6	1.0	0.1429	1.0	0.1429	7.0	1.0	5.0	1.0	1.0	0.2	1.0	0.1429	7.0	1.0	7.0	1.0	0.0643	0.4152	0.0702	0.4502	0.0054
7	1.0	8.0	3.0	3.0	0.125	1.0	0.25	0.3333	0.3333	4.0	1.0	1.0	0.3333	3.0	1.0	1.0	0.5488	0.0603	0.203	0.1879	0.0062
8	1.0	0.125	1.0	0.25	8.0	1.0	8.0	3.0	1.0	0.125	1.0	0.3333	4.0	0.3333	3.0	1.0	0.0701	0.6191	0.0749	0.2359	0.0062
9	1.0	2.0	2.0	9.0	0.5	1.0	1.0	4.0	0.5	1.0	1.0	6.0	0.1111	0.25	0.1667	1.0	0.4687	0.2281	0.2531	0.0502	0.0062
10	1.0	1.0	1.0	0.5	1.0	1.0	1.0	0.5	1.0	1.0	0.3333	2.0	2.0	3.0	1.0	1.0	0.1948	0.1948	0.1768	0.4336	0.0078
11	1.0	0.5	0.3333	4.0	2.0	1.0	1.0	7.0	3.0	1.0	1.0	8.0	0.25	0.1429	0.125	1.0	0.1704	0.3633	0.4177	0.0485	0.0086
12	1.0	0.2	0.1429	1.0	5.0	1.0	1.0	8.0	7.0	1.0	1.0	9.0	1.0	0.125	0.1111	1.0	0.0678	0.414	0.4617	0.0565	0.0097
13	1.0	0.5	0.2	1.0	2.0	1.0	0.3333	3.0	5.0	3.0	1.0	8.0	1.0	0.3333	0.125	1.0	0.1024	0.2153	0.6	0.0823	0.0103
14	1.0	5.0	1.0	9.0	0.2	1.0	0.3333	2.0	1.0	3.0	1.0	6.0	0.1111	0.5	0.1667	1.0	0.4657	0.1082	0.3708	0.0554	0.0104
15	1.0	0.1429	1.0	0.3333	7.0	1.0	7.0	3.0	1.0	0.1429	1.0	0.25	3.0	0.3333	4.0	1.0	0.08	0.6006	0.075	0.2445	0.0104
16	1.0	0.3333	1.0	0.1429	3.0	1.0	3.0	0.25	1.0	0.3333	1.0	0.1111	7.0	4.0	9.0	1.0	0.0763	0.2001	0.0712	0.6524	0.0104
17	1.0	0.5	0.1429	0.125	2.0	1.0	0.2	0.3333	7.0	5.0	1.0	1.0	8.0	3.0	1.0	1.0	0.055	0.109	0.4382	0.3978	0.0116
18	1.0	1.0	4.0	6.0	1.0	1.0	3.0	4.0	0.25	0.3333	1.0	1.0	0.1667	0.25	1.0	1.0	0.4377	0.3671	0.1059	0.0893	0.0117
19	1.0	6.0	4.0	0.5	0.1667	1.0	1.0	0.125	0.25	1.0	1.0	0.1667	2.0	8.0	6.0	1.0	0.3202	0.0653	0.0775	0.537	0.0117
20	1.0	0.3333	0.1111	0.1111	3.0	1.0	0.2	0.3333	9.0	5.0	1.0	1.0	9.0	3.0	1.0	1.0	0.0435	0.1157	0.4491	0.3917	0.0124
21	1.0	0.5	0.1111	0.1111	2.0	1.0	0.25	0.1429	9.0	4.0	1.0	1.0	9.0	7.0	1.0	1.0	0.0459	0.0854	0.4026	0.4661	0.0124
22	1.0	7.0	9.0	1.0	0.1429	1.0	2.0	0.25	0.1111	0.5	1.0	0.125	1.0	4.0	8.0	1.0	0.471	0.0862	0.0478	0.395	0.0124
23	1.0	0.1111	0.1429	0.5	9.0	1.0	2.0	8.0	7.0	0.5	1.0	3.0	2.0	0.125	0.3333	1.0	0.0491	0.5696	0.2934	0.0879	0.015
24	1.0	0.1111	0.2	0.1667	9.0	1.0	2.0	1.0	5.0	0.5	1.0	1.0	6.0	1.0	1.0	1.0	0.0475	0.4018	0.2446	0.3061	0.0164
25	1.0	2.0	9.0	2.0	0.5	1.0	4.0	0.5	0.1111	0.25	1.0	0.1429	0.5	2.0	7.0	1.0	0.461	0.187	0.0465	0.3055	0.0168
26	1.0	0.125	0.3333	1.0	8.0	1.0	2.0	8.0	3.0	0.5	1.0	6.0	1.0	0.125	0.1667	1.0	0.0766	0.564	0.2949	0.0644	0.0173
27	1.0	8.0	2.0	4.0	0.125	1.0	0.25	1.0	0.5	4.0	1.0	3.0	0.25	1.0	0.3333	1.0	0.5301	0.0788	0.29	0.1011	0.0173
28	1.0	2.0	9.0	0.5	0.5	1.0	3.0	0.25	0.1111	0.3333	1.0	0.1111	2.0	4.0	9.0	1.0	0.3061	0.1365	0.0448	0.5126	0.0173
29	1.0	0.125	1.0	0.1667	8.0	1.0	6.0	2.0	1.0	0.1667	1.0	0.3333	6.0	0.5	3.0	1.0	0.0669	0.5433	0.085	0.3047	0.0173
30	1.0	0.1667	2.0	0.25	6.0	1.0	8.0	2.0	0.5	0.125	1.0	0.125	4.0	0.5	8.0	1.0	0.0891	0.5264	0.05	0.3346	0.0173
31	1.0	2.0	6.0	2.0	0.5	1.0	3.0	0.5	0.1667	0.3333	1.0	0.25	0.5	2.0	4.0	1.0	0.4513	0.1898	0.0692	0.2897	0.0173
32	1.0	4.0	0.3333	0.5	0.25	1.0	0.1667	0.1667	3.0	6.0	1.0	1.0	2.0	6.0	1.0	1.0	0.1777	0.0565	0.4048	0.3609	0.0173
33	1.0	0.5	6.0	2.0	2.0	1.0	8.0	4.0	0.1667	0.125	1.0	0.25	0.5	0.25	4.0	1.0	0.2821	0.5156	0.0492	0.1532	0.0173
34	1.0	6.0	1.0	0.5	0.1667	1.0	0.25	0.1667	1.0	4.0	1.0	0.5	2.0	6.0	2.0	1.0	0.2642	0.058	0.2355	0.4423	0.0173
35	1.0	0.5	0.5	0.3333	2.0	1.0	1.0	1.0	2.0	1.0	1.0	0.5	3.0	1.0	2.0	1.0	0.1238	0.2778	0.2326	0.3659	0.0173
36	1.0	0.5	6.0	2.0	2.0	1.0	6.0	3.0	0.1667	0.1667	1.0	0.5	0.5	0.3333	2.0	1.0	0.3073	0.4787	0.0671	0.1469	0.0173
37	1.0	6.0	6.0	4.0	0.1667	1.0	0.5	0.5	0.1667	2.0	1.0	1.0	0.25	2.0	1.0	1.0	0.63	0.0818	0.1376	0.1506	0.0173

Rys. 6.8 Przykładowe dane wyjściowe aplikacji "PC Matrices Generator" - plik CSV z losowymi macierzami w podanym zakresie CR

6.5. Wnioski

Celem niniejszego rozdziału było przedstawienie nowej metody generowania losowych macierzy porównań przy użyciu algorytmów redukcji niespójności oraz wykazanie znacznego wzrostu wydajności związanego z ich zastosowaniem. Zaprezentowano również narzędzie online implementujące te metody w praktyce. Zapropozowane metody i narzędzia pomogą przyspieszyć pracę badaczy zajmujących się macierzami porównań.

Podsumowanie rozprawy doktorskiej

Rozprawa doktorska skupia się na problemie redukcji niespójności w macierzach porównań parami, przedstawiając nowy algorytm FAST-PCM oraz różne narzędzia i metody, które wspierają ten proces.

Teza rozprawy doktorskiej koncentruje się na algorytmie FAST-PCM, który bazuje na znanych metodach redukcji niespójności w macierzach porównań parami. Jest to kluczowe dla poprawy wydajności generowania losowych macierzy porównań o zadanym zakresie współczynników niespójności. Rozprawa doktorska była skoncentrowana na osiągnięciu zarówno podstawowych, jak i dodatkowych celów, które obejmowały numeryczne porównanie algorytmów, badanie potencjału wykorzystania algorytmów uczenia maszynowego do redukcji niespójności macierzy, stworzenie założeń algorytmu FAST-PCM, oraz rozwój biblioteki REDUCE.py i ogólnodostępnych aplikacji.

Przegląd literatury stanowi fundament dla całej rozprawy, przedstawiając kluczowe koncepcje macierzy porównań parami, ich zastosowania i problem niespójności. To zrozumienie pozwoliło na zidentyfikowanie luk w istniejących metodach i wytyczyło drogę do dalszych badań.

W rozdziale drugim przeprowadzono szczegółowe badania nad iteracyjnymi algorytmami redukcji niespójności. Wyniki tych badań były kluczowe dla zrozumienia, jak różne algorytmy zachowują się w różnych sytuacjach, co prowadziło do wniosku, że nie ma jednego algorytmu, który byłby najlepszy pod każdym względem.

Rozdział 3 skupił się na stworzeniu biblioteki REDUCE.py, która była kluczowym narzędziem w procesie redukcji niespójności. Ta biblioteka nie tylko przyczyniła się do lepszego procesu podejmowania decyzji, ale także otworzyła możliwości dla małych i średnich przedsiębiorstw.

W rozdziale 4 przedstawiono narzędzie REDUCE, które umożliwiło automatyczną redukcję niespójności. Jest to ważny krok w kierunku demokratyzacji technologii podejmowania decyzji, umożliwiając dostęp do zaawansowanych narzędzi bez konieczności posiadania drogiego oprogramowania.

Rozdział 5 przedstawił innowacyjne podejście do problemu redukcji niespójności, wykorzystując techniki uczenia maszynowego. To badanie otworzyło nowe perspektywy w dziedzinie redukcji niespójności, łącząc tradycyjne metody z nowoczesnymi technologiami. Spełnił także postawiony cel podstawowy zbadania potencjału wykorzystania takich technik do redukcji, wykazując, że np. regresja drzew decyzyjnych wykazuje obiecujące wyniki.

W rozdziale 6 przedstawiono nowy algorytm FAST-PCM i narzędzie online, które znacznie zwiększyły wydajność generowania macierzy porównań. To osiągnięcie nie tylko potwierdziło tezę rozprawy, ale także przyczyniło się do postępu w wielu dziedzinach wykorzystujących porównania parami.

Rozprawa doktorska zintegrowała różne metody i narzędzia, aby osiągnąć cele i potwierdzić tezę. Poprzez połączenie tradycyjnych algorytmów, nowoczesnych technologii uczenia maszynowego i praktycznych narzędzi, rozprawa doktorska (zdaniem autora) przyczynia się do postępu w dziedzinie redukcji niespójności w macierzach porównań parami. Wprowadzenie biblioteki REDUCE.py i narzędzi online uczyniło te technologie dostępnymi dla szerszego kręgu użytkowników, promując lepsze praktyki w podejmowaniu decyzji. Algorytm FAST-PCM, będący centralnym elementem tezy, został skutecznie zbadany i zaimplementowany, co otwiera nowe możliwości w skutecznym generowaniu losowych macierzy porównań.

Bibliografia pracy doktorskiej

Bibliografia rozdziału pierwszego

- [1] Triantaphyllou, Evangelos, and Evangelos Triantaphyllou. Multi-criteria decision making methods. Springer US, 2000.
- [2] Brans, Jean-Pierre, and Yves De Smet. "PROMETHEE methods." Multiple criteria decision analysis: state of the art surveys (2016): 187-219.
- [3] Huneau, Frederic, et al. "Intrinsic vulnerability mapping for small mountainous karst aquifers, implementation of the new PaPRIKa method to Western Pyrenees (France)." Engineering Geology 161 (2013): 81-93.
- [4] Ollivier, Chloé, et al. "Challenges and limitations of karst aquifer vulnerability mapping based on the PaPRIKa method—Application to a large European karst aquifer (Fontaine de Vaucluse, France)." Environments 6.3 (2019): 39.
- [5] Kavouri, Konstantina, et al. "PaPRIKa: a method for estimating karst resource and source vulnerability--application to the Ousse karst system (southwest France)." Hydrogeology Journal 19.2 (2011): 339.
- [6] Opricovic, Serafim, and Gwo-Hsiung Tzeng. "Extended VIKOR method in comparison with outranking methods." European journal of operational research 178.2 (2007): 514-529.
- [7] Jahan, Ali, et al. "A comprehensive VIKOR method for material selection." Materials & Design 32.3 (2011): 1215-1221.
- [8] Chatterjee, P., and Sh Chakraborty. "A comparative analysis of VIKOR method and its variants." Decision Science Letters 5.4 (2016): 469-486.
- [9] Mazurek, J. (2023). Advances in Pairwise Comparisons: The Detection, Evaluation and Reduction of Inconsistency, Springer, Heidelberg.
- [10] Saaty T. L. A scaling method for priorities in hierarchical structures. Journal of mathematical psychology 15.3 (1977): 234-281.
- [11] Ishizaka, A., Lusti, M. (2006). How to derive priorities in AHP: A comparative study, Central European Journal of Operations Research 14(4), 387–400. <https://doi.org/10.1007/s10100-006-0012-9>
- [12] Crawford, G. B. (1987). The geometric mean procedure for estimating the scale of a judgement matrix, Mathematical Modelling, 9(3), 327–334. [https://doi.org/10.1016/0270-0255\(87\)90489-1](https://doi.org/10.1016/0270-0255(87)90489-1)
- [13] Bozóki, S., Rapsak, T. (2008). On Saaty's and Koczkodaj's inconsistencies of pairwise comparison matrices, Journal of Global Optimization, 42(2), 157–175.
- [14] Chu, A. T. W., Kalaba, R. E., Spingarn, K. (1979). A comparison of two methods for determining the weights of belonging to fuzzy sets, Journal of Optimization Theory and Applications, 27(4), 531–538.
- [15] Saaty, T. L., Vargas, L. G. (1984). Comparison of eigenvalue, logarithmic least squares and least squares methods in estimating ratios, Mathematical Modelling, 5(5), 309–324
- [16] Siraj, S., Mikhailov, L., Keane, J. (2012). Enumerating all spanning trees for pairwise comparisons, Computers and Operations Research, 39(2), 191–199. <https://doi.org/10.1016/j.cor.2011.03.010>.
- [17] Crawford, G., Williams, C. (1985). A Note on the Analysis of Subjective Judgment Matrices, Journal of Mathematical Psychology, 29(4), 387–405.
- [18] Greco, S., Siraj, S., Lundy, M. (2021). Supporting decisions by unleashing multiple mindsets using pairwise comparisons method. ArXiv, abs/2107.01731.
- [19] Mazurek, J., Kulakowski, K. (2022). On the Derivation of Weights from Incomplete Pairwise Comparisons Matrices via Spanning Trees with Crisp and Fuzzy Confidence Levels. International Journal of Approximate Reasoning (in print).
- [20] Lundy, M., Siraj, S., Greco, S. (2017). The mathematical equivalence of the "spanning tree" and row geometric mean preference vectors and its implications for preference analysis, European Journal of Operational Research, 257(1), 197–208.
- [21] Bozóki, S., Tsyganok, V. (2019). The (logarithmic) least squares optimality of the arithmetic (geometric) mean of weight vectors calculated from all spanning trees for incomplete additive (multiplicative) pairwise comparison matrices, International Journal of General Systems, 48(4), 362–381. <https://doi.org/10.1080/03081079.2019.1585432>.
- [22] Herman, M. W., Koczkodaj, W. W. (1996). A Monte Carlo Study of Pairwise Comparison, Information Processing Letters, 57, 25–29. [https://doi.org/10.1016/0020-0190\(95\)00185-9](https://doi.org/10.1016/0020-0190(95)00185-9).
- [23] Choo, E. U., Wedley, W. C. (2004). A common framework for deriving preference values from pairwise comparison matrices, Computers and Operations Research, 31(6), 893–908. [https://doi.org/10.1016/S0305-0548\(03\)00042-X](https://doi.org/10.1016/S0305-0548(03)00042-X).
- [24] Lin, C.-C. (2007). A revised framework for deriving preference values from pairwise comparison matrices, European Journal of Operational Research, 176(2), 1145–1150. <https://doi.org/10.1016/j.ejor.2005.09.022>.

- [25] Lipovetsky, S. (2009). Comparison of a Dozen AHP Techniques for Global Vectors in Multiperson Decision Making and Complex Hierarchy. 10th International Symposium on the Analytic Hierarchy Process, 1-15. Pittsburgh, EUA
- [26] Mazurek, J., Ramik, J. (2019). Some new properties of inconsistent pairwise comparison matrices. *International Journal of Approximate Reasoning*, 113, 119–132.
- [27] Basile, L., d'Apuzzo, L. (2002). Weak consistency and quasi-linear means imply the actual ranking, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(3), 227–239.
- [28] Bryson, N. (1995). A Goal Programming Method for Generating Priority Vectors. *The Journal of the Operational Research Society*, 46(5), 641–648. <https://doi.org/10.2307/2584536>.
- [29] Dopazo, E., Gonzalez-Pachon, J. (2003). Consistency-driven approximation of a pairwise comparison matrix, *Kybernetika*, 39(5), 561–568.
- [30] Kazibudzki, P. T. (2016). An examination of performance relations among selected consistency measures for simulated pairwise judgments, *Annals of Operations Research*, 244, 525–544. <https://doi.org/10.1007/s10479-016-2131-6>.
- [31] Lin, C., Kou, G., Ergu, D. (2013). A heuristic approach for deriving the priority vector in AHP, *Applied Mathematical Modelling*, 37(8), 5828–5836. <https://doi.org/10.1016/j.apm.2012.11.023>.
- [32] Hovanov, N. V., Kolari, J. W., Sokolov, M. V. (2008). Deriving weights from general pairwise comparison matrices, *Mathematical Social Sciences*, 55(2), 205–220.
- [33] Harker, P. T. (1987). Incomplete pairwise comparisons in the analytic hierarchy process. *Mathematical Modelling*, 9(11), 837–848.
- [34] Boender, C. G. E., De Graan, J. G., & Lootsma, F. (1989). Multi-criteria decision analysis with fuzzy pairwise comparisons. *Fuzzy sets and Systems*, 29(2), 133-143.
- [35] Lootsma, F. A., & Ragsdell, K. M. (1988). State-of-the-art in parallel nonlinear optimization. *Parallel Computing*, 6(2), 133-155.
- [36] Lootsma, Freerk A. *Fuzzy logic for planning and decision making*. Vol. 8. Springer Science & Business Media, 2013.
- [37] Dodd, F. J., H. A. Donegan, and T. B. M. McMaster. "Inverse inconsistency in analytic hierarchies." *European Journal of Operational Research* 80.1 (1995): 86-93.
- [38] Franek, Jiří, and Aleš Kresta. "Judgment scales and consistency measure in AHP." *Procedia economics and finance* 12 (2014): 164-173.
- [39] Gustafsson, Janne, Ahti Salo, and Tommi Gustafsson. "PRIME decisions: An interactive tool for value tree analysis." *Multiple Criteria Decision Making in the New Millennium: Proceedings of the Fifteenth International Conference on Multiple Criteria Decision Making (MCDM) Ankara, Turkey, July 10–14, 2000*. Springer Berlin Heidelberg, 2001.
- [40] Ishizaka, A., Balkenborg, D., & Kaplan, T. (2011). Influence of aggregation and measurement scale on ranking a compromise alternative in AHP. *Journal of the Operational Research Society*, 62(4), 700-710.
- [41] Dong, Y., Xu, Y., Li, H., Dai, M. (2008). A comparative study of the numerical scales and the prioritization methods in AHP. *European Journal of Operational Research*, 186, 229–242. <https://doi.org/10.1016/j.ejor.2007.01.044>.
- [42] Elliott, M. A. (2010). Selecting numerical scales for pairwise comparisons, *Reliability Engineering and System Safety*, 95(7), 750–763.
- [43] Triantaphyllou, E., Lootsma, F. A., Pardalos, P. M., Mann, S. H. (1994). On the Evaluation and Application of Different Scales For Quantifying Pairwise Comparisons in Fuzzy Sets, *Journal of Multi-Criteria Decision Analysis*, 3(3), 133–155.
- [44] Franek, J., Kresta, K. (2014). Judgment Scales and Consistency Measure in AHP, *Procedia, Economics and Finance*, 12, 164–173.
- [45] Cavallo, B., Ishizaka, A. (2022). Evaluating scales for pairwise comparisons, *Annals of Operations Research*. <https://doi.org/10.1007/s10479-022-04682-8>.
- [46] Ishizaka, A., Balkenborg, D., Kaplan T. (2010). Influence of aggregation and measurement scale on ranking a compromise alternative in AHP. *Journal of the Operational Research Society*, 62(4), 700–710.
- [47] Siraj, S., Mikhailov, L., Keane, J. A. (2015). Contribution of individual judgments toward inconsistency in pairwise comparisons, *European Journal of Operational Research*, 242(2), 557–567. <https://doi.org/10.1016/j.ejor.2014.10.024>.
- [48] Lifson, Melvin W., and Edward F. Shaifer. "Decision and Risk Analysis for Construction Management." JOHN WILEY & SONS, INC., 605 THIRD AVE., NEW YORK, NY 10158. 1982. (1982).
- [49] Schuyler, John R. *Risk and decision analysis in projects*. Project Management Inst, 2001.
- [50] Belton, Valerie. *Multiple criteria decision analysis: practically the only way to choose*. University of Strathclyde, Strathclyde Business School, 1990.
- [51] French, Simon, ed. *Decision theory: an introduction to the mathematics of rationality*. Halsted Press, 1986.

- [52] Russell, J. S. (1990). Surety bonding and owner-contractor prequalification: comparison. *Journal of professional issues in engineering*, 116(4), 360-374.
- [53] Von Winterfeldt, Detlof, and Ward Edwards. "Decision analysis and behavioral research." (1993).
- [54] Watson, Stephen R., Dennis M. Buede, and Dennis M. Buede. *Decision synthesis: The principles and practice of decision analysis*. Cambridge University Press, 1987.
- [55] Figueira, J. R., Mousseau, V., & Roy, B. (2016). ELECTRE methods. *Multiple criteria decision analysis: State of the art surveys*, 155-185.
- [56] Figueira, J. R., Greco, S., Roy, B., & Słowiński, R. (2013). An overview of ELECTRE methods and their recent extensions. *Journal of Multi-Criteria Decision Analysis*, 20(1-2), 61-85.
- [57] Figueira, Jose Rui, et al. "ELECTRE methods: Main features and recent developments." *Handbook of multicriteria analysis* (2010): 51-89.
- [58] Rezaei, Jafar. "Best-worst multi-criteria decision-making method." *Omega* 53 (2015): 49-57.
- [59] Mi, Xiaomei, et al. "The state-of-the-art survey on integrations and applications of the best worst method in decision making: Why, what, what for and what's next?." *Omega* 87 (2019): 205-225.
- [60] Vargas, Luis G. "An overview of the analytic hierarchy process and its applications." *European journal of operational research* 48.1 (1990): 2-8.
- [61] Golden, Bruce L., Edward A. Wasil, and Patrick T. Harker. "The analytic hierarchy process." *Applications and Studies*, Berlin, Heidelberg 2.1 (1989): 1-273.
- [62] Saaty, Thomas L. *What is the analytic hierarchy process?*. Springer Berlin Heidelberg, 1988.
- [63] Saaty, Thomas L. "How to make a decision: the analytic hierarchy process." *European journal of operational research* 48.1 (1990): 9-26.
- [64] Saaty, Thomas L., and Kevin P. Kearns. *Analytical planning: The organization of system*. Vol. 7. Elsevier, 2014.
- [65] Expert Choice, Inc., *Expert Choice software and manual*. 4922 Elsworth Ave., Pittsburgh, PA 15213, USA
- [66] Goodman, Claire M. "The Delphi technique: a critique." *Journal of advanced nursing* 12.6 (1987): 729-734.
- [67] Turner, Marlene E., and Anthony R. Pratkanis. "Twenty-five years of groupthink theory and research: Lessons from the evaluation of a theory." *Organizational behavior and human decision processes* 73.2-3 (1998): 105-115.
- [68] Rose, James D. "Diverse perspectives on the groupthink theory—a literary review." *Emerging Leadership Journeys* 4.1 (2011): 37-57.
- [69] Saaty, Thomas L. "Group decision making and the AHP." *The analytic hierarchy process: applications and studies* (1989): 59-67.
- [70] Mazurek, J., Perzina, R., Ramik, J., Bartl, D. (2021). A Numerical Comparison of the Sensitivity of the Geometric Mean Method, Eigenvalue Method, and Best-Worst Method. *Mathematics*, 9, 554. <https://doi.org/10.3390/math9050554>.
- [71] Rezaei, Jafar. "A concentration ratio for nonlinear best worst method." *International Journal of Information Technology & Decision Making* 19.03 (2020): 891-907.
- [72] Rezaei, J. (2015). Best-worst multi-criteria decision-making method, 450 *Omega*, 53, 40-57.
- [73] Rezaei, J. (2016). Best-worst multi-criteria decision-making method: Some properties and a linear model, *Omega*, 64, 126-130.
- [74] Brunelli, M., Rezaei, J. (2019). A multiplicative best-worst method for multi-criteria decision making, *Operations Research Letters*, 47(1), 12-15.
- [75] Saaty, Thomas L. (2005). *Theory and Applications of the Analytic Network Process: Decision Making with Benefits, Opportunities, Costs and Risks*. Pittsburgh, Pennsylvania: RWS Publications. ISBN 1-888603-06-2.
- [76] Saaty, Thomas L. (1996). *Decision Making with Dependence and Feedback: The Analytic Network Process*. Pittsburgh, Pennsylvania: RWS Publications. ISBN 0-9620317-9-8.
- [77] Saaty, Thomas L.; Luis G. Vargas (2006). *Decision Making with the Analytic Network Process: Economic, Political, Social and Technological Applications with Benefits, Opportunities, Costs and Risks*. New York: Springer. ISBN 0-387-33859-4.
- [78] Saaty, Thomas L.; Mütjgan S. Özermir (2005). *The Encyclicon: A Dictionary of Decisions with Dependence and Feedback Based on the Analytic Network Process*. Pittsburgh, Pennsylvania: RWS Publications. ISBN 1-888603-05-4.
- [79] Saaty, Thomas L.; Brady Cillo (2009). *The Encyclicon, Volume 2: A Dictionary of Complex Decisions using the Analytic Network Process*. Pittsburgh, Pennsylvania: RWS Publications. ISBN 978-1-888603-09-5.
- [80] Barzilai, J. (1998). Consistency measures for pairwise comparison matrices, *Journal of Multi-Criteria Decision Analysis*, 7(3), 123–132.
- [81] Golden, B., Wang, Q. (1989). An alternate measure of consistency. In: Golden B., Wasil E. and Harker P. T. (eds), *The Analytic Hierarchy Process, Applications and Studies*, Springer-Verlag: Berlin–Heidelberg, 68–81.

- [82] Koczkodaj, W. W. (1993). A new definition of consistency of pairwise comparisons, *Mathematical and Computer Modeling*, 18(7), 79–84.
- [83] Obata, T. Shiraishi, S., Daigo, M., Nakajima, N. (1999). Assessment for an incomplete comparison matrix and improvement of an inconsistent comparison: computational experiments, ISAHF, Kobe, Japan.
- [84] Peláez, J. I., Lamata, M. T. (2003). A new measure of inconsistency for positive reciprocal matrices. *Computer and Mathematics with Applications*, 2003, 46(12), 1839–1845.
- [85] Ramík, J., Korviny, P. (2010). Inconsistency of pairwise comparison matrix with fuzzy elements based on geometric mean. *Fuzzy Sets and Systems*, 161, 1604–1613.
- [86] Salo, A. A., Hämäläinen, R. (1995). Preference Programming through Approximate Ratio Comparisons. *European Journal of Operational Research* 82(3), 458–475.
- [87] Mazurek, J., Perzina, R., Strzałka, D., Kowal, B., Kuraś, P. (2021). A numerical comparison of iterative algorithms for inconsistency reduction in pairwise comparisons. *IEEE Access*, 9, 62553-62561.
- [88] Cao, D., Leung, L. C., Law, J. S. (2008). Modifying Inconsistent Comparison Matrix in Analytic Hierarchy Process: A Heuristic Approach *Decision Support Systems*, 44, 944-953. doi:10.1016/j.dss.2007.11.002.
- [89] Szybowski, J. (2018). The improvement of data in pairwise comparison matrices. *Procedia Computer Science*, 126, 1006–1013.
- [90] Zeshui, X., Cuiping, W. (1999). A consistency improving method in the analytic hierarchy process. *European journal of operational research*, 116(2), 443–449.
- [91] Kou, G., Ergu, D., Shang, J. (2014). Enhancing data consistency in decision matrix: Adapting Hadamard model to mitigate judgment contradiction. *European Journal of Operational Research*, 236(1), 261–271.
- [92] Mazurek, J., Perzina, R., Strzałka, D., Kowal, B. (2020). A new step-by-step (SBS) algorithm for inconsistency reduction in pairwise comparisons. *IEEE Access*, 8, 135821-135828.
- [93] Abel, E., Mikhailov, L., Keane, J. (2018). Inconsistency reduction in decision making via multi-objective optimisation. *European Journal of Operational Research*, 267(1), 212–226.
- [94] Bozóki, S., Fülöp, J., Poesz, A. (2011). On pairwise comparison matrices that can be made consistent by the modification of a few elements. *Central European Journal of Operations Research*, 19(2), 157–175.
- [95] Bozóki, S., Fülöp, J., Poesz, A. (2015). On reducing inconsistency of pairwise comparison matrices below an acceptance threshold. *Central European Journal of Operations Research*, 23(4), 849–866.
- [96] Negahban, A. (2018). Optimizing consistency improvement of positive reciprocal matrices with implications for Monte Carlo analytic hierarchy process. *Computers & Industrial Engineering*, 124, 113–124.
- [97] Gao, J., Shan, R. (2012). A new method for modification consistency of the judgment matrix based on genetic ant algorithm. *Applied Mathematics & Information Sciences*, 6(1), 35–39.
- [98] Girsang, A. S., Tsai, C. W., Yang, C. S. (2015). Ant algorithm for modifying an inconsistent pairwise weighting matrix in an analytic hierarchy process. *Neural Computing and Applications*, 26(2), 313–327.
- [99] Zhang, H., Sekhari, A., Ouzrout, Y., Bouras, A. (2014). Optimal inconsistency repairing of pairwise comparison matrices using integrated linear programming and eigenvector methods. *Mathematical Problems in Engineering*, 2014.
- [100] Li, H. L., Ma, L. C. (2007). Detecting and adjusting ordinal and cardinal inconsistencies through a graphical and optimal approach in AHP models. *Computers & operations research*, 34(3), 780–798.
- [101] Ergu, D., Kou, G., Peng, Y., Shi, Y. (2011). A simple method to improve the consistency ratio of the pair-wise comparison matrix in ANP. *European Journal of Operational Research*, 213(1), 246–259.
- [102] Kułakowski, K., Juszczyk, R., Ernst, S. (2015). A concurrent inconsistency reduction algorithm for the pairwise comparisons method. In *International Conference on Artificial Intelligence and Soft Computing* (pp. 214–222). Springer, Cham.
- [103] Pereira, V., Costa, H. G. (2015). Nonlinear programming applied to the reduction of inconsistency in the AHP method. *Annals of Operations Research*, 229(1), 635–655.
- [104] Stein, W. E., Mizzi, P. J. (2007). The Harmonic Consistency Index for the Analytic Hierarchy Process, *European Journal of Operational Research* 177(1), 488–497.
- [105] Opricovic, S., & Tzeng, G. H. (2004). Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS. *European journal of operational research*, 156(2), 445-455.
- [106] Yu, Po-Lung. "A class of solutions for group decision problems." *Management science* 19.8 (1973): 936-946.
- [107] Brans, Jean-Pierre. *L'ingénierie de la décision: l'élaboration d'instruments d'aide a la décision*. Université Laval, Faculté des sciences de l'administration, 1982.
- [108] Brans, J. Pierre, and Bertrand Mareschal. "PROMETHEE V: MCDM problems with segmentation constraints." *INFOR: Information Systems and Operational Research* 30.2 (1992): 85-96.

- [109] Brans, Jean-Pierre, and Bertrand Mareschal. "The PROMETHEE VI procedure: how to differentiate hard from soft multicriteria problems." *Journal of Decision Systems* 4.3 (1995): 213-223.
- [110] Mareschal, Bertrand, Jean Pierre Brans, and Cathy Macharis. *The GDSS PROMETHEE procedure: a PROMETHEE-GAIA based procedure for group decision support*. No. 2013/9373. ULB--Universite Libre de Bruxelles, 1998.
- [111] Mareschal, Bertrand, and Jean-Pierre Brans. "Geometrical representations for MCDA." *European journal of operational research* 34.1 (1988): 69-77.
- [112] Brans, Jean-Pierre. "The PROMETHEE-GAIA decision support system for multicriteria investigations." *Investigation Operativa* 4.2 (1994): 107-117.
- [113] Brans, J. P., and B. Mareschal. "Multiple Criteria Decision Analysis: State of the Art Surveys, Figueira, vol. 78." (2005): 1048.
- [114] Figueira, José JR, Yves De Smet, and Jean Pierre Brans. "MCDA methods for sorting and clustering problems: Promethee TRI and Promethee CLUSTER." (2005).
- [115] Macharis, Cathy, et al. "PROMETHEE and AHP: The design of operational synergies in multicriteria analysis.: Strengthening PROMETHEE with ideas of AHP." *European journal of operational research* 153.2 (2004): 307-317.
- [116] Laffy, R.: *La méthode MARSAN pour la recherche de produits nouveaux*. Presented at the ESOMAR congress, Copenhagen (1966)
- [117] Benayoun, R., Roy, B., Sussman, B.: *ELECTRE: Une méthode pour guider le choix en présence de points de vue multiples*. Note de travail 49, SEMA-METRA International, Direction Scientifique (1966)
- [118] Roy, B., Skalka, J.: *ELECTRE IS: aspects méthodologiques et guide d'utilisation*. Document du LAMSADE 30, Université Paris Dauphine (1984)
- [119] Roy, B.: *ELECTRE III: Un algorithme de classements fondé sur une représentation floue des préférences en présence de critères multiples*. *Cahiers du CERO* 20(1), 3–24 (1978)
- [120] Roy, B., Présent, M., Silhol, D.: *A programming method for determining which Paris metro stations should be renovated*. *Eur. J. Oper. Res.* 24, 318–334 (1986)
- [121] Roy, B.: *From optimization to multicriteria decision aid: three main operational attitudes*. In: Thiriez, H., Zionts, S. (eds.) *Multiple Criteria Decision Making*. *Lecture Notes in Economics and Mathematical Systems*, vol. 130, pp. 1–34. Springer, Berlin (1976)
- [122] Gargaillo, L.: Réponse à l'article "Le plan d'extension du métro en banlieue parisienne, un cas type de l'analyse multicritère". *Les Cahiers Scientifiques de la Revue Transports* 7, 52–57 (1982)
- [123] Hugonnard, J., Roy, B.: *Le plan d'extension du métro en banlieue parisienne, un cas type d'application de l'analyse multicritère*. *Les Cahiers Scientifiques de la Revue Transports* 6, 77–108 (1982)
- [124] Roy, B., Hugonnard, J.: *Classement des prolongements de lignes de métro en banlieue parisienne (présentation d'une méthode multicritère originale)*. *Cahiers du CERO* 24(2–4), 153–171 (1982)
- [125] Roy, B., Hugonnard, J.: *Le plan d'extension du métro en banlieue parisienne, un cas type de l'analyse multicritère*. *Les Cahiers Scientifiques de la Revue Transports* 6, 77–108 (1982)
- [126] Roy, B., Hugonnard, J.: Réponse à Monsieur Gargaillo. *Les Cahiers Scientifiques de la Revue Transports* 7, 58–64 (1982)
- [127] Roy, B.: *The outranking approach and the foundations of ELECTRE methods*. *Theor. Decis.* 31, 49–73 (1991)
- [128] Roy, B.: *The outranking approach and the foundations of ELECTRE methods*. In: Bana e Costa, C.A. (ed.) *Readings in Multiple Criteria Decision Aid*, pp. 155–183. Springer, Berlin (1990)
- [129] Roy, B., Bertier, P.: *La méthode ELECTRE II*. Note de travail 142, SEMA-METRA Metra International (1971)
- [130] Roy, B., Bertier, P.: *La méthode ELECTRE II – Une application au média-planning*. In: Ross, M. (ed.) *OR'72*, pp. 291–302. North-Holland, Amsterdam (1973)
- [131] Roy, B., Bouyssou, D.: *Comparison of two decision-aid models applied to a nuclear plant siting example*. *Eur. J. Oper. Res.* 25, 200–215 (1986)
- [132] Roy, B., Bouyssou, D.: *Aide Multicritère à la Décision: Méthodes et Cas*. *Economica*, Paris (1993)
- [133] Mousseau, V., Słowiński, R.: *Inferring an ELECTRE TRI model from assignment examples*. *J. Glob. Optim.* 12(2), 157–174 (1998)
- [134] Mousseau, V., Słowiński, R., Zielniewicz, P.: *ELECTRE TRI 2.0a: methodological guide and user's documentation*. Document du LAMSADE 111, Université Paris-Dauphine (1999)
- [135] Maystre, L., Pictet, J., Simos, J.: *Les Méthodes Multicritères ELECTRE*. Presses Polytechniques et Universitaires Romandes, Lausanne (1994)
- [136] Roy, B., Bouyssou, D.: *Aide Multicritère à la Décision: Méthodes et Cas*. *Economica*, Paris (1993)
- [137] Bozóki, Sándor & Lewis, Robert. (2005). *Solving the Least Squares Method problem in the AHP for 3 x 3 and 4 x 4 matrices*.
- [138] Ra, J. W. (1999). *Chainwise paired comparisons*. *Decision Sciences*, 30(2), 581-599.

- [139] Saaty, T. L. (2003). Decision-making with the AHP: Why is the principal eigenvector necessary. *European journal of operational research*, 145(1), 85-91.
- [140] Ringuest, J. L. (1997). Lp-metric sensitivity analysis for single and multi-attribute decision analysis. *European Journal of Operational Research*, 98(3), 563-570.

Bibliografia rozdziału drugiego

- [1] E. Abel, L. Mikhailov and J. Keane, "Inconsistency reduction in decision making via multi-objective optimisation", *Eur. J. Oper. Res.*, vol. 267, no. 1, pp. 212-226, May 2018.
- [2] J. A. Alonso and M. T. Lamata, "Consistency in the analytic hierarchy process: A new approach", *Int. J. Uncertainty Fuzziness Knowl.-Based Syst.*, vol. 14, no. 4, pp. 445-459, 2006.
- [3] J. Benítez, X. Delgado-Galván, J. A. Gutiérrez and J. Izquierdo, "Balancing consistency and expert judgment in AHP", *Math. Comput. Model.*, vol. 54, no. 7, pp. 1785-1790, Oct. 2011.
- [4] J. Benítez, X. Delgado-Galván, J. Izquierdo and R. Pérez-García, "Improving consistency in AHP decision-making processes", *Appl. Math. Comput.*, vol. 219, no. 5, pp. 2432-2441, Nov. 2012.
- [5] J. Benítez, J. Izquierdo, R. Pérez-García and E. Ramos-Martínez, "A simple formula to find the closest consistent matrix to a reciprocal matrix", *Appl. Math. Model.*, vol. 38, no. 15, pp. 3968-3974, Aug. 2014.
- [6] S. Bozóki, J. Fülöp and A. Poesz, "On pairwise comparison matrices that can be made consistent by the modification of a few elements", *Central Eur. J. Oper. Res.*, vol. 19, no. 2, pp. 157-175, Jun. 2011.
- [7] S. Bozóki, J. Fülöp and A. Poesz, "On reducing inconsistency of pairwise comparison matrices below an acceptance threshold", *Central Eur. J. Oper. Res.*, vol. 23, no. 4, pp. 849-866, Dec. 2015.
- [8] M. Brunelli, L. Canal and M. Fedrizzi, "Inconsistency indices for pairwise comparison matrices: A numerical study", *Ann. Oper. Res.*, vol. 211, no. 1, pp. 493-509, Dec. 2013.
- [9] M. Brunelli and M. Fedrizzi, "Axiomatic properties of inconsistency indices for pairwise comparisons", *J. Oper. Res. Soc.*, vol. 66, no. 1, pp. 1-15, Jan. 2015.
- [10] M. Brunelli, "Studying a set of properties of inconsistency indices for pairwise comparisons", *Ann. Oper. Res.*, vol. 248, no. 1, pp. 143-161, Jan. 2017.
- [11] D. Cao, L. C. Leung and J. S. Law, "Modifying inconsistent comparison matrix in analytic hierarchy process: A heuristic approach", *Decis. Support Syst.*, vol. 44, no. 4, pp. 944-953, Mar. 2008.
- [12] L. Csató, "Axiomatizations of inconsistency indices for triads", *Ann. Oper. Res.*, vol. 280, no. 1, pp. 99-110, Sep. 2019.
- [13] D. Ergu, G. Kou, Y. Peng and Y. Shi, "A simple method to improve the consistency ratio of the pair-wise comparison matrix in ANP", *Eur. J. Oper. Res.*, vol. 213, no. 1, pp. 246-259, Aug. 2011.
- [14] J. Figueira, S. Greco and M. Ehrgott, *Multiple Criteria Decision Analysis: State Art Surveys*, New York, NY, USA:Springer, 2005.
- [15] J. Gao, R. Shan, H. Cui and W. Ren, "A new method for modification consistency of the judgment matrix based on genetic ant algorithm", *Appl. Math. Inf. Sci.*, vol. 6, no. 1, pp. 1903-1906, 2011.
- [16] Z. Gao, L. Guo, W. Guan, A.-A. Liu, T. Ren and S. Chen, "A pairwise attentive adversarial spatiotemporal network for cross-domain few-shot action recognition-R2", *IEEE Trans. Image Process.*, vol. 30, pp. 767-782, 2021.
- [17] Z. Gao, Y. Li, W. Guan, W. Nie and Z. A. Cheng Liu, "Pairwise view weighted graph network for view-based 3D model retrieval", *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, pp. 129-138, 2020.
- [18] A. S. Girsang, C.-W. Tsai and C.-S. Yang, "Ant algorithm for modifying an inconsistent pairwise weighting matrix in an analytic hierarchy process", *Neural Comput. Appl.*, vol. 26, no. 2, pp. 313-327, Feb. 2015.
- [19] W. Holsztynski and W. W. Koczkodaj, "Convergence of Inconsistency Algorithms for the Pairwise Comparisons", *Inf. Process. Lett.*, vol. 59, no. 4, pp. 197-202, 1996.
- [20] S. Jarek, "Removing inconsistency in pairwise comparison matrix in the AHP", *Multiple Criteria Decis. Making*, vol. 11, pp. 63-76, Dec. 2016.
- [21] M. Kendall, *Rank Correlation Methods*, Glasgow, Scotland :Charles Griffin & Company, 1948.
- [22] W. W. Koczkodaj, "A new definition of consistency of pairwise comparisons", *Math. Comput. Model.*, vol. 18, no. 7, pp. 79-84, Oct. 1993.
- [23] W. W. Koczkodaj and S. J. Szarek, "On distance-based inconsistency reduction algorithms for pairwise comparisons", *Log. J. IGPL*, vol. 18, no. 6, pp. 859-869, Dec. 2010.

- [24] W. W. Koczkodaj, M. Kosiek, J. Szybowski and D. Xu, "Fast convergence of distance-based inconsistency in pairwise comparisons", *Fundam. Inf.*, vol. 137, no. 3, pp. 355-367, 2015.
- [25] G. Kou, D. Ergu and J. Shang, "Enhancing data consistency in decision matrix: Adapting Hadamard model to mitigate judgment contradiction", *Eur. J. Oper. Res.*, vol. 236, no. 1, pp. 261-271, Jul. 2014.
- [26] K. Kulakowski and J. Szybowski, "The new triad based inconsistency indices for pairwise comparisons", *Procedia Comput. Sci.*, vol. 35, pp. 1132-1137, Dec. 2014.
- [27] K. Kulakowski, R. Juszczyk and S. Ernst, "A concurrent inconsistency reduction algorithm for the pairwise comparisons method", *Proc. Int. Conf. Artif. Intell. Soft Comput.*, pp. 214-222, 2015.
- [28] M. T. Lamata and J. I. Pelaez, "A Method for Improving the Consistency of Judgments", *Int. J. Uncertainty Fuzziness Knowl.-Based Syst.*, vol. 10, pp. 677-686, 2002.
- [29] I. Lerche and B. S. Mudford, "How many Monte Carlo simulations does one need to do?", *Energy Explor.*, vol. 23, no. 6, pp. 405-427, 2005.
- [30] H.-L. Li and L.-C. Ma, "Detecting and adjusting ordinal and cardinal inconsistencies through a graphical and optimal approach in AHP models", *Comput. Oper. Res.*, vol. 34, no. 3, pp. 780-798, Mar. 2007.
- [31] A. Mardani, A. Jusoh, K. Md Nor, Z. Khalifah, N. Zakwan and A. Valipour, "Multiple criteria decision-making techniques and their applications—A review of the literature from 2000 to 2014", *Econ. Res.-Ekonomika Istraživanja*, vol. 28, no. 1, pp. 516-571, Jan. 2015.
- [32] J. Mazurek, "Some notes on the properties of inconsistency indices in pairwise comparisons", *Oper. Res. Decis.*, vol. 1, pp. 27-42, Dec. 2018.
- [33] J. Mazurek and R. Perzina, "On the inconsistency of pairwise comparisons: An experimental study", *Sci. Univ. Pardubice-Series D3 Fac. Econ. Admin.*, vol. 41, pp. 102-109, Dec. 2017.
- [34] J. Mazurek and J. Ramik, "Some new properties of inconsistent pairwise comparisons matrices", *Int. J. Approx. Reasoning*, vol. 113, pp. 119-132, Oct. 2019.
- [35] J. Mazurek, R. Perzina, D. Strzalka and B. Kowal, "A new step-by-step (SBS) algorithm for inconsistency reduction in pairwise comparisons", *IEEE Access*, vol. 8, pp. 135821-135828, 2020.
- [36] A. Negahban, "Optimizing consistency improvement of positive reciprocal matrices with implications for Monte Carlo analytic hierarchy process", *Comput. Ind. Eng.*, vol. 124, pp. 113-124, Oct. 2018.
- [37] J. I. Peláez and M. T. Lamata, "A new measure of consistency for positive reciprocal matrices", *Comput. Math. with Appl.*, vol. 46, no. 12, pp. 1839-1845, Dec. 2003.
- [38] N. Pankratova and N. Nedashkovskaya, "Methods of evaluation and improvement of consistency of expert pairwise comparison judgements", *Inf. Theories Appl.*, vol. 22, no. 3, pp. 203-223, 2015.
- [39] V. Pereira and H. G. Costa, "AHP inconsistency reduction through two greedy algorithms application", *engrXiv*, 2019.
- [40] V. Pereira and H. G. Costa, "Nonlinear programming applied to the reduction of inconsistency in the AHP method", *Ann. Oper. Res.*, vol. 229, no. 1, pp. 635-655, Jun. 2015.
- [41] J. Rezaei, "Best-worst multi-criteria decision-making method: Some properties and a linear model", *Omega*, vol. 64, pp. 126-130, Oct. 2016.
- [42] T. L. Saaty, "A scaling method for priorities in hierarchical structures", *J. Math. Psychol.*, vol. 15, pp. 234-281, Oct. 1977.
- [43] T. L. Saaty, *Analytic Hierarchy Process*, New York, NY, USA:McGraw-Hill, 1980.
- [44] J. Szybowski, "The improvement of data in pairwise comparison matrices", *Procedia Comput. Sci.*, vol. 126, pp. 1006-1013, Mar. 2018.
- [45] T.-Y. Tseng, S.-W. Lin, C.-L. Huang and R. Lee, "Inconsistency adjustment in the AHP using the complete transitivity convergence algorithm", *Proc. IEEE Int. Conf. Syst. Man Cybern.*, pp. 2808-2812, Oct. 2006.
- [46] G. R. Vasconcelos and C. Maria De Miranda Mota, "Exploring multicriteria elicitation model based on pairwise comparisons: Building an interactive preference adjustment algorithm", *Math. Problems Eng.*, vol. 2019, Jun. 2019.
- [47] Z. Xu and C. Wei, "A consistency improving method in the analytic hierarchy process", *Eur. J. Oper. Res.*, vol. 116, pp. 443-449, Oct. 1999.
- [48] K. Xu and J. Xu, "A direct consistency test and improvement method for the analytic hierarchy process", *Fuzzy Optim. Decis. Making*, vol. 19, no. 3, pp. 359-388, Sep. 2020.
- [49] H. Zhang, A. Sekhari, Y. Ouzrout and A. Bouras, "Optimal inconsistency repairing of pairwise comparison matrices using integrated linear programming and eigenvector methods", *Math. Problems Eng.*, vol. 2014, Dec. 2014.
- [50] GitHub Repository, [online] Available: <https://github.com/pawkuras/>

- [51] J. Mazurek, R. Perzina, D. Strzałka, B. Kowal and P. Kuraś, "A Numerical Comparison of Iterative Algorithms for Inconsistency Reduction in Pairwise Comparisons," in *IEEE Access*, vol. 9, pp. 62553-62561, 2021, doi: 10.1109/ACCESS.2021.3074274.

Bibliografia rozdziału trzeciego

- [1] Saaty T.L., A Scaling Method for Priorities in Hierarchical Structures. *Journal of Mathematical Psychology*. 1977;15:234–281.
- [2] Mazurek J., *Advances in Pairwise Comparisons: The Detection, Evaluation and Reduction of Inconsistency*. Heidelberg: Springer; 2023.
- [3] Barzilai J., Consistency measures for pairwise comparison matrices. *Journal of Multi-Criteria Decision Analysis*. 1998;7(3):123–132.
- [4] Golden B., Wang Q., An alternate measure of consistency. In: Golden B, Wasil E, Harker PT, editors. *The Analytic Hierarchy Process, Applications and Studies*. Berlin–Heidelberg: Springer-Verlag; 1989. p. 68–81.
- [5] Koczkodaj W.W., A new definition of consistency of pairwise comparisons. *Mathematical and Computer Modeling*. 1993;18(7):79–84.
- [6] Obata T., Shiraishi S., Daigo M., Nakajima N., Assessment for an incomplete comparison matrix and improvement of an inconsistent comparison: computational experiments. *ISAHP*. Kobe, Japan; 1999.
- [7] Peláez J.I., Lamata M.T., A new measure of inconsistency for positive reciprocal matrices. *Computer and Mathematics with Applications*. 2003; 46(12):1839–1845.
- [8] Aguarón J., Moreno-Jiménez J.M., The geometric consistency index: Approximated threshold. *European Journal of Operational Research*. 2003;147(1):137–145.
- [9] Aguarón J., Escobar M.T., Moreno-Jiménez J.M., Turón A., The Triads Geometric Consistency Index in AHP-Pairwise Comparison Matrices. *Mathematics*. 2020;8:926.
- [10] Stein W.E., Mizzi P.J., The Harmonic Consistency Index for the Analytic Hierarchy Process. *European Journal of Operational Research*. 2007;177(1):488–497.
- [11] Mazurek J, Perzina R, Strzałka D, Kowal B, Kuraś P. A numerical comparison of iterative algorithms for inconsistency reduction in pairwise comparisons. *IEEE Access*. 2021;9:62553-62561.
- [12] Cao D., Leung L.C., Law J.S., Modifying Inconsistent Comparison Matrix in Analytic Hierarchy Process: A Heuristic Approach. *Decision Support Systems*. 2008;44:944-953. doi:10.1016/j.dss.2007.11.002.
- [13] Szybowski J., The improvement of data in pairwise comparison matrices. *Procedia Computer Science*. 2018;126:1006–1013.
- [14] Zeshui X., Cuiping W., A consistency improving method in the analytic hierarchy process. *European Journal of Operational Research*. 1999;116(2):443–449.
- [15] Kou G., Ergu D., Shang J., Enhancing data consistency in decision matrix: Adapting Hadamard model to mitigate judgment contradiction. *European Journal of Operational Research*. 2014;236(1):261–271.
- [16] Mazurek J., Perzina R., Strzałka D, Kowal B., A new step-by-step (SBS) algorithm for inconsistency reduction in pairwise comparisons. *IEEE Access*. 2020;8:135821-135828.
- [17] Harris C.R., Millman K.J., van der Walt S.J., Gommers R., Virtanen P., Cournapeau D., et al. Array programming with NumPy. *Nature*. 2020;585(7825):357–362. Available from: <https://doi.org/10.1038/s41586-020-2649-2>
- [18] Virtanen P., Gommers R, Oliphant T.E., Haberland M., Reddy T., Cournapeau D., et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat Methods*. 2020;17:261–272. Available from: <https://doi.org/10.1038/s41592-019-0686-2>
- [19] Meurer A, Smith C.P., Paprocki M., Čertík O., Kirpichev S.B., Rocklin M., et al. SymPy: symbolic computing in Python. *PeerJ Computer Science*. 2017;3:e103.
- [20] Department of Complex Systems, Rzeszow University of Technology. REDUCE [Internet]. GitHub. 2023. Available from: <https://github.com/zszprz/reduce>. DOI:10.5281/zenodo.7335819
- [21] Wu, Y., Zhang, Y., Wang, T., & Wang, H. (2020). Characterizing the occurrence of dockerfile smells in open-source software: An empirical study. *IEEE Access*, 8, 34127-34139.
- [22] Zhang, Y., Yin, G., Wang, T., Yu, Y., & Wang, H. (2018, July). An insight into the impact of dockerfile evolutionary trajectories on quality and latency. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (Compsac)* (Vol. 1, pp. 138-143). IEEE.
- [23] Simoes, A., Fonseca, R., & Almeida, J. J. (2007). Makefile:: Parallel dependency specification language. In *Euro-Par 2007 Parallel Processing: 13th International Euro-Par Conference, Rennes, France, August 28-31, 2007. Proceedings 13* (pp. 33-41). Springer Berlin Heidelberg.

- [24] python-poetry/poetry: Python Packaging and Dependency Management Made Easy. URL: <https://github.com/python-poetry/poetry> (visited on 04/03/2023).
- [25] Liu, X. (2022). Object-sensitive Type Analysis for Python (Master's thesis).
- [26] Rak-Amnourykit, I., McCrean, D., Milanova, A., Hirzel, M., & Dolby, J. (2020, November). Python 3 types in the wild: a tale of two type systems. In Proceedings of the 16th ACM SIGPLAN International Symposium on Dynamic Languages (pp. 57-70).
- [27] Okken, B. (2022). Python Testing with pytest. Pragmatic Bookshelf.
- [28] Combe, T., Martin, A., & Di Pietro, R. (2016). To docker or not to docker: A security perspective. *IEEE Cloud Computing*, 3(5), 54-62.
- [29] Maryka, T., German, D. M., & Poo-Caamaño, G. (2015). On the Variability of the BSD and MIT Licenses. In Open Source Systems: Adoption and Impact: 11th IFIP WG 2.13 International Conference, OSS 2015, Florence, Italy, May 16-17, 2015, Proceedings 11 (pp. 146-156). Springer International Publishing.
- [30] Laato, S., Mäntymäki, M., Islam, A. N., Hyrynsalmi, S., & Birkstedt, T. (2023). Trends and Trajectories in the Software Industry: implications for the future of work. *Information Systems Frontiers*, 25(2), 929-944.
- [31] Lakhani, K. R., & Von Hippel, E. (2004). How open source software works: "free" user-to-user assistance (pp. 303-339). Gabler Verlag.
- [32] Kumar, S. (2006). Enforcing the Gnu GPL. U. Ill. JL Tech. & Pol'y, 1.
- [33] Sinclair, A. (2010). License profile: Apache license, ver. 2.0. *IFOSS L. Rev.*, 2, 107.
- [34] Rosen, L. (2005). Open source licensing. *Soft. Freedom & Intellectual Property Law*.
- [35] Laurent, A. M. S. (2004). Understanding open source and free software licensing: guide to nav. licensing issues in existing & new software. "O'Reilly Media, Inc."
- [36] Olan, M. (2003). Unit testing: test early, test often. *Journal of Computing Sciences in Colleges*, 19(2), 319-328.
- [37] Khorikov, V. (2020). Unit Testing Principles, Practices, & Patterns. Simon & Schuster.
- [38] Papadimitriou, C. H. (2003). Computational complexity. In *Encyclopedia of computer science* (pp. 260-265).
- [39] Goldreich, O. (2008). Computational complexity: a conceptual perspective. *ACM Sigact News*, 39(3), 35-39.
- [40] Fortnow, L., & Homer, S. (2003). A short history of computational complexity. *Bulletin of the EATCS*, 80(01), 2003.
- [41] Sileshi, B. G., Ferrer, C., & Oliver, J. (2013, October). Particle filters and resampling techniques: Importance in computational complexity analysis. In 2013 Conference on Design and Architectures for Signal and Image Processing (pp. 319-325). IEEE.
- [42] Du, D. Z., & Ko, K. I. (2011). *Theory of computational complexity* (Vol. 58). John Wiley & Sons.
- [43] Van Veldhuizen, D. A., & Lamont, G. B. (2000, July). On measuring multiobjective evolutionary algorithm performance. In Proceedings of the 2000 congress on evolutionary computation. CEC00 (Cat. No. 00TH8512) (Vol. 1, pp. 204-211). IEEE.
- [44] Thissen, D., & Mislevy, R. J. (2000). Testing algorithms. In *Computerized adaptive testing* (pp. 101-133). Routledge.
- [45] Tedford, N. P., & Martins, J. R. (2010). Benchmarking multidisciplinary design optimization algorithms. *Optimization and Engineering*, 11, 159-183.
- [46] <https://reduce.prz.edu.pl/dane>
- [47] Kuraś P. GitHub [Internet]. 2022. Available from: https://github.com/pawkuras/PCM_CR
- [48] Kowal B, Kuraś P, Strzałka D, Mazurek J, Perzina R. REDUCE. 3rd International conference on Decision making for Small and Medium-Sized Enterprises DEMSME 2021. Conference Proceedings.
- [49] Kowal B, Kuraś P, Strzałka D, Mazurek J, Perzina R. REDUCE [Internet]. 2021. Available from: <https://reduce.prz.edu.pl/>
- [50] Kuraś P, Gerka A. Using inconsistency reduction algorithms in comparison matrices to improve the performance of generating random comparison matrices with a given inconsistency coefficient range. *Advances in Science and Technology Research Journal*. 2023;17(1):222-229.
- [51] Kuraś P. Matrices generator [Internet]. 2022. Available from: https://reduce.prz.edu.pl/pc_matrices_generator
- [52] Erdogan S.A., Šaparauskas J., Turskis Z. Decision making in Construction Management: AHP and Expert Choice approach. *Procedia Engineering*. 2017;172:270-276.
- [53] Sajjad A, Ahmad W, Hussain S. Decision-Making Process Development for Industry 4.0 Transformation. *Advances in Science and Technology Research Journal*. 2022;16(3):1-11. <https://doi.org/10.12913/22998624/147237>
- [54] Kozera R, Smarzewski R. Constructive Consistent Approximations in Pairwise Comparisons. *Advances in Science and Technology Research Journal*. 2022;16(4):243-255. <https://doi.org/10.12913/22998624/153086>

- [55] Stevanović D, Lekić M, Krzanovic D, Ristović I. Application of MCDA in Selection of Different Mining Methods and Solutions. *Advances in Science and Technology Research Journal*. 2018;12(1):171-180.
<https://doi.org/10.12913/22998624/85804>
- [56] Ramík J, Korviny P. Inconsistency of pairwise comparison matrix with fuzzy elements based on geometric mean. *Fuzzy Sets and Systems*. 2010;161:1604–1613.
- [57] Salo A.A., Hämäläinen R. Preference Programming through Approximate Ratio Comparisons. *European Journal of Operational Research*. 1995;82(3):458–475.
- [58] Abel E, Mikhailov L, Keane J. Inconsistency reduction in decision making via multi-objective optimisation. *European Journal of Operational Research*. 2018;267(1):212–226.
- [59] Bozóki S, Fülöp J, Poesz A. On pairwise comparison matrices that can be made consistent by the modification of a few elements. *Central European Journal of Operations Research*. 2011;19(2):157–175.
- [60] Bozóki S, Fülöp J, Poesz A. On reducing inconsistency of pairwise comparison matrices below an acceptance threshold. *Central European Journal of Operations Research*. 2015;23(4):849–866.
- [61] Negahban A. Optimizing consistency improvement of positive reciprocal matrices with implications for Monte Carlo analytic hierarchy process. *Computers & Industrial Engineering*. 2018;124:113–124.
- [62] Gao J, Shan R. A new method for modification consistency of the judgment matrix based on genetic ant algorithm. *Applied Mathematics & Information Sciences*. 2012;6(1):35–39.
- [63] Girsang A.S., Tsai C.W., Yang C.S. Ant algorithm for modifying an inconsistent pairwise weighting matrix in an analytic hierarchy process. *Neural Computing and Applications*. 2015;26(2):313–327.s
- [64] Zhang H, Sekhari A, Ouzrout Y, Bouras A. Optimal inconsistency repairing of pairwise comparison matrices using integrated linear programming and eigenvector methods. *Mathematical Problems in Engineering*. 2014.
- [65] Li H.L, Ma L.C. Detecting and adjusting ordinal and cardinal inconsistencies through a graphical and optimal approach in AHP models. *Computers & Operations Research*. 2007;34(3):780–798.
- [66] Ergu D, Kou G, Peng Y, Shi Y. A simple method to improve the consistency ratio of the pair-wise comparison matrix in ANP. *European Journal of Operational Research*. 2011;213(1):246–259.
- [67] Kułakowski K, Juszczak R, Ernst S. A concurrent inconsistency reduction algorithm for the pairwise comparisons method. In *International Conference on Artificial Intelligence and Soft Computing*. Cham: Springer; 2015. p. 214–222.
- [68] Pereira V, Costa H.G. Nonlinear programming applied to the reduction of inconsistency in the AHP method. *Annals of Operations Research*. 2015;229(1):635–655.
- [69] Perzina, R., Ramík, J. (2016). Fuzzy Decision Analysis Module for Excel. In: *Proceedings of the 34th International Conference on Mathematical Methods in Economics 2016*. Liberec: Technical University Liberec, pp. 663-668.
- [70] Perzina, R., Ramík, J. (2015). Educational Microsoft excel add-ins: Solving multicriteria decision making problems. In: *Proceedings of CSEDU 2015 - 7th International Conference on Computer Supported Education*. Lisabon: SciTePress, pp. 510-515.
- [71] Perzina, R., Ramík, J. (2012). DAME - Microsoft excel add-in for solving multicriteria decision problems with scenarios. In: *Proceedings of 30th International Conference Mathematical Methods in Economics*. Karviná: OPF SU, pp. 697-702.

Bibliografia rozdziału czwartego

- [1] <https://reduce.prz.edu.pl/>
- [2] Grinberg, M. (2018). *Flask web development: developing web applications with python*. " O'Reilly Media, Inc."
- [3] Kuraś, P., Strzalka, D., Kowal, B., & Mazurek, J. (2023). REDUCE: A Python module for reducing inconsistency in Pairwise Comparison Matrices. *Advances in Science and Technology. Research Journal*, 17(4), 227-234.
- [3] Robbins, J. N. (2012). *Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics*. " O'Reilly Media, Inc."
- [4] Duckett, J. (2014). *Web design with HTML, CSS, JavaScript and jQuery set*. Wiley Publishing.
- [5] Goodman, D. (2002). *Dynamic HTML: The definitive reference: A comprehensive resource for HTML, CSS, DOM & JavaScript*. " O'Reilly Media, Inc."
- [6] Ranjan, A., Sinha, A., & Battewad, R. (2020). *JavaScript for modern web development: building a web application using HTML, CSS, and JavaScript*. BPB Publications.
- [7] Mueller, J. P. (2015). *Security for Web Developers: Using JavaScript, HTML, and CSS*. " O'Reilly Media, Inc."
- [8] Bibeault, B., De Rosa, A., & Katz, Y. (2015). *jQuery in Action*. Simon and Schuster.
- [9] Ghimire, D. (2020). *Comparative study on Python web frameworks: Flask and Django*.

- [10] Eichorn, J. (2006). *Understanding AJAX: Using JavaScript to create rich internet applications*. Prentice Hall PTR.
- [11] Rajabov, S. B. (2023). The role of backend and frontend information systems infrastructure. *Science and Education*, 4(3), 212-216.
- [12] Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357-362.
- [13] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... & Van Mulbregt, P. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods*, 17(3), 261-272.
- [14] Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., ... & Scopatz, A. (2017). SymPy: symbolic computing in Python. *PeerJ Computer Science*, 3, e103.
- [15] Botwe, D. A., & Davis, J. G. (2015). A comparative study of web development technologies using open source and proprietary software. *International Journal of Computer Science and Mobile Computing*, 4(2), 154-165.
- [16] <https://zsz.prz.edu.pl/laboratoria/laboratorium-eksploatacji-systemow-informatycznych>
- [17] LaCroix, J. (2018). *Mastering Ubuntu Server: Master the art of deploying, configuring, managing, and troubleshooting Ubuntu Server 18.04*. Packt Publishing Ltd.
- [18] Gupta, A., & Sharma, L. S. (2020). Performance evaluation of snort and suricata intrusion detection systems on ubuntu server. In *Proceedings of ICRIC 2019: Recent Innovations in Computing* (pp. 811-821). Springer International Publishing.
- [19] Chesneau, B. (2017). Unicorn documentation.
- [20] Reese, W. (2008). Nginx: the high-performance web server and reverse proxy. *Linux Journal*, 2008(173), 2.
- [21] Aas, J., Barnes, R., Case, B., Durumeric, Z., Eckersley, P., Flores-López, A., ... & Warren, B. (2019, November). Let's Encrypt: an automated certificate authority to encrypt the entire web. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (pp. 2473-2487).
- [22] Kyambade, G., & Ngubiri, J. (2018). A tool to mitigate denial of service attacks on wired networks.
- [23] Denney, R. (2008). Configuring a Cisco ASA 5505.
- [24] <https://usk.prz.edu.pl/>
- [25] Saaty, T. L. (1977) A Scaling Method for Priorities in Hierarchical Structures. *Journal of Mathematical Psychology*, No. 15, pp. 234-281.
- [26] Saaty, T. L. (1980). *Analytic Hierarchy Process*. McGraw-Hill, New York.
- [27] Xu Z., Wei C. (1999). A Consistency Improving Method in the Analytic Hierarchy Process. *European Journal of Operational Research*, No. 116, pp. 443--449. doi: 10.1016/S0377-2217(98)00109-X.
- [28] Szybowski, J. (2018). The improvement of data in pairwise comparison matrices, *Procedia Computer Science*, No. 126, pp. 1006-1013. doi: 10.1016/j.procs.2018.08.036.
- [29] Koczkodaj, W. W. (1993). A new definition of consistency of pairwise comparisons, *Mathematical and Computer Modeling*, 18(7), 79–84.
- [30] Ramík, J., Korviny, P. (2010). Inconsistency of pairwise comparison matrix with fuzzy elements based on geometric mean. *Fuzzy Sets and Systems*, 161, 1604–1613.
- [31] Golden, B., Wang, Q. (1989). An alternate measure of consistency. In: Golden B., Wasil E. and Harker P. T. (eds), *The Analytic Hierarchy Process, Applications and Studies*, Springer-Verlag: Berlin–Heidelberg, 68–81.
- [32] Peláez, J. I., Lamata, M. T. (2003). A new measure of inconsistency for positive reciprocal matrices. *Com-puter and Mathematics with Applications*, 2003, 46(12), 1839–1845.
- [33] Salo, A. A., Hämäläinen, R. (1995). Preference Programming through Approximate Ratio Comparisons. *European Journal of Operational Research* 82(3), 458–475.
- [34] Barzilai, J. (1998). Consistency measures for pairwise comparison matrices, *Journal of Multi-Criteria Deci-sion Analysis*, 7(3), 123–132.
- [35] Stein, W. E., Mizzi, P. J. (2007). The Harmonic Consistency Index for the Analytic Hierarchy Process, *European Journal of Operational Research* 177(1), 488–497.
- [36] Idris, N., Foozy, C. F. M., & Shamala, P. (2020). A generic review of web technology: Django and flask. *International Journal of Advanced Science Computing and Engineering*, 2(1), 34-40.
- [37] Mirbadiev, S. Z., Filipkina, T. N., & Popov, A. A. (2019). THE COMPARATIVE ANALYSIS OF FLASK AND DJANGO WEB FRAMEWORKS. *Modern Science*, (2), 19-23.
- [38] Ashley, D., & Ashley, D. (2020). Comparing CGI, SSI, Flask, and Django. *Foundation Dynamic Web Pages with Python: Create Dynamic Web Pages with Django and Flask*, 201-208.
- [39] Göransson, J. (2021). Exploring the technical requirements for an immersive biathlon experience: Evaluating Nginx, Flask and Django for video streaming.

- [40] Ashley, D., & Ashley, D. (2020). Using flask and jinja. *Foundation Dynamic Web Pages with Python: Create Dynamic Web Pages with Django and Flask*, 159-181.
- [41] Lokhande, P. S., Aslam, F., Hawa, N., Munir, J., & Gulamgaus, M. (2015). Efficient way of web development using python and flask.
- [42] Agresti, A., & Kateri, M. (2021). *Foundations of statistics for data scientists: with R and Python*. CRC Press.
- [43] Sarmiento, R., & Costa, V. (Eds.). (2017). *Comparative approaches to using R and Python for statistical data analysis*. IGI Global.
- [44] Team, R. C. (2000). *R language definition*. Vienna, Austria: R foundation for statistical computing, 3(1).
- [45] Rahlf, T. (2017). *Data visualisation with R: 100 examples*. Springer.
- [46] Rahlf, T. (2019). *Data visualisation with R: 111 examples*. Springer Nature.
- [47] Cheshire, J., & Lovelace, R. (2015). *Spatial data visualisation with R*. *Geocomputation: A Practical Primer*, 3.
- [48] Gesmann, M., & de Castillo, D. (2011). Using the Google Visualisation API with R. *R J.*, 3(2), 40.
- [49] <https://mazamascience.github.io/beakr/>
- [50] <https://www.rplumber.io/>
- [51] <https://github.com/devOpifex/ambiorix>
- [52] <https://cloud.r-project.org/web/packages/fiery/index.html>
- [53] Potter, G., Wong, J., Alcaraz, I., & Chi, P. (2016). Web application teaching tools for statistics using R and shiny. *Technology Innovations in Statistics Education*, 9(1).
- [54] Carmona, F., & Subirana, I. (2016). Shiny: aplicaciones web interactivas con R. In *VI GENAEIO 2015: VI Jornadas de Enseñanza y Aprendizaje de la Estadística y la Investigación Operativa: Huelva, del 25 al 26 de junio de 2015: Programa, comunicaciones* (pp. 21-42). Universidad de Huelva.
- [55] <https://arkiana.com/can-r-be-used-for-web-development/>
- [56] Зиганшин, И. А., & Валиуллина, Д. И. (2021). ОСОБЕННОСТИ И ПРЕИМУЩЕСТВА ИСПОЛЬЗОВАНИЯ ФРЕЙМВОРКА DJANGO В РАЗРАБОТКЕ ПОЛЬЗОВАТЕЛЬСКИХ САЙТОВ. In *ЛУЧШИЕ НАУЧНЫЕ ИССЛЕДОВАНИЯ 2021* (pp. 25-27).
- [57] Bilbija, Ž. (2019). IMPLEMENTACIJA SPOTIFY VEB KLIJENTA U REACT I DJANGO RADNOM OKRUŽENJU. *Zbornik radova Fakulteta tehničkih nauka u Novom Sadu*, 34(06), 1080-1083.
- [58] Насиров, Э. Ф., Кириллов, Д. С., & Чернова, М. В. (2019). ФРЕЙМВОРК DJANGO. In *СОВРЕМЕННОЕ ОБРАЗОВАНИЕ: АКТУАЛЬНЫЕ ВОПРОСЫ, ДОСТИЖЕНИЯ И ИННОВАЦИИ* (pp. 213-215).
- [59] Lerner, R. M. (2007). At the forge: First steps with Django. *Linux Journal*, 2007(159), 11.
- [60] Moore, J. (2016). *Building a reusable application with Django: Building a reusable application with Django*.
- [61] Ho, H. P. T. (2020). Leveraging web scraping for collecting competitive market data: Case: A case study of an Airbnb rental unit in Helsinki.
- [62] Палтаева, М. Н. (2020). ФРЕЙМВОРК FLASK. ОСОБЕННОСТИ И ВОЗМОЖНОСТИ. In *Лучшая студенческая статья 2020* (pp. 311-314).
- [63] Gaspar, D., & Stouffer, J. (2018). *Mastering Flask Web Development: Build Enterprise-grade, Scalable Python Web Applications*. Packt Publishing Ltd.
- [64] Ксендзовский, И. Д., Калужный, Е. Р., & Зариковская, Н. В. (2020). Разработка серверной части приложений и систем: технологии и языки программирования. In *European reserch* (pp. 30-32).
- [65] PUTRI, W. L. (2022). Aplikasi Model Prediksi Gaji Profesi Data Berbasis Website Menggunakan Flask Dan Metode Decision Tree.
- [66] Nicholus, R., & Nicholus, R. (2016). Understanding the Web API and “Vanilla” JavaScript. *Beyond jQuery*, 19-29.
- [67] Gackenheimer, C. (2015). *Introduction to React*. Apress.
- [68] Beighley, L. (2010). *jQuery for Dummies*. John Wiley & Sons.
- [69] Castillo, A. A. (2017). *Curso de Programación Web: JavaScript, Ajax y jQuery*. IT Campus Academy.
- [70] Li, J., & Peng, C. (2012, June). jQuery-based Ajax general interactive architecture. In *2012 IEEE International Conference on Computer Science and Automation Engineering* (pp. 304-306). IEEE.
- [71] Li, J., & Peng, C. (2012, June). jQuery-based Ajax general interactive architecture. In *2012 IEEE International Conference on Computer Science and Automation Engineering* (pp. 304-306). IEEE.
- [72] Ahmed, M. Z. (2014). Which one is better-JavaScript or jQuery. *International Journal of Computer Science and Mobile Computing*, 3(6), 193-207.

- [73] Drobyazkin, R. S., & Dmitrienko, N. A. (2021). COMPARISON OF ACTUAL JAVASCRIPT FRAMEWORKS FOR DEVELOPING WEB-APPLICATIONS. In *НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И СИСТЕМЫ В РЕШЕНИИ ЗАДАЧ ИННОВАЦИОННОГО РАЗВИТИЯ* (pp. 25-30).
- [74] van der Geest, J., & Ettema, M. (2011). Comparison of JavaScript libraries. 8th SC@ RUG 2010-2011, 107.
- [75] Chavare, N. N., & Nadargi, A. V. (2016). Performance Comparison and Evaluation of jQuery with AngularJS.
- [76] Gackenheim, C., & Gackenheim, C. (2015). What is react?. introduction to react, 1-20.
- [77] Boduch, A., & Derks, R. (2020). React and React Native: A complete hands-on guide to modern web and mobile development with React.js. Packt Publishing Ltd.
- [78] Król, K., & Szomorova, L. (2015). The possibilities of using chosen jQuery JavaScript components in creating interactive maps. *Geomatics, Landmanagement and Landscape*, (2), 45-54.
- [79] Parnin, C., & Treude, C. (2011, May). Measuring API documentation on the web. In *Proceedings of the 2nd international workshop on Web 2.0 for software engineering* (pp. 25-30).
- [80] Severance, C. (2015). John resig: Building jquery. *Computer*, 48(5), 7-8.
- [81] Soni, R. (2016). Nginx. Berkeley: Apress.
- [82] Sharma, R. (2015). NGINX high performance. Packt Publishing Ltd.
- [83] Lei, Z., Zhou, H., Ye, S., Hu, W., & Liu, G. P. (2020). Cost-effective server-side re-deployment for web-based online laboratories using NGINX reverse proxy. *IFAC-PapersOnLine*, 53(2), 17204-17209.
- [84] Chi, X., Liu, B., Niu, Q., & Wu, Q. (2012, July). Web load balance and cache optimization design based nginx under high-concurrency environment. In *2012 Third International Conference on Digital Manufacturing & Automation* (pp. 1029-1032). IEEE.
- [85] Butler, T. (2017). NGINX Cookbook. Packt Publishing Ltd.
- [86] Han, S., Marshall, S., Chun, B. G., & Ratnasamy, S. (2012). {MegaPipe}: A New Programming Interface for Scalable Network {I/O}. In *10th USENIX Symposium on Operating Systems Design and Implementation (OSDI 12)* (pp. 135-148).
- [87] Hu, Y., Nanda, A., & Yang, Q. (1999, February). Measurement, analysis and performance improvement of the Apache web server. In *1999 IEEE International Performance, Computing and Communications Conference (Cat. No. 99CH36305)* (pp. 261-267). IEEE.
- [88] Chandra, A. Y. (2019). Analisis performansi antara apache & nginx web server dalam menangani client request. *Jurnal Sistem dan Informatika (JSI)*, 14(1), 48-56.
- [89] Al-Saydali, J., & Al-Saydali, M. (2021). Performance comparison between Apache and NGINX under slow rate DoS attacks.
- [90] Jiang, M., Luo, X., Miu, T., Hu, S., & Rao, W. (2017, June). Are HTTP/2 servers ready yet?. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)* (pp. 1661-1671). IEEE.
- [91] Wang, J., & Kai, Z. (2021, June). Performance analysis and optimization of nginx-based web server. In *Journal of Physics: Conference Series (Vol. 1955, No. 1, p. 012033)*. IOP Publishing.
- [92] Wijeratne, W. V. D. P. (2016). Integrated Simple Intrusion Detection and Prevention System for Nginx Web Server (Doctoral dissertation).
- [93] Morton, M., Werner, J., Kintis, P., Snow, K., Antonakakis, M., Polychronakis, M., & Monroe, F. (2018, April). Security risks in asynchronous web servers: When performance optimizations amplify the impact of data-oriented attacks. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)* (pp. 167-182). IEEE.
- [94] Lui, H., & Yu, C. H. (2019). Security Lessons from Building a Back-End Service for Data Collection. In *Proceedings of the International Conference on Security and Management (SAM)* (pp. 103-106). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).
- [95] Fjordvald, M. B., & Nedelcu, C. (2018). Nginx HTTP Server: Harness the power of Nginx to make the most of your infrastructure and serve pages faster than ever before. Packt Publishing Ltd.
- [96] Łapiński, M. (2021). Web application project for the settlement of teaching hours conducted by university lecturers (Doctoral dissertation, Instytut Metrologii i Inżynierii Biomedycznej).
- [97] Supriya, M. H. Modular Microservice based GPU Utilization Manager with Gunicorn.
- [98] Brzyszczyk, D. (2018). Aplikacja internetowa do zarządzania pracownią fizyczną w oparciu o framework DJANGO.
- [99] Du, X., Du, Z., Wang, X., & Zhou, D. (2017). Social Web Application Development Based on Data Analysis and Data Mining.
- [100] Ramesh, A. Auto Code Generation For ML Deployment.
- [101] Мельников, М. О., & Игоница, Е. В. (2021). НАСТРОЙКА DEBIAN-СЕРВЕРА ДЛЯ РАЗРАБОТКИ PYTHON И DJANGO ВЕБ-ПРИЛОЖЕНИЙ. *Modern Science*, (2-2), 389-398.
- [102] Relan, K., & Relan, K. (2019). Deploying flask applications. *Building REST APIs with Flask: Create Python Web Services with MySQL*, 159-182.

- [103] Homssi, R., & Möller, J. (2023). Load Balancing In The Edge Cloud With Service Degradation: Combining Application-level Service Degradation With Load Balancing in a Kubernetes-based Edge Cloud.
- [104] Devine, K. D., Boman, E. G., Heaphy, R. T., Hendrickson, B. A., Teresco, J. D., Faik, J., ... & Gervasio, L. G. (2005). New challenges in dynamic load balancing. *Applied Numerical Mathematics*, 52(2-3), 133-152.
- [105] Loubser, N., & Loubser, N. (2021). Creating a RESTful API: Flask. *Software Engineering for Absolute Beginners: Your Guide to Creating Software Products*, 193-233.
- [106] Torquato, M., Maciel, P., Araujo, J., & Umesh, I. M. (2017, June). An approach to investigate aging symptoms and rejuvenation effectiveness on software systems. In *2017 12th Iberian conference on Information systems and technologies (CISTI)* (pp. 1-6). IEEE.
- [107] Olenčín, M., & Perhác, J. (2019, November). Automated configuration of a Linux web server security. In *2019 IEEE 15th International Scientific Conference on Informatics* (pp. 000491-000496). IEEE.
- [108] Chang, B. R., Tsai, H. F., Huang, C. F., Lin, Z. Y., & Chen, C. M. (2012). Fast access security on cloud computing: ubuntu enterprise server and cloud with face and fingerprint identification. In *Proceedings of the 2011 2nd International Congress on Computer Applications and Computational Science: Volume 1* (pp. 451-457). Springer Berlin Heidelberg.
- [109] Yunan, A. (2020). Optimization Of Server Computer Security Using The Port Knocking Method On Ubuntu Server 12.04 LTS. *Jurnal Inotera*, 5(1), 26-34.
- [110] Van Vugt, S. (2009). *Pro Ubuntu Server Administration* (Vol. 1). Apress.
- [111] Hill, B. M., Helmke, M., Graner, A., & Burger, C. (2011). *The official Ubuntu book: Official Ubuntu book The_6*. Pearson Education.
- [112] Chapelle, G. (2013). *The official Ubuntu book*, by Matthew Helmke and Amber Graner. *ACM SIGSOFT Software Engineering Notes*, 38(1), 54-55.
- [113] Haines, N., & Haines, N. (2015). *Ubuntu Releases. Beginning Ubuntu for Windows and Mac Users*, 203-204.
- [114] Voras, I., Mihaljević, B., Orlić, M., Pletikosa, M., Žagar, M., Pavić, T., ... & Tomić, S. (2011, May). Evaluating open-source cloud computing solutions. In *2011 Proceedings of the 34th International Convention MIPRO* (pp. 209-214). IEEE.
- [115] Treseangrat, K., Kolahi, S. S., & Sarrafpour, B. (2015, July). Analysis of UDP DDoS cyber flood attack and defense mechanisms on Windows Server 2012 and Linux Ubuntu 13. In *2015 International Conference on Computer, Information and Telecommunication Systems (CITS)* (pp. 1-5). IEEE.
- [116] Zebari, G. M. O., Faraj, K., & Zeebaree, S. (2016). Hand writing code-php or wire shark ready application over tier architecture with windows servers operating systems or linux server operating systems. *International Journal of Computer Sciences and Engineering*, 4, 142-149.
- [117] Saab, A., Nakhle, V., Chehade, G. A. H., & Al Moussawi, H. (2022). Testing and Comparing the Performances of Windows Server 2022, Ubuntu 20.04 and CentOS 8 under DDoS Attacks.
- [118] Nedelcu, C. (2010). *Nginx http server*. Packt Publishing Ltd..
- [119] Правоткин, И. А. (2020). Настройка и запуск программ на Python на удалённом хостинге. In *Приоритетные направления инновационной деятельности в промышленности* (pp. 78-80).
- [120] Murthy, M. M., Sanjay, H. A., & Ashwini, J. P. (2012, August). Pricing models and pricing schemes of IaaS providers: a comparison study. In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics* (pp. 143-147).
- [121] Aas, J., Barnes, R., Case, B., Durumeric, Z., Eckersley, P., Flores-López, A., ... & Warren, B. (2019, November). Let's Encrypt: an automated certificate authority to encrypt the entire web. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (pp. 2473-2487).
- [122] Aertsen, M., Korczyński, M., Moura, G. C., Tajalizadehkhooob, S., & Van Den Berg, J. (2017, July). No domain left behind: is Let's Encrypt democratizing encryption?. In *Proceedings of the applied networking research workshop* (pp. 48-54).
- [123] Tiefenau, C., von Zezschwitz, E., Häring, M., Krombholz, K., & Smith, M. (2019, November). A usability evaluation of Let's Encrypt and Certbot: usable security done right. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1971-1988).
- [124] Birge-Lee, H., Wang, L., McCarney, D., Shoemaker, R., Rexford, J., & Mittal, P. (2021). Experiences Deploying {Multi-Vantage-Point} Domain Validation at Let's Encrypt. In *30th usenix security symposium (usenix security 21)* (pp. 4311-4327).
- [125] Manousis, A., Ragsdale, R., Draffin, B., Agrawal, A., & Sekar, V. (2016). Shedding light on the adoption of let's encrypt. *arXiv preprint arXiv:1611.00469*.
- [126] Lukovenko, A. (2016). Let's automate let's encrypt. *Linux Journal*, 2016(266), 1.
- [127] Robinson, M. (2017). How to Get Free HTTPS from Let's Encrypt. *Journal of Intellectual Freedom & Privacy*, 2(1), 11-12.
- [128] Aertsen, M. (2019). How to bring HTTPS to the masses? Measuring issuance in the first year of Let's Encrypt.

- [129] Buchanan, W. J., Helme, S., & Woodward, A. (2018). Analysis of the adoption of security headers in HTTP. *IET Information Security*, 12(2), 118-126.
- [130] Eckersley, P., & Zhu, Y. (2016). Several Horror Stories about the Encrypted Web.
- [131] Sheth, C., & Thakker, R. (2011, February). Performance evaluation and comparative analysis of network firewalls. In 2011 International Conference on Devices and Communications (ICDeCom) (pp. 1-5). IEEE.
- [132] Kumar, S., & Gade, R. S. R. (2012). Experimental evaluation of Cisco ASA-5510 intrusion prevention system against denial of service attacks.
- [133] Xu, J., & Su, W. (2013). Performance evaluations of Cisco ASA and linux IPTables firewall solutions.
- [134] Hayajneh, T., Mohd, B. J., Itradat, A., & Quttoum, A. N. (2013). Performance and information security evaluation with firewalls. *International Journal of Security and Its Applications*, 7(6), 355-372.
- [135] Sheth, C., & Thakker, R. (2013). Performance Evaluation and Comparison of Network Firewalls under DDoS Attack. *International Journal of Computer Network & Information Security*, 5(12).
- [136] Alsharabi, N., Alqunun, M., & Murshed, B. A. H. (2023). Detecting Unusual Activities in Local Network Using Snort and Wireshark Tools. *Journal of Advances in Information Technology*, 14(4).
- [137] Saboor, A., Akhlaq, M., & Aslam, B. (2013, December). Experimental evaluation of Snort against DDoS attacks under different hardware configurations. In 2013 2nd National Conference on Information Assurance (NCIA) (pp. 31-37). IEEE.
- [138] Abdulrezzak, S., & Sabir, F. (2023). An Empirical Investigation on Snort NIDS versus Supervised Machine Learning Classifiers. *Journal of Engineering*, 29(2), 164-178.
- [139] Waili, A. R. A. (2023). Analysis of Traffic Using the Snort Tool for the Detection of Malware Traffic. *International Journal of Information Technology & Computer Engineering (IJITC) ISSN: 2455-5290*, 3(03), 30-37.
- [140] Stegeby, P. (2023). Intrusion Detection systems: A comparison in configuration and implementation between OSSEC and Snort.
- [141] Lamshöft, K., Neubert, T., Hielscher, J., Vielhauer, C., & Dittmann, J. (2022). Knock, knock, log: Threat analysis, detection & mitigation of covert channels in syslog using port scans as cover. *Forensic Science International: Digital Investigation*, 40, 301335.
- [142] Taib, A. M., Abdullah, A. A. S., Ariffin, M. A. M., & Ruslan, R. (2022, October). Threats and Vulnerabilities Handling via Dual-stack Sandboxing Based on Security Mechanisms Model. In 2022 IEEE 12th International Conference on Control System, Computing and Engineering (ICCSCE) (pp. 113-118). IEEE.
- [143] Godinho, A., Rosado, J., Sá, F., Caldeira, F., & Cardoso, F. (2022). Torrent Poisoning Protection with a Reverse Proxy Server. *Electronics*, 12(1), 165.
- [144] Handa, A., Negi, R., & Shukla, S. K. (Eds.). (2022). *Implementing Enterprise Cybersecurity with Opensource Software and Standard Architecture*. CRC Press.
- [145] Киздермишов, А. А., & Киздермишова, С. Х. (2023). ОТЕЧЕСТВЕННЫЙ ИНТЕРНЕТ-ШЛЮЗ ДЛЯ НЕБОЛЬШОГО ОФИСА. *Вестник Адыгейского государственного университета. Серия 4: Естественно-математические и технические науки*, (1 (316)), 52-58.
- [146] Keller, M., & Nussbaumer, M. (2010, September). CSS code quality: a metric for abstractness; or why humans beat machines in CSS coding. In 2010 Seventh International Conference on the Quality of Information and Communications Technology (pp. 116-121). IEEE.
- [147] Baker, S. C. (2014). Making it work for everyone: HTML5 and CSS level 3 for responsive, accessible design on your library's web site. *Journal of Library & Information Services in Distance Learning*, 8(3-4), 118-136.
- [148] Frain, B. (2020). *Responsive Web Design with HTML5 and CSS: Develop future-proof responsive websites using the latest HTML5 and CSS techniques*. Packt Publishing Ltd.
- [149] Schifreen, R. (2009). *How to create Web sites and applications with HTML, CSS, Javascript, PHP and MySQL*.
- [150] Cederholm, D. (2009). *Applying CSS. Web Standards Solutions: The Markup and Style Handbook*, 149-165.
- [151] Casario, M., Wormser, N., Saltzman, D., Bradford, A., Reid, J., Improta, F., ... & Congleton, A. (2012). *CSS Basics. CSS3 Solutions: Essential Techniques for CSS3 Developers*, 1-24.
- [152] Kennedy, A., de León, I., Kennedy, A., & de León, I. (2011). *Creating Your CSS. Pro CSS for High Traffic Websites*, 331-360.
- [153] Naylor, L., & Naylor, L. (2016). *Restyling the web site: An introduction. ASP. NET MVC with Entity Framework and CSS*, 467-484.
- [154] Kennedy, A., de León, I., Kennedy, A., & de León, I. (2011). *CSS Style Guide. Pro CSS for High Traffic Websites*, 37-66.
- [155] <https://learn.jquery.com/using-jquery-core/document-ready/>

- [156] Keller, M. S. (1999). Take command: cron: Job scheduler. *Linux Journal*, 1999(65es), 15-es.
- [157] Reznick, L. (1993). Using cron and crontab. *Sys Admin*, 2(4), 29-32.
- [158] Panda, S., & Ramani, S. (2013). Protection of web application against SQL injection attacks. *International Journal of Modern Engineering Research (IJMER)*, 3(1), 166-168.
- [159] Mohanty, S., Acharya, A. A., Mishra, D. B., & Panda, N. (2019). Security Testing of Web Applications Using Threat Modeling: A Systematic Review. *IJCSMC International Journal of Computer Science and Mobile Computing*, 8(1), 50-57.
- [160] Shar, L. K., & Tan, H. B. K. (2012). Defeating SQL injection. *Computer*, 46(3), 69-77.
- [161] Gupta, S., & Gupta, B. B. (2017). Cross-Site Scripting (XSS) attacks and defense mechanisms: classification and state-of-the-art. *International Journal of System Assurance Engineering and Management*, 8, 512-530.
- [162] Hydera, I., Sultan, A. B. M., Zulzalil, H., & Admodisastro, N. (2015). Current state of research on cross-site scripting (XSS)—A systematic literature review. *Information and Software Technology*, 58, 170-186.
- [163] Sarmah, U., Bhattacharyya, D. K., & Kalita, J. K. (2018). A survey of detection methods for XSS attacks. *Journal of Network and Computer Applications*, 118, 113-143.
- [164] Grossman, J. (2007). *XSS attacks: cross site scripting exploits and defense*. Syngress.
- [165] Klein, A. (2005). DOM based cross site scripting or XSS of the third kind. *Web Application Security Consortium, Articles*, 4, 365-372.
- [166] Dayanandam, G., Rao, T. V., Bujji Babu, D., & Nalini Durga, S. (2019). DDoS attacks—analysis and prevention. In *Innovations in Computer Science and Engineering: Proceedings of the Fifth ICICSE 2017* (pp. 1-10). Springer Singapore.
- [167] Bhattacharyya, D. K., & Kalita, J. K. (2016). *DDoS attacks: evolution, detection, prevention, reaction, and tolerance*. CRC Press.
- [168] Zhang, B., Zhang, T., & Yu, Z. (2017, December). DDoS detection and prevention based on artificial intelligence techniques. In *2017 3rd IEEE International Conference on Computer and Communications (ICCC)* (pp. 1276-1280). IEEE.
- [169] Badotra, S., & Panda, S. N. (2021). SNORT based early DDoS detection system using Opendaylight and open networking operating system in software defined networking. *Cluster Computing*, 24, 501-513.
- [170] Saboor, A., Akhlaq, M., & Aslam, B. (2013, December). Experimental evaluation of Snort against DDoS attacks under different hardware configurations. In *2013 2nd National Conference on Information Assurance (NCIA)* (pp. 31-37). IEEE.
- [171] Mesbah, A., & Prasad, M. R. (2011, May). Automated cross-browser compatibility testing. In *Proceedings of the 33rd International Conference on Software Engineering* (pp. 561-570).
- [172] Xu, L., Xu, B., Nie, C., Chen, H., & Yang, H. (2003, June). A browser compatibility testing method based on combinatorial testing. In *International Conference on Web Engineering* (pp. 310-313). Berlin, Heidelberg: Springer Berlin Heidelberg.
- [173] Sabaren, L. N., Mascheroni, M. A., Greiner, C. L., & Irrazábal, E. (2018). A systematic literature review in cross-browser testing. *Journal of Computer Science & Technology*, 18.
- [174] Li, X., & Zeng, H. (2014, June). Modeling web application for cross-browser compatibility testing. In *15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)* (pp. 1-5). IEEE.
- [175] Shi, H., & Zeng, H. (2015, August). Cross-browser compatibility testing based on model comparison. In *2015 International Conference on Computer Application Technologies* (pp. 103-107). IEEE.
- [176] Xu, S., & Zeng, H. (2015, July). Static analysis technique of cross-browser compatibility detecting. In *2015 3rd International Conference on Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence* (pp. 103-107). IEEE.
- [177] Prandl, S., Lazarescu, M., & Pham, D. S. (2015). A study of web application firewall solutions. In *Information Systems Security: 11th International Conference, ICISS 2015, Kolkata, India, December 16-20, 2015. Proceedings 11* (pp. 501-510). Springer International Publishing.
- [178] Torrano-Gimenez, C., Perez-Villegas, A., & Alvarez, G. (2009). A self-learning anomaly-based web application firewall. In *Computational Intelligence in Security for Information Systems: CISIS'09, 2nd International Workshop Burgos, Spain, September 2009 Proceedings* (pp. 85-92). Springer Berlin Heidelberg.
- [179] Muzaki, R. A., Briliyant, O. C., Hasditama, M. A., & Ritchi, H. (2020, October). Improving Security of Web-Based Application Using ModSecurity and Reverse Proxy in Web Application Firewall. In *2020 International Workshop on Big Data and Information Security (IWBIS)* (pp. 85-90). IEEE.
- [180] El-Atawy, A., Ibrahim, K., Hamed, H., & Al-Shaer, E. (2005, November). Policy segmentation for intelligent firewall testing. In *1st IEEE ICNP Workshop on Secure Network Protocols, 2005.(NPSec)*. (pp. 67-72). IEEE.

- [181] Lyu, M. R., & Lau, L. K. (2000, October). Firewall security: Policies, testing and performance evaluation. In Proceedings 24th Annual International Computer Software and Applications Conference. COMPSAC2000 (pp. 116-121). IEEE.
- [182] Senn, D., Basin, D., & Caronni, G. (2005, May). Firewall conformance testing. In IFIP International Conference on Testing of Communicating Systems (pp. 226-241). Berlin, Heidelberg: Springer Berlin Heidelberg.
- [183] Brubaker, C., Jana, S., Ray, B., Khurshid, S., & Shmatikov, V. (2014, May). Using frankencerts for automated adversarial testing of certificate validation in SSL/TLS implementations. In 2014 IEEE Symposium on Security and Privacy (pp. 114-129). IEEE.
- [184] Chen, Y., & Su, Z. (2015, August). Guided differential testing of certificate validation in SSL/TLS implementations. In Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering (pp. 793-804).
- [185] Tian, C., Chen, C., Duan, Z., & Zhao, L. (2019). Differential testing of certificate validation in SSL/TLS implementations: an RFC-guided approach. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 28(4), 1-37.
- [186] De Vries, S. (2006). Security testing web applications throughout automated software tests. In OWASP Europe Conference (Vol. 1, No. 1, pp. 1-13).
- [187] Scott Jr, E. A. (2015). *SPA Design and Architecture: Understanding single-page web applications*. Simon and Schuster.
- [188] Mills, D. L. (2008). Testing for Web Applications. In *Software Engineering for Modern Web Applications: Methodologies and Technologies* (pp. 207-216). IGI Global.
- [189] Tvoroshenko, I. S., & Maksimenko, H. (2021). To the question of analysis of existing mechanisms of web application testing.
- [190] Vala, R., & Jasek, R. (2011). Security testing of web applications. *Proceedings of the Annals of DAAAM and Proceedings*, 1533-1535.
- [191] Saia, S. M., Nelson, N. G., Young, S. N., Parham, S., & Vandegrift, M. (2022). Ten simple rules for researchers who want to develop web apps. *PLoS Computational Biology*, 18(1), e1009663.
- [192] Nikfard, P., Zadeh, M. H. A., & Ibrahim, S. B. (2013). A comparative evaluation of approaches for web application testing. In *International Conference on Soft Computing and Software Engineering (Vol. 2013)*.
- [193] <https://smartbear.com/product/bitbar/>
- [194] Madhusudhan, R., & Surashe, S. V. (2022, March). Privacy and Security Comparison of Web Browsers: A Review. In *International Conference on Advanced Information Networking and Applications* (pp. 459-470). Cham: Springer International Publishing.
- [195] Tewari, N., & Datt, G. (2021, November). A study on the systematic review of security vulnerabilities of popular web browsers. In *2021 International Conference on Technological Advancements and Innovations (ICTAI)* (pp. 314-318). IEEE.
- [196] Batishchev, A. M. (2004). LOIC (Low Orbit Ion Cannon). [online] <http://sourceforge.net/projects/loic>.
- [197] Cheepborisuttikul, T., & Teng-amnuay, Y. (2013). Using Low Orbit Ion Cannon for Denial of Service Attack Based on CVE. In *Proceedings of the Second International Conference on Advances in Information Technology-AIT 2013* (pp. 145-149).
- [197] Arunadevi, R. (2018). Experimentation Of Denial Of Service Attack In Wireless Local Area Infrastructure Network Using Loic Tool. *Journal of Engineering Research and Application*, 51-55.
- [198] Nagpal, B., Sharma, P., Chauhan, N., & Panesar, A. (2015, March). DDoS tools: Classification, analysis and comparison. In *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 342-346). IEEE.
- [199] Nardi, T. (2012). Low orbit ion cannon: Exposed. *The Power Base*, 4.
- [200] Moses, A. (2010). The Aussie Who Blitzed Visa, MasterCard and PayPal with the Low Orbit Ion Cannon. *The Age*. December, 9.
- [201] <https://statystyka.policja.pl/st/kodeks-karny/przestepstwa-przeciwno-14/63625,naruszenie-tajemnicy-korespondencji-art-267.html>
- [202] <https://www.ssllabs.com/ssltest/analyze.html?d=reduce.prz.edu.pl>
- [203] SOUSA, T. A. D. A. D. (2017). Estudo de caso: análise de certificado SSL de sites e-commerce utilizando Qualys SSLLABS.
- [204] Ristić, I. (2012). SSL/TLS deployment best practices. Internet: https://www.ssllabs.com/downloads/SSL_TLS_Deployment_Best_Practices_1, 3.
- [205] Akopov, M. S. (2018). *Securing your Site in Development and Beyond* (No. PNNL-SA-129189). Pacific Northwest National Lab.(PNNL), Richland, WA (United States).
- [206] Syahab, A. S. (2023). ANALISIS AUDIT KEAMANAN INFORMASI WEBSITE MENGGUNAKAN METODE NETWORK MAPPER DAN QUALYS SSL. *Jurnal Manajemen Informatika dan Sistem Informasi*, 6(1), 39-47.
- [207] Bairwa, A. K., & Joshi, S. (2021). Mutual authentication of nodes using session token with fingerprint and MAC address validation. *Egyptian Informatics Journal*, 22(4), 479-491.
- [208] Cheng, N., Wang, Y., Zhao, X., & Li, N. (2011, May). The digital fingerprint of xml electronic medical records based on hmac-sha256 algorithm. In *2011 IEEE 3rd International Conference on Communication Software and Networks* (pp. 338-340). IEEE.

- [209] Briliyant, O. C., & Baihaqi, A. (2017, October). Implementation of RSA 2048-bit and AES 128-bit for Secure e-learning web-based application. In 2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA) (pp. 1-5). IEEE.
- [210] Yilek, S., Rescorla, E., Shacham, H., Enright, B., & Savage, S. (2009, November). When private keys are public: Results from the 2008 Debian OpenSSL vulnerability. In Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement (pp. 15-27).
- [211] <https://letsencrypt.org/certificates/>
- [212] Santesson, S., & Housley, R. (2005). Internet X. 509 public key infrastructure authority information access certificate revocation list (CRL) extension (No. rfc4325).
- [213] Krawczyk, H., & Wee, H. (2016, March). The OPTLS protocol and TLS 1.3. In 2016 IEEE European Symposium on Security and Privacy (EuroS&P) (pp. 81-96). IEEE.
- [214] Tran, D. D., & Ogata, K. (2022). Formal verification of TLS 1.2 by automatically generating proof scores. *Computers & Security*, 123, 102909.
- [215] Sury, O. (2012). Use of the SHA-256 Algorithm With RSA, Digital Signature Algorithm (DSA), and Elliptic Curve DSA (ECDSA) in SSHFP Resource Records (No. rfc6594).
- [216] Adeniyi, E. A., Falola, P. B., Maashi, M. S., Aljebreen, M., & Bharany, S. (2022). Secure sensitive data sharing using RSA and ElGamal cryptographic algorithms with hash functions. *Information*, 13(10), 442.
- [217] Laurie, B. (2014). Certificate transparency. *Communications of the ACM*, 57(10), 40-46.
- [218] Biddle, R., Van Oorschot, P. C., Patrick, A. S., Sobey, J., & Whalen, T. (2009, November). Browser interfaces and extended validation SSL certificates: an empirical study. In Proceedings of the 2009 ACM workshop on Cloud computing security (pp. 19-30).
- [219] Chung, T., Lok, J., Chandrasekaran, B., Choffnes, D., Levin, D., Maggs, B. M., ... & Wilson, C. (2018, October). Is the web ready for ocsrp must-staple?. In Proceedings of the Internet Measurement Conference 2018 (pp. 105-118).
- [220] Myers, M., Ankney, R., Malpani, A., Galperin, S., & Adams, C. (1999). X. 509 Internet public key infrastructure online certificate status protocol-OCSP (No. rfc2560).
- [221] Scheitle, Q., Chung, T., Hiller, J., Gasser, O., Naab, J., van Rijswijk-Deij, R., ... & Carle, G. (2018). A first look at certification authority authorization (CAA). *ACM SIGCOMM Computer Communication Review*, 48(2), 10-23.
- [222] Hiller, J., Amann, J., & Hohlfeld, O. (2020, October). The boon and bane of cross-signing: Shedding light on a common practice in public key infrastructures. In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (pp. 1289-1306).
- [223] Cooper, M., Dzambasow, Y., Hesse, P., Joseph, S., & Nicholas, R. (2005). Internet X. 509 public key infrastructure: Certification path building (No. rfc4158).
- [224] Surani, A., Bawaked, A., Wheeler, M., Kelsey, B., Roberts, N., Vincent, D., & Das, S. (2023, July). Security and Privacy of Digital Mental Health: An Analysis of Web Services and Mobile Applications. In IFIP Annual Conference on Data and Applications Security and Privacy (pp. 319-338). Cham: Springer Nature Switzerland.
- [225] Dahlmanns, M., Lohmöller, J., Pennekamp, J., Bodenhausen, J., Wehrle, K., & Henze, M. (2022, May). Missed opportunities: Measuring the untapped TLS support in the industrial Internet of Things. In Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security (pp. 252-266).
- [226] Schwenk, J. (2022). A Short History of TLS. In *Guide to Internet Cryptography: Security Protocols and Real-World Attack Implications* (pp. 243-265). Cham: Springer International Publishing.
- [227] Luxemburk, J., & Čejka, T. (2023). Fine-grained TLS services classification with reject option. *Computer Networks*, 220, 109467.
- [228] Benjamin, D., & Wood, C. A. (2022). RFC 9258 Importing External Pre-Shared Keys (PSKs) for TLS.
- [229] Sakka, S., Liagkou, V., & Stylios, C. (2023). Exploiting Security Issues in Human Activity Recognition Systems (HARSs). *Information*, 14(6), 315.
- [230] Nainar, N. K., & Panda, A. (2022). Capturing Secured Application Traffic for Analysis. In *Wireshark for Network Forensics: An Essential Guide for IT and Cloud Professionals* (pp. 65-105). Berkeley, CA: Apress.
- [231] Birghan, M., & van der Merwe, T. (2022, May). A Client-Side Seat to TLS Deployment. In 2022 IEEE Security and Privacy Workshops (SPW) (pp. 13-19). IEEE.
- [232] Petersson, L., & Lindkvist, R. (2022). The Everyday Internet, a Minefield in Disguise: Characterization of different types of domains including malicious and popularity.

- [233] Dahlmanns, M., Lohmöller, J., Pennekamp, J., Bodenhausen, J., Wehrle, K., & Henze, M. (2022, May). Missed opportunities: Measuring the untapped TLS support in the industrial Internet of Things. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security* (pp. 252-266).
- [234] Oppliger, R. (2023). *SSL and TLS: Theory and Practice*. Artech House.
- [235] Alhanahnah, M., & Yan, Q. (2018, April). Towards best secure coding practice for implementing SSL/TLS. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (pp. 1-6). IEEE.
- [236] Dworkin, M. J., Barker, E. B., Nechvatal, J. R., Foti, J., Bassham, L. E., Roback, E., & Dray Jr, J. F. (2001). Advanced encryption standard (AES).
- [237] De Santis, F., Schauer, A., & Sigl, G. (2017, March). ChaCha20-Poly1305 authenticated encryption for high-speed embedded IoT applications. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017* (pp. 692-697). IEEE.
- [238] Itoh, K., Izu, T., & Takenaka, M. (2003). Address-bit differential power analysis of cryptographic schemes OK-ECDH and OK-ECDSA. In *Cryptographic Hardware and Embedded Systems-CHES 2002: 4th International Workshop Redwood Shores, CA, USA, August 13-15, 2002 Revised Papers 4* (pp. 129-143). Springer Berlin Heidelberg.
- [239] Hanaoka, G., & Kurosawa, K. (2010). Between hashed DH and computational DH: Compact encryption from weaker assumption. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 93(11), 1994-2006.
- [240] Fadaei, T., Schrittwieser, S., Kieseberg, P., & Mulazzani, M. (2015, August). Trust me, I'm a Root CA! Analyzing SSL Root CAs in Modern Browsers and Operating Systems. In *2015 10th International Conference on Availability, Reliability and Security* (pp. 174-179). IEEE.
- [241] Bhargavan, K., Fournet, C., Kohlweiss, M., Pironti, A., Strub, P. Y., & Zanella-Béguelin, S. (2014). Proving the TLS handshake secure (as it is). In *Advances in Cryptology-CRYPTO 2014: 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II 34* (pp. 235-255). Springer Berlin Heidelberg.
- [242] Santesson, S. (2006). *TLS Handshake Message for Supplemental Data* (No. rfc4680).
- [243] Bhargavan, K., Lavaud, A. D., Fournet, C., Pironti, A., & Strub, P. Y. (2014, May). Triple handshakes and cookie cutters: Breaking and fixing authentication over TLS. In *2014 IEEE Symposium on Security and Privacy* (pp. 98-113). IEEE.
- [244] Morrissey, P., Smart, N. P., & Warinschi, B. (2008). A modular security analysis of the TLS handshake protocol. In *Advances in Cryptology-ASIACRYPT 2008: 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, December 7-11, 2008. Proceedings 14* (pp. 55-73). Springer Berlin Heidelberg.
- [245] Stebila, D., & Sullivan, N. (2015, August). An analysis of TLS handshake proxying. In *2015 IEEE Trustcom/BigDataSE/ISPA (Vol. 1, pp. 279-286)*. IEEE.
- [246] Adrian, D., Bhargavan, K., Durumeric, Z., Gaudry, P., Green, M., Halderman, J. A., ... & Zimmermann, P. (2015, October). Imperfect forward secrecy: How Diffie-Hellman fails in practice. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (pp. 5-17).
- [247] Park, D., Boyd, C., & Moon, S. J. (2000). Forward secrecy and its application to future mobile communications security. In *Public Key Cryptography: Third International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2000, Melbourne, Victoria, Australia, January 18-20, 2000. Proceedings 3* (pp. 433-445). Springer Berlin Heidelberg.
- [248] Huang, L. S., Adhikarla, S., Boneh, D., & Jackson, C. (2014). An experimental study of TLS forward secrecy deployments. *IEEE Internet Computing*, 18(6), 43-51.
- [249] Aviram, N., Schinzel, S., Somorovsky, J., Heninger, N., Dankel, M., Steube, J., ... & Shavitt, Y. (2016). {DROWN}: Breaking {TLS} Using {SSLv2}. In *25th USENIX Security Symposium (USENIX Security 16)* (pp. 689-706).
- [250] Ivanov, O., Ruzhentsev, V., & Oliynykov, R. (2018, October). Comparison of modern network attacks on TLS protocol. In *2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T)* (pp. 565-570). IEEE.
- [251] Giesen, F., Kohlar, F., & Stebila, D. (2013, November). On the security of TLS renegotiation. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security* (pp. 387-398).
- [252] Rescorla, E., Ray, M., Dispensa, S., & Oskov, N. (2010). Transport layer security (TLS) renegotiation indication extension (No. rfc5746).
- [253] Ray, M., & Dispensa, S. (2009). Renegotiating tls.
- [254] Sarkar, P. G., & Fitzgerald, S. (2013). Attacks on ssl a comprehensive study of beast, crime, time, breach, lucky 13 & rc4 biases. Internet: https://www.isecpartners.com/media/106031/ssl_attacks_survey.pdf [June, 2014].
- [255] Möller, B., Duong, T., & Kotowicz, K. (2014). This POODLE bites: exploiting the SSL 3.0 fallback. *Security Advisory*, 21, 34-58.
- [256] Fogel, B., Farmer, S., Alkofahi, H., Skjellum, A., & Hafiz, M. (2016). POODLEs, more POODLEs, FREAK attacks too: how server administrators responded to three serious web vulnerabilities. In *Engineering Secure Software and Systems: 8th*

- International Symposium, ESSoS 2016, London, UK, April 6–8, 2016. Proceedings 8 (pp. 122-137). Springer International Publishing.
- [257] Stebila, D. Attacks on TLS.
- [258] Budihardjo, E. W., Dewi, L. P., & Noertjahyana, A. (2021). Pembuatan Konfigurasi SSL yang Aman untuk Diimplementasikan pada Apache dan Nginx. *Jurnal Infra*, 9(2), 114-119.
- [259] Georgiev, M., Iyengar, S., Jana, S., Anubhai, R., Boneh, D., & Shmatikov, V. (2012, October). The most dangerous code in the world: validating SSL certificates in non-browser software. In *Proceedings of the 2012 ACM conference on Computer and communications security* (pp. 38-49).
- [260] Hlavacek, T., Jeitner, P., Mirdita, D., Shulman, H., & Waidner, M. (2022). Stalloris: {RPKI} Downgrade Attack. In *31st USENIX Security Symposium (USENIX Security 22)* (pp. 4455-4471).
- [261] Möller, B., & Langley, A. (2015). TLS fallback Signaling Cipher Suite Value (SCSV) for preventing protocol downgrade attacks (No. rfc7507).
- [262] CHEN, Yue, et al. Downgrade attack on trustzone. arXiv preprint arXiv:1707.05082, 2017.
- [263] Satapathy, A., & Livingston, J. (2016). A Comprehensive Survey on SSL/TLS and their Vulnerabilities. *International Journal of Computer Applications*, 153(5), 31-38.
- [264] Okamoto, N., Kimura, S., & Ebihara, Y. (2003, March). An introduction of compression algorithms into SSL/TLS and proposal of compression algorithms specialized for application. In *17th International Conference on Advanced Information Networking and Applications, 2003. AINA 2003.* (pp. 817-820). IEEE.
- [265] Eldewahi, A. E., Sharfi, T. M., Mansor, A. A., Mohamed, N. A., & Alwabhani, S. M. (2015, September). Ssl/tls attacks: Analysis and evaluation. In *2015 International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE)* (pp. 203-208). IEEE.
- [266] AlFardan, N., Bernstein, D. J., Paterson, K. G., Poettering, B., & Schuldt, J. C. (2013). On the security of {RC4} in {TLS}. In *22nd USENIX Security Symposium (USENIX Security 13)* (pp. 305-320).
- [267] AlFardan, N. J., Bernstein, D. J., Paterson, K. G., Poettering, B., & Schuldt, J. C. (2013, August). On the security of RC4 in TLS and WPA. In *USENIX Security Symposium (Vol. 173)*.
- [268] Popov, A. (2015). Prohibiting RC4 cipher suites (No. rfc7465).
- [269] Vanhoef, M., & Piessens, F. (2015). All Your Biases Belong to Us: Breaking {RC4} in {WPA-TKIP} and {TLS}. In *24th USENIX Security Symposium (USENIX Security 15)* (pp. 97-112).
- [270] Garman, C., Paterson, K. G., & Van der Merwe, T. (2015). Attacks Only Get Better: Password Recovery Attacks Against {RC4} in {TLS}. In *24th USENIX Security Symposium (USENIX Security 15)* (pp. 113-128).
- [271] Seggelmann, R., Tuexen, M., & Williams, M. (2012). Transport layer security (TLS) and datagram transport layer security (DTLS) heartbeat extension (No. rfc6520).
- [272] Berbecaru, D. G., & Petraglia, G. (2023, January). TLS-Monitor: A Monitor for TLS Attacks. In *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)* (pp. 1-6). IEEE.
- [273] Durumeric, Z., Li, F., Kasten, J., Amann, J., Beekman, J., Payer, M., ... & Halderman, J. A. (2014, November). The matter of heartbleed. In *Proceedings of the 2014 conference on internet measurement conference* (pp. 475-488).
- [274] Wheeler, D. A. (2014). Preventing heartbleed. *Computer*, 47(08), 80-83.
- [275] Zhang, L., Choffnes, D., Levin, D., Dumitras, T., Mislove, A., Schulman, A., & Wilson, C. (2014, November). Analysis of SSL certificate reissues and revocations in the wake of Heartbleed. In *Proceedings of the 2014 Conference on Internet Measurement Conference* (pp. 489-502).
- [276] Lee, J., & Wallach, D. S. (2018). Removing Secrets from Android's TLS. In *NDSS*.
- [277] Keerthi, V. K. (2016). Taxonomy of SSL/TLS attacks. *International Journal of Computer Network and Information Security*, 8(2), 15.
- [278] Durieux, T. (2014). Web Application Security Detection of Misconfigurations in CERN servers (No. CERN-STUDENTS-Note-2014-079).
- [279] Widjarto, A., Lubis, M., & Ayuningtyas, V. (2021). Vulnerability and risk assessment for operating system (OS) with framework STRIDE: Comparison between VulnOS and Vulnix. *Ind. J. of Electrical and Com. Science*, 23(3), 1643-1653.
- [280] Aranha, D. F., Novaes, F. R., Takahashi, A., Tibouchi, M., & Yarom, Y. Breaking ECDSA with Less than One Bit of Nonce Leakage *ACM CCS'20*.
- [281] Merget, R., Somorovsky, J., Aviram, N., Young, C., Fliegenschmidt, J., Schwenk, J., & Shavitt, Y. (2019). Scalable scanning and automatic classification of {TLS} padding oracle vulnerabilities. In *28th USENIX Security Symposium (USENIX Security 19)* (pp. 1029-1046).

- [282] Paterson, K. G., & Watson, G. J. (2008, September). Immunising CBC mode against padding oracle attacks: A formal security treatment. In *International Conference on Security and Cryptography for Networks* (pp. 340-357). Berlin, Heidelberg: Springer Berlin Heidelberg.
- [283] Rizzo, J., & Duong, T. (2010). Practical padding oracle attacks. In *4th USENIX Workshop on Offensive Technologies (WOOT 10)*.
- [284] Mu, C., Yang, M., Chen, Z., & Wang, B. (2020, June). Research on RSA padding identification method in IoT firmwares. In *Journal of Physics: Conference Series* (Vol. 1570, No. 1, p. 012061). IOP Publishing.
- [285] Böck, H., Somorovsky, J., & Young, C. (2018). Return Of {Bleichenbacher's} Oracle Threat ({{{ROBOT}}}). In *27th USENIX Security Symposium (USENIX Security 18)* (pp. 817-849).
- [286] Friedl, S., Popov, A., Langley, A., & Stephan, E. (2014). Transport layer security (tls) application-layer protocol negotiation extension (No. rfc7301).
- [287] Shoemaker, R. B. (2020). RFC 8737: Automated Certificate Management Environment (ACME) TLS Application-Layer Protocol Negotiation (ALPN) Challenge Extension.
- [288] Brinkmann, M., Dresen, C., Merget, R., Poddebniak, D., Müller, J., Somorovsky, J., ... & Schinzel, S. (2021). {ALPACA}: Application layer protocol confusion-analyzing and mitigating cracks in {TLS} authentication. In *30th USENIX Security Symposium (USENIX Security 21)* (pp. 4293-4310).
- [289] Rescorla, E. The Road to TLS 1.3.
- [290] Salowey, J., Zhou, H., Eronen, P., & Tschofenig, H. (2008). Transport layer security (TLS) session resumption without server-side state (No. rfc5077).
- [291] Sy, E., Burkert, C., Federrath, H., & Fischer, M. (2018, December). Tracking users across the web via TLS session resumption. In *Proceedings of the 34th Annual Computer Security Applications Conference* (pp. 289-299).
- [292] Aviram, N., Gellert, K., & Jager, T. (2021). Session resumption protocols and efficient forward security for TLS 1.3 0-RTT. *Journal of Cryptology*, 34(3), 20.
- [293] Tange, K., Mödersheim, S., Lalos, A., Fafoutis, X., & Dragoni, N. (2021). rTLS: Secure and efficient TLS session resumption for the Internet of Things. *Sensors*, 21(19), 6524.
- [294] Arfaoui, G., Bultel, X., Fouque, P. A., Nedelcu, A., & Onete, C. (2019). The privacy of the TLS 1.3 protocol. *Cryptology ePrint Archive*.
- [295] AbdelKhalek, Y. (2023). Incorporating OCSP Stapling in EDHOC for Certificate Revocation in Resource Constrained Environments (Doctoral dissertation, Technische Universität Berlin).
- [296] Kim, E. G. OCSP modification for fast processing of stapled message.
- [297] Hodges, J., Jackson, C., & Barth, A. (2012). Http strict transport security (hsts) (No. rfc6797).
- [298] Selvi, J. (2014). Bypassing HTTP strict transport security. *Black Hat Europe*, 54.
- [299] Dolnák, I., & Litvik, J. (2017, October). Introduction to HTTP security headers and implementation of HTTP strict transport security (HSTS) header for HTTPS enforcing. In *2017 15th International Conference on Emerging eLearning Technologies and Applications (ICETA)* (pp. 1-4). IEEE.
- [300] Amiruddin, A., Yusa, D. A. P., & Rofiq, R. A. (2021, October). Conformity Analysis of HTTP Strict Transport Security (HSTS) Configuration and Implementation Using Bettercap Tools. In *2021 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)* (pp. 13-18). IEEE.
- [301] Evans, C., Palmer, C., & Sleevi, R. (2015). Public key pinning extension for HTTP (No. rfc7469).
- [302] Toth, G., & Vlieg, T. (2013). Public Key Pinning for TLS Using a Trust on First Use Model.
- [303] Kim, T. H. J., Huang, L. S., Perrig, A., Jackson, C., & Gligor, V. (2013, May). Accountable key infrastructure (AKI) a proposal for a public-key validation infrastructure. In *Proceedings of the 22nd international conference on World Wide Web* (pp. 679-690).
- [304] Oppliger, R. (2013). Certification authorities under attack: A plea for certificate legitimation. *IEEE internet computing*, 18(1), 40-47.
- [305] Gillmor, D. (2016). Negotiated finite field Diffie-Hellman ephemeral parameters for transport layer security (TLS) (No. rfc7919).
- [306] Akhawe, D., Amann, J., Vallentin, M., & Sommer, R. (2013, May). Here's my cert, so trust me, maybe? Understanding TLS errors on the web. In *Proceedings of the 22nd international conference on World Wide Web* (pp. 59-70).
- [307] Huawei, Z., & Ruixia, L. (2009, May). A scheme to improve security of SSL. In *2009 Pacific-Asia Conference on Circuits, Communications and Systems* (pp. 401-404). IEEE.
- [308] Cremers, C., Horvat, M., Scott, S., & van der Merwe, T. (2016, May). Automated analysis and verification of TLS 1.3: 0-RTT, resumption and delayed authentication. In *2016 IEEE Symposium on Security and Privacy (SP)* (pp. 470-485). IEEE.

- [309] Hale, B., Jager, T., Lauer, S., & Schwenk, J. (2017). Simple security definitions for and constructions of 0-RTT key exchange. In *Applied Cryptography and Network Security: 15th International Conference, ACNS 2017, Kanazawa, Japan, July 10-12, 2017, Proceedings 15* (pp. 20-38). Springer International Publishing.
- [310] Cao, X., Zhao, S., & Zhang, Y. (2019, December). 0-rtt attack and defense of quic protocol. In *2019 IEEE Globecom Workshops (GC Wkshps)* (pp. 1-6). IEEE.
- [311] <https://gtmetrix.com/>
- [312] Fryonanda, H., & Ahmad, T. (2017). Analisis Website Perguruan Tinggi Berdasarkan Keinginan Search Engine Menggunakan Automated Software Testing GTmetrix. *Kalbiscientia: Jurnal Sains dan Teknologi*, 4(2), 179-183.
- [313] Dawis, A. M., & Setiawan, I. (2022). Evaluation of The Website 'Aisyiyah Surakarta of University Performance Based on Search Engine Optimization Using Automated Software Testing Gtmetrix. *International Journal of Computer and Information System (IJCIS)*, 3(1), 17-20.
- [314] Mokhtari, H., Saberi, M. K., Amiri, M. R., Vakilmofrad, H., & Moradi, Z. (2022). Evaluating the Speed and Performance of the Websites of Hospitals and Specialty and Super-specialty Clinics of Hamadan University of Medical Sciences by GTmetrix. *Informology*, 1(1), 57-66.
- [315] Widyani, N. P. K., Sudana, A. K. O., & Piarsa, I. N. (2021). Pengujian Performa Sistem Informasi Perpustakaan Online pada Universitas Hindu Indonesia (Astakali UNHI) Menggunakan Tools GTmetrix. *jurnal Ilmiah Teknologi dan Komputer*, 2(3).
- [316] Sarita, K., Kaur, P., & Kaur, S. (2022, February). Accessibility and performance evaluation of healthcare and e-learning sites in India: a comparative study using TAW and GTMetrix. In *International conference on computing in engineering & technology* (pp. 172-187). Singapore: Springer Nature Singapore.
- [317] Manuaba, I. B. K. (2021). Performance Comparison of Text Based Game Prototypes Using GTmetrix. *Journal of Games, Game Art, and Gamification*, 6(1), 1-6.
- [318] Hidayati, N. H. (2022, December). Analisis Performa Website Kantor Pencarian Dan Pertolongan Palembang Menggunakan Gtmetrix. In *Seminar Nasional Teknologi Dan Multidisiplin Ilmu (SEMNASTEKMU) (Vol. 2, No. 1, pp. 229-234)*.
[reduce.prz.edu.pl/pliki/raport_gmetric.pdf](https://www.researchgate.net/publication/361111111-reduce.prz.edu.pl/pliki/raport_gmetric.pdf)
- [320] Hossain, M. T., Hassan, R., Amjad, M., & Rahman, M. A. (2021). Web Performance Analysis: An Empirical Analysis of E-Commerce Sites in Bangladesh. *International Journal of Information Engineering & Electronic Business*, 13(4).
- [321] Semma, A. (2023). Speed Matters: Measuring Time to First Byte and Other Key Metrics in VueJS Components Frameworks: Array. *Indonesian Journal of Informatics and Research*, 4(1), 1-7.
- [322] Janssen, K., Pelle, T., De Geus, L., Van Der Gronden, R., Islam, T., & Malavolta, I. (2022, June). On the impact of the critical css technique on the performance and energy consumption of mobile browsers. In *Proceedings of the 26th International Conference on Evaluation and Assessment in Software Engineering* (pp. 130-139).
- [323] Budiman, E., Puspitasari, N., Alam, S. N., Akbar, M. A., & Indra, D. (2018, October). Performance Analysis of the Resource Loading Time for Borneo Biodiversity Information System. In *2018 Third International Conference on Informatics and Computing (ICIC)* (pp. 1-5). IEEE.
- [324] Nurshuhada, A., Yusop, R. O. M., Azmi, A., Ismail, S. A., Sarkan, H. M., & Kama, N. (2019, December). Enhancing performance aspect in usability guidelines for mobile web application. In *2019 6th International Conference on Research and Innovation in Information Systems (ICRIIS)* (pp. 1-6). IEEE.
- [325] Thomas, I. (2023). The Effect of Website Interactivity on the Popularity of News Sites.
- [326] Netravali, R., Nathan, V., Mickens, J., & Balakrishnan, H. (2018). Vesper: Measuring {Time-to-Interactivity} for Web Pages. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)* (pp. 217-231).
- [327] Iskandar, T. F., Lubis, M., Kusumasari, T. F., & Lubis, A. R. (2020, May). Comparison between client-side and server-side rendering in the web development. In *IOP Conference Series: Materials Science and Engineering (Vol. 801, No. 1, p. 012136)*. IOP Publishing.
- [328] Hoßfeld, T., Metzger, F., & Rossi, D. (2018, May). Speed index: Relating the industrial standard for user perceived web performance to web qoe. In *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)* (pp. 1-6). IEEE.
- [329] Liu, W., Yang, X., Lin, H., Li, Z., & Qian, F. (2022). Fusing speed index during Web page loading. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 6(1), 1-23.
- [330] Wassermann, S., Casas, P., Seufert, M., Wehner, N., Schüler, J., & Hossfeld, T. (2020). How good is your mobile (web) surfing? Speed index inference from encrypted traffic. In *Proceedings of the SIGCOMM'20 Poster and Demo Sessions* (pp. 35-36).
- [331] Nilsson, S. (2021). Evaluation of the Blazor and Angular frameworks performance for web applications.

- [332] Vasiljević, V., Kojić, N., & Vugdelija, N. (2020, October). A new approach in quantifying user experience in web-oriented applications. In 4th International Scientific Conference on Recent Advances in Information Technology, Tourism, Economics, Management and Agriculture–ITEMA (pp. 9-16).
- [333] Sulistiyani, E. (2021, October). Evaluation of IBSI Education System Use ISO/IEC 9126 Quality Model: How is the Quality?. In 2021 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE) (pp. 68-73). IEEE.
- [334] Wibagso, S. S. (2023). The empirical study of Joomla CMS map extension and location performance. *Matrix: Jurnal Manajemen Teknologi dan Informatika*, 13(2), 94-105.
- [335] Edgar, M. (2023). Page Experience: Core Web Vitals and More. In *Tech SEO Guide: A Reference Guide for Developers and Marketers Involved in Technical SEO* (pp. 95-106). Berkeley, CA: Apress.
- [336] Munandar, A. (2022, November). Evaluating the Village Website for Improving Its Quality. In 2022 IEEE 7th International Conference on Information Technology and Digital Applications (ICITDA) (pp. 1-4). IEEE.
- [337] Wehner, N., Amir, M., Seufert, M., Schatz, R., & Hoßfeld, T. (2022, September). A Vital Improvement? Relating Google's Core Web Vitals to Actual Web QoE. In 2022 14th International Conference on Quality of Multimedia Experience (QoMEX) (pp. 1-6). IEEE.
- [338] Jun, B., Bustamante, F., Greenstein, B., & Clelland, I. (2022, March). Reining in mobile web performance with document and permission policies. In *Proceedings of the 23rd Annual International Workshop on Mobile Computing Systems and Applications* (pp. 95-101).
- [339] Ardaiz, O., Freitag, F., & Navarro, L. (2001). Improving the service time of web clients using server redirection. *ACM SIGMETRICS Performance Evaluation Review*, 29(2), 39-44.
- [340] Abuqaddom, I., Alazzam, H., Hudaib, A., & Al-Zaghoul, F. (2019, June). A measurable website usability model: Case Study University of Jordan. In 2019 10th International Conference on Information and Communication Systems (ICICS) (pp. 83-87). IEEE.
- [341] Iliev, I., & Dimitrov, G. P. (2014, May). Front end optimization methods and their effect. In 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (pp. 467-473). IEEE.
- [342] Lécué, F. (2009). Optimizing qos-aware semantic web service composition. In *The Semantic Web-ISWC 2009: 8th International Semantic Web Conference, ISWC 2009, Chantilly, VA, USA, October 25-29, 2009. Proceedings 8* (pp. 375-391). Springer Berlin Heidelberg.
- [343] Crovella, M. E., Frangioso, R., & Harchol-Balter, M. (1999). Connection scheduling in web servers. Boston University Computer Science Department.
- [345] Halepovic, E., Pang, J., & Spatscheck, O. (2012, November). Can you GET me now? Estimating the time-to-first-byte of HTTP transactions with passive measurements. In *Proceedings of the 2012 Internet Measurement Conference* (pp. 115-122).
- [346] Budiman, E., Puspitasari, N., Alam, S. N., Akbar, M. A., & Indra, D. (2018, October). Performance Analysis of the Resource Loading Time for Borneo Biodiversity Information System. In 2018 Third International Conference on Informatics and Computing (ICIC) (pp. 1-5). IEEE.
- [347] Sundaresan, S., Magharei, N., Feamster, N., Teixeira, R., & Crawford, S. (2013, June). Web performance bottlenecks in broadband access networks. In *Proceedings of the ACM SIGMETRICS/international conference on Measurement and modeling of computer systems* (pp. 383-384).
- [348] Wang, J., & Kai, Z. (2021, June). Performance analysis and optimization of nginx-based web server. In *Journal of Physics: Conference Series* (Vol. 1955, No. 1, p. 012033). IOP Publishing.
- [349] Liu, X., Sha, L., Diao, Y., Froehlich, S., Hellerstein, J. L., & Parekh, S. (2003). Online response time optimization of apache web server. In *Quality of Service—IWQoS 2003: 11th International Workshop Berkeley, CA, USA, June 2-4, 2003 Proceedings 11* (pp. 461-478). Springer Berlin Heidelberg.
- [350] Asrese, A. S., Eravuchira, S. J., Bajpai, V., Sarolahti, P., & Ott, J. (2019). Measuring web latency and rendering performance: Method, tools, and longitudinal dataset. *IEEE Transactions on Network and Service Management*, 16(2), 535-549.
- [351] Wang, X. S., Balasubramanian, A., Krishnamurthy, A., & Wetherall, D. (2013). Demystifying page load performance with {WProf}. In 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13) (pp. 473-485).
- [352] Hakim, M. (2016, April). Speed index and critical path rendering performance for isomorphic single page applications. In *The 16th Winona Computer Science Undergraduate Research Symposium* (p. 41).
- [353] Da Silva, W. O., & Farah, P. R. (2018, September). Characteristics and performance assessment of approaches pre-rendering and isomorphic javascript as a complement to spa architecture. In *Proceedings of the VII Brazilian Symposium on Software Components, Architectures, and Reuse* (pp. 63-72).

- [354] Rey, W. P., & Juanatas, R. (2022, March). Towards a Performance Optimization of Mobile Automated Fingerprint Identification System (MAFIS) for the Philippine National Police. In *Proceedings of the 8th International Conference on Computing and Artificial Intelligence* (pp. 380-386).
- [355] Vogel, L., & Springer, T. (2022, July). An in-depth analysis of web page structure and efficiency with focus on optimization potential for initial page load. In *International Conference on Web Engineering* (pp. 101-116). Cham: Springer International Publishing.
- [356] Losada, J., Raposo, J., Pan, A., Montoto, P., & Álvarez, M. (2015, October). Optimization techniques to speed up the page loading in custom web browsers. In *2015 IEEE 12th International Conference on e-Business Engineering* (pp. 7-14). IEEE.
- [357] Enghardt, T., Zinner, T., & Feldmann, A. (2019). Web performance pitfalls. In *Passive and Active Measurement: 20th International Conference, PAM 2019, Puerto Varas, Chile, March 27–29, 2019, Proceedings 20* (pp. 286-303). Springer International Publishing.
- [358] Ramakrishnan, R., & Kaur, A. (2020). An empirical comparison of predictive models for web page performance. *Information and Software Technology*, 123, 106307.
- [359] Vaughan-Nichols, S. J. (2010). Will HTML 5 restandardize the web?. *Computer*, 43(4), 13-15.
- [360] Beatty, P., Dick, S., & Miller, J. (2008). Is HTML in a race to the bottom? A large-scale survey and analysis of conformance to W3C standards. *IEEE Internet Computing*, 12(2), 76-80.
- [361] Hantke, F., & Stock, B. (2022, October). HTML violations and where to find them: a longitudinal analysis of specification violations in HTML. In *Proceedings of the 22nd ACM Internet Measurement Conference* (pp. 358-373).
- [362] HTML (2022). HTML (HyperText Markup Language). MDN Web Docs [online].
- [363] Sreedhar, G. (2019). Improving usability of website design using W3C guidelines. In *Advanced Methodologies and Technologies in Network Architecture, Mobile Computing, and Data Analytics* (pp. 1744-1753). IGI Global.
- [364] Alajarmeh, N. (2022). Evaluating the accessibility of public health websites: an exploratory cross-country study. *Universal access in the information society*, 21(3), 771-789.
- [365] Najadat, H., Al-Badameh, A., & Alodibat, S. (2021). A review of website evaluation using web diagnostic tools and data envelopment analysis. *Bulletin of Electrical Engineering and Informatics*, 10(1), 258-265.
- [366] Chen, S., Hong, D., & Shen, V. Y. (2005). An experimental study on validation problems with existing html webpages. In *Proceedings of the 2005 International Conference on Internet Computing, ICOMP'05* (p. 373).
- [367] Kosek, J., & Nálezka, P. (2006, May). Relaxed: on the way towards true validation of compound documents. In *Proceedings of the 15th international conference on World Wide Web* (pp. 427-436).
- [368] Henick, B. (2010). *HTML & CSS: The Good Parts: Better Ways to Build Websites That Work*. " O'Reilly Media, Inc."
- [369] Aronson, L. (2010). *HTML Manual of Style: A Clear, Concise Reference for Hypertext Markup Language (including HTML5)*. Pearson Education.
- [370] McManus, S. (2023). *Web design in easy steps*. In Easy Steps Limited.
- [371] Lie, H. W., & Saarela, J. (1999). Multipurpose Web publishing using HTML, XML, and CSS. *Communications of the ACM*, 42(10), 95-101.
- [372] Cyroň, M. (2006). *CSS-kaskádové styly: praktický manuál*. Grada Publishing as.
- [373] Baker, S. C. (2014). Making it work for everyone: HTML5 and CSS level 3 for responsive, accessible design on your library's web site. *Journal of Library & Information Services in Distance Learning*, 8(3-4), 118-136.
- [374] Durango, A. (2015). *Diseño Web con CSS: 2ª Edición*. IT Campus Academy.
- [375] Costa, L. M., Leal, J. P., & Queirós, R. (2023). Automated Assessment of Simple Web Applications (Short Paper). In *4th International Computer Programming Education Conference (ICPEC 2023)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
- [376] Alhadreti, O. (2023). Investigating the accessibility of banking websites in Saudi Arabia. *Universal Access in the Information Society*, 1-16.
- [377] Karumanchi, P. C., Dasari, Y., Kolli, V. K. K., & Mothadakala, R. A Novel Recommendation System for Code Validation and Optimal Refactoring (Nrscvor). Available at SSRN 4474677.
- [378] Sholikhan, M. (2022). HTML, CSS dan Javascript. Penerbit Yayasan Prima Agus Teknik, 1-343.
- [379] Akgül, Y., Uymaz, A. O., & Baba, A. (2023). Accessibility, usability, readability, and security analysis of company websites.
- [380] Kelly, B., Nevile, L., Sloan, D., Fanou, S., Ellison, R., & Herrod, L. (2009). From web accessibility to web adaptability. *Disability and Rehabilitation: Assistive Technology*, 4(4), 212-226.
- [381] Plessers, P., Casteleyn, S., Yesilada, Y., De Troyer, O., Stevens, R., Harper, S., & Goble, C. (2005, May). Accessibility: a web engineering approach. In *Proceedings of the 14th international conference on World Wide Web* (pp. 353-362).
- [382] Ferati, M., & Vogel, B. (2020, March). Accessibility in web development courses: A case study. In *Informatics (Vol. 7, No. 1, p. 8)*. MDPI.

- [383] Elisa, N. (2020). Usability, accessibility and web security assessment of e-government websites in Tanzania. arXiv preprint arXiv:2006.14245.
- [384] Abuaddous, H. Y., Jali, M. Z., & Basir, N. (2016). Web accessibility challenges. *International Journal of Advanced Computer Science and Applications (IJACSA)*.
- [385] Ismail, A., & Kuppasamy, K. S. (2022). Web accessibility investigation and identification of major issues of higher education websites with statistical measures: A case study of college websites. *Journal of King Saud University-Computer and Information Sciences*, 34(3), 901-911.
- [386] Brophy, P., & Craven, J. (2007). Web accessibility. *Library trends*, 55(4), 950-972.
- [387] Campoverde-Molina, M., Lujan-Mora, S., & Garcia, L. V. (2020). Empirical studies on web accessibility of educational websites: A systematic literature review. *IEEE Access*, 8, 91676-91700.
- [388] Freire, A. P., Russo, C. M., & Fortes, R. P. D. M. (2008). The perception of accessibility in web development by academy, industry and government: a survey of the brazilian scenario. *New Review of Hypermedia and Multimedia*, 14(2), 149-175.
- [389] Caldwell, B., Cooper, M., Reid, L. G., Vanderheiden, G., Chisholm, W., Slatin, J., & White, J. (2008). Web content accessibility guidelines (WCAG) 2.0. *WWW Consortium (W3C)*, 290, 1-34.
- [390] Rømen, D., & Svanæs, D. (2012). Validating WCAG versions 1.0 and 2.0 through usability testing with disabled users. *Universal Access in the Information Society*, 11, 375-385.
- [391] Kelly, B., Sloan, D., Phipps, L., Petrie, H., & Hamilton, F. (2005, May). Forcing standardization or accommodating diversity? A framework for applying the WCAG in the real world. In *Proceedings of the 2005 International Cross-Disciplinary Workshop on Web Accessibility (W4A)* (pp. 46-54).
- [392] Zadrożny, J. (2014). Web Content Accessibility Guidelines (WCAG) 2.0—zasady i wytyczne do tworzenia dostępnych serwisów internetowych. *Niepełnosprawność—zagadnienia, problemy, rozwiązania*, 3(12), 17-23.
- [393] World Wide Web Consortium. (2008). Web content accessibility guidelines (WCAG) 2.0.
- [394] Reid, L. G., & Snow-Weaver, A. (2008, April). WCAG 2.0: a web accessibility standard for the evolving web. In *Proceedings of the 2008 international cross-disciplinary conference on Web accessibility (W4A)* (pp. 109-115).
- [395] Maliszewska, S. (2023). Zarządzanie dostępnością cyfrową w instytucjach kultury.
- [396] Brajnik, G., Yesilada, Y., & Harper, S. (2012). Is accessibility conformance an elusive property? A study of validity and reliability of WCAG 2.0. *ACM Transactions on Accessible Computing (TACCESS)*, 4(2), 1-28.
- [397] Chee, M., & Weaver, K. D. (2021). Meeting a higher standard: A case study of accessibility compliance in LibGuides upon the adoption of WCAG 2.0 guidelines. *Journal of Web Librarianship*, 15(2), 69-89.
- [398] Connor, J. O. (2010). Is the accessibility audit dead?. In *Computers Helping People with Special Needs: 12th International Conference, ICCHP 2010, Vienna, Austria, July 14-16, 2010. Proceedings 12* (pp. 413-416). Springer Berlin Heidelberg.
- [399] Iniesto, F., McAndrew, P., Minocha, S., & Coughlan, T. (2019). Auditing the accessibility of MOOCs: a four-component approach. In *Transforming Learning with Meaningful Technologies: 14th European Conference on Technology Enhanced Learning, EC-TEL 2019, Delft, The Netherlands, September 16–19, 2019, Proceedings 14* (pp. 650-654). Springer International Publishing.
- [400] <https://validator.w3.org/>
- [401] <https://jigsaw.w3.org/css-validator/>
- [402] <https://accessibe.com/accessscan?website=reduce.prz.edu.pl>
- [403] Bawden, D., Holtham, C., & Courtney, N. (1999, October). Perspectives on information overload. In *Aslib proceedings* (Vol. 51, No. 8, pp. 249-255). MCB UP Ltd.
- [404] Bawden, D., & Robinson, L. (2020). Information overload: An overview.
- [405] Levy, D. M. (2008). Information overload. *The handbook of information and computer ethics*, 497-515.
- [406] Farhoomand, A. F., & Drury, D. H. (2002). Managerial information overload. *Association for computing machinery. Communications of the ACM*, 45(10), 127-127.
- [407] Misra, S., Roberts, P., & Rhodes, M. (2020). Information overload, stress, and emergency managerial thinking. *International Journal of Disaster Risk Reduction*, 51, 101762.
- [408] Buchanan, J., & Kock, N. (2001). Information overload: A decision making perspective. In *Multiple Criteria Decision Making in the New Millennium: Proceedings of the Fifteenth International Conference on Multiple Criteria Decision Making (MCDM)* Ankara, Turkey, July 10–14, 2000 (pp. 49-58). Springer Berlin Heidelberg.
- [409] Phillips-Wren, G., & Adya, M. (2020). Decision making under stress: The role of information overload, time pressure, complexity, and uncertainty. *Journal of Decision Systems*, 29(sup1), 213-225.
- [410] Kocher, M. G., & Sutter, M. (2006). Time is money—Time pressure, incentives, and the quality of decision-making. *Journal of Economic Behavior & Organization*, 61(3), 375-392.

- [411] Ahituv, N., Igbaria, M., & Sella, A. V. (1998). The effects of time pressure and completeness of information on decision making. *Journal of management information systems*, 15(2), 153-172.
- [412] Zhang, Z., & Jia, X. (2023). No Time for Ethics: How and When Time Pressure Leads to Abusive Supervisory Behavior. *Journal of Business Ethics*, 1-19.
- [413] Holsti, O. R. (2019). Crisis, stress, and decisionmaking. In *The War System* (pp. 491-508). Routledge.
- [414] Waldman, A. E. (2019). Power, process, and automated decision-making. *Fordham L. Rev.*, 88, 613.
- [415] Le Bris, S., Madrid-Guijarro, A., & Philippe Martin, D. (2019). Decision-making in complex environments under time pressure and risk of critical irreversibility: The role of meta rules. *M@ n@ gement*, 22(1), 1-29.
- [416] Abadie, M., & Waroquier, L. (2019). Evaluating the benefits of conscious and unconscious thought in complex decision making. *Policy Insights from the Behavioral and Brain Sciences*, 6(1), 72-78.
- [417] Paes, P., Leat, D., & Stewart, J. (2019). Complex decision making in medical training: key internal and external influences in developing practical wisdom. *Medical education*, 53(2), 165-174.
- [418] Kochenderfer, M. J., Wheeler, T. A., & Wray, K. H. (2022). *Algorithms for decision making*. MIT press.
- [419] Chao, X., Kou, G., Peng, Y., & Viedma, E. H. (2021). Large-scale group decision-making with non-cooperative behaviors and heterogeneous preferences: an application in financial inclusion. *European Journal of Operational Research*, 288(1), 271-293.
- [420] Xu, X., Zhang, Q., & Chen, X. (2020). Consensus-based non-cooperative behaviors management in large-group emergency decision-making considering experts' trust relations and preference risks. *Knowledge-Based Systems*, 190, 105108.
- [421] Gao, Y., & Zhang, Z. (2021). Consensus reaching with non-cooperative behavior management for personalized individual semantics-based social network group decision making. *Journal of the Operational Research Society*, 73(11), 2518-2535.
- [422] Evans, A. M., & Rand, D. G. (2019). Cooperation and decision time. *Current opinion in psychology*, 26, 67-71.
- [423] Corfman, K. P., & Lehmann, D. R. (1987). Models of cooperative group decision-making and relative influence: An experimental investigation of family purchase decisions. *Journal of Consumer Research*, 14(1), 1-13.
- [424] Declerck, C. H., Boone, C., & Emonds, G. (2013). When do people cooperate? The neuroeconomics of prosocial decision making. *Brain and cognition*, 81(1), 95-117.
- [425] Eckartz, S. M., Hofman, W. J., & Van Veenstra, A. F. (2014). A decision model for data sharing. In *Electronic Government: 13th IFIP WG 8.5 International Conference, EGOV 2014, Dublin, Ireland, September 1-3, 2014. Proceedings 13* (pp. 253-264). Springer Berlin Heidelberg.
- [426] Malenko, N. (2014). Communication and decision-making in corporate boards. *The Review of Financial Studies*, 27(5), 1486-1532.
- [427] Hilbert, M. (2012). Toward a synthesis of cognitive biases: how noisy information processing can bias human decision making. *Psychological bulletin*, 138(2), 211.
- [428] Kinsey, M. J., Gwynne, S. M. V., Kuligowski, E. D., & Kinatader, M. (2019). Cognitive biases within decision making during fire evacuations. *Fire technology*, 55, 465-485.
- [429] Acciarini, C., Brunetta, F., & Boccardelli, P. (2021). Cognitive biases and decision-making strategies in times of change: a systematic literature review. *Management Decision*, 59(3), 638-652.
- [430] Ehrlinger, J., Readinger, W. O., & Kim, B. (2016). Decision-making and cognitive biases. *Encyclopedia of mental health*, 12(3), 83-7.
- [431] Zhang, Y., Bellamy, R. K., & Kellogg, W. A. (2015, April). Designing information for remediating cognitive biases in decision-making. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems* (pp. 2211-2220).
- [432] Arnott, D. (2006). Cognitive biases and decision support systems development: a design science approach. *Information Systems Journal*, 16(1), 55-78.
- [433] Hwang, C. L., & Masud, A. S. M. (2012). *Multiple objective decision making—methods and applications: a state-of-the-art survey* (Vol. 164). Springer Science & Business Media.
- [434] Tolan, S. (2019). Fair and unbiased algorithmic decision making: Current state and future challenges. *arXiv preprint arXiv:1901.04730*.
- [435] Harrison, J. R., & March, J. G. (1984). Decision making and postdecision surprises. *Administrative Science Quarterly*, 26-42.
- [436] Stasser, G., & Titus, W. (1985). Pooling of unshared information in group decision making: Biased information sampling during discussion. *Journal of personality and social psychology*, 48(6), 1467.
- [437] Dricu, M., Ceravolo, L., Grandjean, D., & Frühholz, S. (2017). Biased and unbiased perceptual decision-making on vocal emotions. *Scientific reports*, 7(1), 16274.
- [438] Einfeld, A. K. (2023). Unbiased decision-making for acute myeloid leukemia still needed. *Haematologica*, 108(3), 668.
- [439] Marshall, J. A., Trimmer, P. C., & Houston, A. I. (2013). Unbiased individuals use valuable information when making decisions: a reply to Johnson and Fowler. *Trends in ecology & evolution*, 28(8), 444-445.

- [440] Cecere, G., Corrocher, N., & Jean, C. (2021). Fair or Unbiased Algorithmic Decision-Making? A Review of the Literature on Digital Economics. A Review of the Literature on Digital Economics (October 15, 2021). Bocconi University Management Research Paper.
- [441] Lerner, J. S., Li, Y., Valdesolo, P., & Kassam, K. S. (2015). Emotion and decision making. *Annual review of psychology*, 66, 799-823.
- [442] Tenenbaum, G., Basevitch, I., Gershgoren, L., & Filho, E. (2013). Emotions–decision-making in sport: Theoretical conceptualization and experimental evidence. *International Journal of Sport and Exercise Psychology*, 11(2), 151-168.
- [443] Zaleskiewicz, T., & Traczyk, J. (2020). Emotions and financial decision making. *Psychological perspectives on financial decision making*, 107-133.
- [444] Schwarz, N. (2000). Emotion, cognition, and decision making. *Cognition & emotion*, 14(4), 433-440.
- [445] Hopkins, A. (2011). Risk-management and rule-compliance: Decision-making in hazardous industries. *Safety science*, 49(2), 110-120.
- [446] Slovic, P., Peters, E., Finucane, M. L., & MacGregor, D. G. (2005). Affect, risk, and decision making. *Health psychology*, 24(4S), S35.
- [447] Xia, D., & Chen, B. (2011). A comprehensive decision-making model for risk management of supply chain. *Expert Systems with Applications*, 38(5), 4957-4966.
- [448] Levy, J. K. (2005). Multiple criteria decision making and decision support systems for flood risk management. *Stochastic Environmental Research and Risk Assessment*, 19, 438-447.
- [449] Beroggi, G. E., & Wallace, W. A. (1994). Operational risk management: a new paradigm for decision making. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(10), 1450-1457.
- [450] Wu, D., Olson, D. L., & Dolgui, A. (2015). Decision making in enterprise risk management: A review and introduction to special issue. *Omega*, 57, 1-4.
- [451] Dey, P. K. (2012). Project risk management using multiple criteria decision-making technique and decision tree analysis: a case study of Indian oil refinery. *Production Planning & Control*, 23(12), 903-921.
- [452] Dey, P. K. (2001). Decision support system for risk management: a case study. *Management Decision*, 39(8), 634-649.
- [453] Rasmussen, J. (1997). Risk management in a dynamic society: a modelling problem. *Safety science*, 27(2-3), 183-213.
- [454] Yoe, C. (2019). *Principles of risk analysis: decision making under uncertainty*. CRC press.
- [455] Choice, E. (2000). Expert choice. *Analytical Hierarchy Process (AHP) Software*, Version, 9.
- [456] Ishizaka, A., & Labib, A. (2009). Analytic hierarchy process and expert choice: Benefits and limitations. *OR insight*, 22(4), 201-220.
- [457] Erdogan, S. A., Šaparauskas, J., & Turskis, Z. (2017). Decision making in construction management: AHP and expert choice approach. *Procedia engineering*, 172, 270-276.
- [458] Alanbay, O. (2005). ERP selection using expert choice software. Honolulu, Hawaii, July, 8(10).
- [459] Yulianto, J. (2017). Pemilihan alat pancang menggunakan expert choice. *Jurnal Riset Rekayasa Sipil*, 1(1), 50-58.
- [460] <https://qvistorp.com/>
- [461] <https://paramountdecisions.com/>
- [462] Barić, T. (2019). Kolaborativno odlučivanje koristeći alat Paramount Decisions (Doctoral dissertation, The Polytechnic of Rijeka).
- [463] Kumar, V. (2014). Making " freemium" work. *Harvard business review*, 92(5), 27-29.
- [464] Gu, X., Kannan, P. K., & Ma, L. (2018). Selling the premium in freemium. *Journal of Marketing*, 82(6), 10-27.
- [465] Holm, A. B., & Günzel-Jensen, F. (2017). Succeeding with freemium: strategies for implementation. *Journal of Business Strategy*, 38(2), 16-24.
- [466] Pujol, N. (2010). Freemium: attributes of an emerging business model. Available at SSRN 1718663.
- [467] Kimppa, K. K., Heimo, O. I., & Harviainen, J. T. (2016). First dose is always freemium. *Acm Sigcas Computers and Society*, 45(3), 132-137.
- [468] Geddes, B. (2014). *Advanced Google AdWords*. John Wiley & Sons.
- [469] Mallipeddi, R. R., Kumar, S., Sriskandarajah, C., & Zhu, Y. (2022). A framework for analyzing influencer marketing in social networks: selection and scheduling of influencers. *Management Science*, 68(1), 75-104.
- [470] Soegoto, E. S., & Semesta, R. B. (2018, August). Use of google AdSense for income generating activity. In *IOP Conference Series: Materials Science and Engineering* (Vol. 407, No. 1, p. 012065). IOP Publishing.
- [471] Prasanna, R. P. I. R., Jayasundara, J. M. S. B., Naradda Gamage, S. K., Ekanayake, E. M. S., Rajapakshe, P. S. K., & Abeyrathne, G. A. K. N. J. (2019). Sustainability of SMEs in the competition: A systemic review on technological challenges and SME performance. *Journal of Open Innovation: Technology, Market, and Complexity*, 5(4), 100.

- [472] Salles, M. (2006). Decision making in SMEs and information requirements for competitive intelligence. *Production Planning & Control*, 17(3), 229-237.
- [473] Gilmore, A., & Carson, D. (2000). The demonstration of a methodology for assessing SME decision making. *Journal of Research in Marketing and Entrepreneurship*, 2(2), 108-124.
- [474] Huang, X. (2009). Strategic decision making in Chinese SMEs. *Chinese Management Studies*, 3(2), 87-101.
- [475] O'Regan, N., Sims, M., & Ghobadian, A. (2005). High performance: ownership and decision-making in SMEs. *Management Decision*, 43(3), 382-396.
- [476] Francioni, B., Musso, F., & Cioppi, M. (2015). Decision-maker characteristics and international decisions for SMEs. *Management Decision*, 53(10), 2226-2249.
- [477] Fulop, L., Linstead, S., Fulop, L., Linstead, S., & Clarke, R. J. (1999). Decision making in organizations. *Management: A critical text*, 295-334.
- [478] Sutcliffe, K. M., & McNamara, G. (2001). Controlling decision-making practice in organizations. *Organization science*, 12(4), 484-501.
- [479] Busenitz, L. W., & Barney, J. B. (1997). Differences between entrepreneurs and managers in large organizations: Biases and heuristics in strategic decision-making. *Journal of business venturing*, 12(1), 9-30.
- [480] Busenitz, L. W., & Barney, J. B. (1994, August). Biases and heuristics in strategic decision making: Differences between entrepreneurs and managers in large organizations. In *Academy of management proceedings* (Vol. 1994, No. 1, pp. 85-89). Briarcliff Manor, NY 10510: Academy of Management.
- [481] Uzonwanne, F. (2015). Leadership styles and decision-making models among corporate leaders in non-profit organizations in North America. *Journal of Public Affairs*, 15(3), 287-299.
- [482] Hess, J. D., & Bacigalupo, A. C. (2013). Applying emotional intelligence skills to leadership and decision making in non-profit organizations. *Administrative Sciences*, 3(4), 202-220.
- [483] Bryson, J. M. (1988). A strategic planning process for public and non-profit organizations. *Long range planning*, 21(1), 73-81.
- [484] Zhu, H., Wang, P., & Bart, C. (2016). Board processes, board strategic involvement, and organizational performance in for-profit and non-profit organizations. *Journal of Business Ethics*, 136, 311-328.
- [485] Heikkila, T., & Isett, K. R. (2004). Modeling operational decision making in public organizations: An integration of two institutional theories. *The American Review of Public Administration*, 34(1), 3-19.
- [486] Coursey, D., & Bozeman, B. (1990). Decision making in public and private organizations: A test of alternative concepts of publicness. *Public Administration Review*, 50(5), 525-535.
- [487] Abdel-Maksoud, A., Elbanna, S., Mahama, H., & Pollanen, R. (2015). The use of performance information in strategic decision making in public organizations. *International Journal of Public Sector Management*, 28(7), 528-549.
- [488] Pollanen, R., Abdel-Maksoud, A., Elbanna, S., & Mahama, H. (2017). Relationships between strategic performance measures, strategic decision-making, and organizational performance: empirical evidence from Canadian public organizations. *Public Management Review*, 19(5), 725-746.
- [489] Wihlborg, E., Larsson, H., & Hedström, K. (2016, January). "The Computer Says No!"--A Case Study on Automated Decision-Making in Public Authorities. In *2016 49th Hawaii International Conference on System Sciences (HICSS)* (pp. 2903-2912). IEEE.
- [490] Carol Rusaw, A. (2007). Changing public organizations: Four approaches. *International Journal of Public Administration*, 30(3), 347-361.
- [491] Ball, C. (2009). What is transparency?. *Public Integrity*, 11(4), 293-308.
- [492] Delimatsis, P. (2014). Transparency in the WTO's Decision-Making. *Leiden Journal of International Law*, 27(3), 701-726.
- [493] Grimmelikhuijsen, S. G. (2010). Transparency of public decision-making: Towards trust in local government?. *Policy & Internet*, 2(1), 5-35.
- [494] Doh, J. P., & Guay, T. R. (2006). Corporate social responsibility, public policy, and NGO activism in Europe and the United States: An institutional-stakeholder perspective. *Journal of Management studies*, 43(1), 47-73.
- [495] Brunsson, N. (1990). Deciding for responsibility and legitimation: Alternative interpretations of organizational decision-making. *Accounting, Organizations and Society*, 15(1-2), 47-59.
- [496] Salancik, G. R., & Pfeffer, J. (1974). The bases and use of power in organizational decision making: The case of a university. *Administrative science quarterly*, 453-473.
- [497] Faraj Allah, A., El Talla, S. A., Abu-Naser, S. S., & Al Shobaki, M. J. (2018). Participation of Administrative Staff in Decision-Making and Their Relation to the Nature of Work in Universities.

- [498] Ysseldyke, J. E., Thurlow, M., Graden, J., Wesson, C., Algozzine, B., & Deno, S. (1983). Generalizations from five years of research on assessment and decision making: The University of Minnesota Institute. *Exceptional Education Quarterly*, 4(1), 75-93.
- [499] Mengash, H. A. (2020). Using data mining techniques to predict student performance to support decision making in university admission systems. *Ieee Access*, 8, 55462-55470.
- [500] Wilmore, A. (2014). IT strategy and decision-making: a comparison of four universities. *Journal of Higher Education Policy and Management*, 36(3), 279-292.
- [501] Aikenhead, G. S. (1985). Collective decision making in the social context of science. *Science education*, 69(4), 453-75.
- [502] Bell, R. L., & Lederman, N. G. (2003). Understandings of the nature of science and decision making on science and technology based issues. *Science education*, 87(3), 352-377.
- [503] Carolan, M. S. (2006). Science, expertise, and the democratization of the decision-making process. *Society and Natural resources*, 19(7), 661-668.
- [504] Corcoran, J., & McLean, F. (1998). The selection of management consultants: How are governments dealing with this difficult decision? An exploratory study. *International Journal of Public Sector Management*, 11(1), 37-54.
- [505] Driever, E. M., Stiggelbout, A. M., & Brand, P. L. (2022). Do consultants do what they say they do? Observational study of the extent to which clinicians involve their patients in the decision-making process. *BMJ open*, 12(1), e056471.
- [506] Kieser, A., & Wellstein, B. (2008). Do activities of consultants and management scientists affect decision making by managers?.
- [507] Bouwmeester, O., & van Werven, R. (2011). Consultants as legitimizers: exploring their rhetoric. *Journal of Organizational Change Management*, 24(4), 427-441.
- [508] E Cunha, M. P., Palma, P., & da Costa, N. G. (2006). Fear of foresight: Knowledge and ignorance in organizational foresight. *Futures*, 38(8), 942-955.
- [509] Reger, G. (2001). Technology foresight in companies: from an indicator to a network and process perspective. *Technology analysis & strategic management*, 13(4), 533-553.
- [510] Vishnevskiy, K., & Karasev, O. (2016). Challenges and opportunities for corporate foresight. Deploying foresight for policy and strategy makers: Creating opportunities through public policies and corporate strategies in science, technology and innovation, 65-79.
- [511] Gokhberg, L., Meissner, D., & Sokolov, A. (2016). Foresight: turning challenges into opportunities. Deploying foresight for policy and strategy makers: Creating opportunities through public policies and corporate strategies in science, technology and innovation, 1-8.
- [512] Toivonen, M. (2004). Foresight in services: possibilities and special challenges. *The Service Industries Journal*, 24(1), 79-98.
- [513] Xu, H., Gupta, S., Rosson, M. B., & Carroll, J. M. (2012). Measuring mobile users' concerns for information privacy.
- [514] Zukowski, T., & Brown, I. (2007, October). Examining the influence of demographic factors on internet users' information privacy concerns. In *Proceedings of the 2007 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries* (pp. 197-204).
- [515] Malhotra, N. K., Kim, S. S., & Agarwal, J. (2004). Internet users' information privacy concerns (IUIPC): The construct, the scale, and a causal model. *Information systems research*, 15(4), 336-355.
- [516] Bélanger, F., & Crossler, R. E. (2011). Privacy in the digital age: a review of information privacy research in information systems. *MIS quarterly*, 1017-1041.
- [517] Son, J. Y., & Kim, S. S. (2008). Internet users' information privacy-protective responses: A taxonomy and a nomological model. *MIS quarterly*, 503-529.
- [518] Schwartz, P. M. (2012). Information privacy in the cloud. *U. Pa. L. Rev.*, 161, 1623.
- [519] Sun, Y., Zhang, J., Xiong, Y., & Zhu, G. (2014). Data security and privacy in cloud computing. *International Journal of Distributed Sensor Networks*, 10(7), 190903.
- [520] Bertino, E. (2016, June). Data security and privacy: Concepts, approaches, and research directions. In *2016 IEEE 40th Annual computer Software and Applications conference (COMPSAc)* (Vol. 1, pp. 400-407). IEEE.
- [521] Geambasu, R., Kohno, T., Levy, A. A., & Levy, H. M. (2009, August). Vanish: Increasing Data Privacy with Self-Destructing Data. In *USENIX security symposium* (Vol. 316, pp. 10-5555).
- [522] Tene, O., & Polonetsky, J. (2011). Privacy in the age of big data: a time for big decisions. *Stan. L. Rev. Online*, 64, 63.
- [523] Samarati, P., & Di Vimercati, S. D. C. (2010, April). Data protection in outsourcing scenarios: Issues and directions. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security* (pp. 1-14).
- [524] Iezzi, M. (2020, December). Practical privacy-preserving data science with homomorphic encryption: an overview. In *2020 IEEE International Conference on Big Data (Big Data)* (pp. 3979-3988). IEEE.

- [525] Geambasu, R., Kohno, T., Levy, A. A., & Levy, H. M. (2009, August). Vanish: Increasing Data Privacy with Self-Destructing Data. In USENIX security symposium (Vol. 316, pp. 10-5555).
- [526] Leung, N. K., & Lau, S. K. (2007). Information technology help desk survey: To identify the classification of simple and routine enquiries. *Journal of Computer Information Systems*, 47(4), 70-81.
- [527] Serbest, S., Goksen, Y., Dogan, O., & Tokdemir, A. (2015). Design and implementation of help desk system on the effective focus of information system. *Procedia Economics and Finance*, 33, 461-467.
- [528] Goffin, K., & New, C. (2001). Customer support and new product development-An exploratory study. *International Journal of Operations & Production Management*, 21(3), 275-301.
- [529] Lowry, P. B., Zhang, D., Zhou, L., & Fu, X. (2010). Effects of culture, social presence, and group composition on trust in technology-supported decision-making groups. *Information Systems Journal*, 20(3), 297-315.
- [530] Leung, H., & Fan, Z. (2002). Software cost estimation. In *Handbook of Software Engineering and Knowledge Engineering: Volume II: Emerging Technologies* (pp. 307-324).
- [531] Gartner, J., Fink, M., & Maresch, D. (2022). The role of fear of missing out and experience in the formation of SME decision makers' intentions to adopt new manufacturing technologies. *Technological Forecasting and Social Change*, 180, 121723.
- [532] Kligyte, V., Connelly, S., Thiel, C., & Devenport, L. (2013). The influence of anger, fear, and emotion regulation on ethical decision making. *Human Performance*, 26(4), 297-326.
- [533] Horwath, J., Kalyva, E., & Spyru, S. (2012). "I want my experiences to make a difference" promoting participation in policy-making and service development by young people who have experienced violence. *Children and Youth Services Review*, 34(1), 155-162.
- [534] Al-Harbi, K. M. A. S. (2001). Application of the AHP in project management. *International journal of project management*, 19(1), 19-27.
- [535] Han Rebekah Wong, S. (2012). Which platform do our users prefer: website or mobile app?. *Reference services review*, 40(1), 103-115.

Bibliografia rozdziału piątego

- [1] SAATY, T. L. (1977). A scaling method for priorities in hierarchical structures. *Journal of mathematical psychology* 15.3: 234-281
- [2] KURAŚ, P. (2023). Pairwise comparison matrices, their applications, properties and the problem of inconsistency and its reduction. *Advances in IT and Electrical Engineering*,
- [3] MAZUREK, J. (2023). *Advances in Pairwise Comparisons: The Detection, Evaluation and Reduction of Inconsistency*, Springer, Heidelberg
- [4] MAZUREK, J., PERZINA, R., STRZAŁKA, D., KOWAL, B., & KURAŚ, P. (2021). A numerical comparison of iterative algorithms for inconsistency reduction in pairwise comparisons. *IEEE Access*, 9, 62553-62561
- [5] CAO, D., LEUNG, L. C., LAW, J. S. (2008). Modifying Inconsistent Comparison Matrix in Analytic Hierarchy Process. *A Heuristic Approach Decision Support Systems*, 44, 944-953.
- [6] SZYBOWSKI, J. (2018). The improvement of data in pairwise comparison matrices. *Procedia Computer Science*, 126, 1006-1013.
- [7] ZESHUI, X., CUIPING, W. (1999). A consistency improving method in the analytic hierarchy process. *European journal of operational research*, 116(2), 443-449.
- [8] MAZUREK, J., PERZINA, R., STRZAŁKA, D., KOWAL, B. (2020). A new step-by-step (SBS) algorithm for in-consistency reduction in pairwise comparisons. *IEEE Access*, 8, 135821-135828.
- [9] ZAGATTI, Fernando Rezende, et al. MetaPrep: Data preparation pipelines recommendation via meta-learning. In: 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, 2021. p. 1197-1202.
- [10] STEVENS, Laura M., et al. Recommendations for reporting machine learning analyses in clinical research. *Circulation: Cardiovascular Quality and Outcomes*, 2020, 13.10: e006556.
- [11] Pajankar, A., & Joshi, A. (2022). Preparing data for machine learning. In *Hands-on Machine Learning with Python: Implement Neural Network Solutions with Scikit-learn and PyTorch* (pp. 79-97). Berkeley, CA: Apress.
- [12] ATLA, Abhinav, et al. Sensitivity of different machine learning algorithms to noise. *Journal of Computing Sciences in Colleges*, 2011, 26.5: 96-103.
- [13] SASEENDRAN, Arun Thundyill, et al. Impact of noise in dataset on machine learning algorithms. *Machine Learning Research*, 2019, 0-8.

- [14] KHALID, Faiq, et al. Fademl: Understanding the impact of pre-processing noise filtering on adversarial machine learning. In: 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2019. p. 902-907.
- [15] Kalapanidas, E., Avouris, N., Craciun, M., & Neagu, D. (2003, November). Machine learning algorithms: a study on noise sensitivity. In Proc. 1st Balcan Conference in Informatics (pp. 356-365). sn.
- [16] Talmon, J. L., & McNair, P. (1992). The effect of noise and biases on the performance of machine learning algorithms. *International journal of bio-medical computing*, 31(1), 45-57.
- [17] Molnar, C. (2020). *Interpretable machine learning*. Lulu. com.
- [18] Pääkkönen, J., & Ylikoski, P. (2021). Humanistic interpretation and machine learning. *Synthese*, 199(1-2), 1461-1497.
- [19] PEREIRA, V., & COSTA, H. G. (2018). AHP inconsistency reduction through two greedy algorithms application. *CEP*, 24, 240.
- [20] <https://www.sciencedirect.com/topics/computer-science/machine-learning>
- [21] <https://www.theguardian.com/science/2022/dec/18/the-10-biggest-science-stories-of-2022-chosen-by-scientists>
- [22] <https://www.nature.com/collections/jhjibhhicg>
- [23] <https://www.ilovephd.com/top-100-journals-in-the-world-with-impact-factor/>
- [24] REYES, Kristofer G.; MARUYAMA, Benji. The machine learning revolution in materials?. *MRS Bulletin*, 2019, 44.7: 530-537.
- [25] DEAN, Jeff; PATTERSON, David; YOUNG, Cliff. A new golden age in computer architecture: Empowering the machine-learning revolution. *IEEE Micro*, 2018, 38.2: 21-29.
- [26] Lemonick, S. (2018). Is machine learning overhyped. *Chem. Eng. News*, 96(34).
- [27] MANLHIOT, Cedric. Machine learning for predictive analytics in medicine: real opportunity or overblown hype?. *European Heart Journal-Cardiovascular Imaging*, 2018, 19.7: 727-728.
- [28] Cabitza, F., Rasoini, R., & Gensini, G. F. (2017). Unintended consequences of machine learning in medicine. *Jama*, 318(6), 517-518.
- [29] Rashidi, H. H., & Chen, M. (2023, March). Preface: Artificial Intelligence (AI), machine learning (ML) and digital pathology integration are the next major chapter in our diagnostic pathology and laboratory medicine arena. In *Seminars in Diagnostic Pathology*. WB Saunders.
- [30] Goodell, J. W., Kumar, S., Lim, W. M., & Pattnaik, D. (2021). Artificial intelligence and machine learning in finance: Identifying foundations, themes, and research clusters from bibliometric analysis. *Journal of Behavioral and Experimental Finance*, 32, 100577.
- [31] The Machine Learning Future Is Now: How AI Is Disrupting Entire Industries, Nasdaq, [Online]. Available: <https://www.nasdaq.com/articles/the-machine-learning-future-is-now%3A-how-ai-is-disrupting-entire-industries>. [Accessed: 20-August-2023].
- [32] Why Machine Learning is the Future, GeeksforGeeks, [Online]. Available: <https://www.geeksforgeeks.org/why-machine-learning-is-the-future/>. [Accessed: 20-August-2023].
- [33] "Uczenie maszynowe i metawersum technologiami przyszłości według NCBR," WNP, [Online]. Available: <https://www.wnp.pl/tech/uczenie-maszynowe-i-metawersum-technologiami-przyszlosci-wedlug-ncbr,522177.html>. [Accessed: 20-August-2023].
- [34] "Sztuczna inteligencja na rynku pracy czy zastąpi tradycyjnych pracowników? Opinie ekspertów," SuperBiz, [Online]. Available: <https://superbiz.se.pl/wiadomosci/sztuczna-inteligencja-na-ryнку-pracy-czy-zastapi-tradycyjnych-pracownikow-opinie-ekspertow-aa-A9z3-jbSk-9UBX.html>. [Accessed: 20-August-2023].
- [35] "PKO BP: Sztuczna inteligencja obsługa klienta," Telepolis, [Online]. Available: <https://www.telepolis.pl/fintech/taryfy-promocje-uslugi/pkobp-sztuczna-inteligencja-obsługa-klienta>. [Accessed: 20-August-2023].
- [36] "Sztuczna inteligencja: Jak może wyglądać nasza przyszłość?" Speedtest, [Online]. Available: <https://www.speedtest.pl/wiadomosci/esej/sztuczna-inteligencja-jak-moze-wygladac-nasza-przyszlosc/>. [Accessed: 20-August-2023].
- [37] "Przyszłość jest teraz: Poznaj 7 najciekawszych narzędzi AI do," GeekWeek, [Online]. Available: <https://geekweek.interia.pl/technologie/news-przyszlosc-jest-teraz-poznaj-7-najciekawszych-narzedzi-ai-do,nId,6412572>. [Accessed: 20-August-2023].
- [38] M. Maheshik, "Unraveling the Wonders of Machine Learning: A Journey into the Future," Medium, [Online]. Available: <https://medium.com/@maheshik998/unraveling-the-wonders-of-machine-learning-a-journey-into-the-future-6d1f99f1a852>. [Accessed: 20-August-2023].

- [39] "Ile jest warta sztuczna inteligencja? Najnowszy raport wycenia AI na fortune," Bithub, [Online]. Available: <https://bithub.pl/gospodarka/ile-jest-warta-sztuczna-inteligencja-najnowszy-raport-wycenia-ai-na-fortune/>. [Accessed: 20-August-2023].
- [40] "Sztuczna inteligencja: Nvidia, Microsoft, Google," Strefa Inwestorów, [Online]. Available: <https://strefainwestorow.pl/strefa-global-market/spolki/sztuczna-inteligencja-nvidia-microsoft-google>. [Accessed: 20-August-2023].
- [41] ROSENBERG, Michael A. Trusting magic: interpretability of predictions from machine learning algorithms. *Circulation*, 2021, 143.13: 1299-1301.
- [42] MODINE, Thomas; OVERTCHOUK, Pavel. Machine learning is No magic: a plea for critical appraisal during periods of hype. *Cardiovascular Interventions*, 2019, 12.14: 1339-1341.
- [43] SANDEEP, S. R., et al. To understand the relationship between Machine learning and Artificial intelligence in large and diversified business organisations. *Materials Today: Proceedings*, 2022, 56: 2082-2086.
- [44] MOTHILAL, Ramaravind K.; SHARMA, Amit; TAN, Chenhao. Explaining machine learning classifiers through diverse counterfactual explanations. In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 2020. p. 607-617.
- [45] HITRON, Tom, et al. Can children understand machine learning concepts? The effect of uncovering black boxes. In: *Proceedings of the 2019 CHI conference on human factors in computing systems*. 2019. p. 1-11.
- [46] FÜRNRANZ, Johannes. Machine learning in computer chess: The next generation. *ICGA Journal*, 1996, 19.3: 147-161.
- [47] Stilgoe, J. (2018). Machine learning, social learning and the governance of self-driving cars. *Social studies of science*, 48(1), 25-56.
- [48] <https://fortune.com/2023/07/25/video-game-studios-scared-ai-forcing-managers-study-machine-learning-offering-employees-7000-bounties-gala-sports/>
- [49] Clark, S. (2023, June 27). How Is AI Changing Digital Transformation? CMSWire. <https://www.cmswire.com/digital-experience/how-is-ai-changing-digital-transformation/>
- [50] Cheatham, B., Javanmardian, K., & Samandari, H. (2019). *Confronting the Risks of Artificial Intelligence*. McKinsey & Company. Retrieved from <https://www.mckinsey.com/capabilities/quantumblack/our-insights/confronting-the-risks-of-artificial-intelligence>
- [51] DEDEHAYIR, Ozgur; STEINERT, Martin. The hype cycle model: A review and future directions. *Technological Forecasting and Social Change*, 2016, 108: 28-41.
- [52] BOSCH-SIJTSEMA, Petra, et al. The hype factor of digital technologies in AEC. *Construction Innovation*, 2021, 21.4: 899-916.
- [53] EL NAQA, Issam; MURPHY, Martin J. *What is machine learning?*. Springer International Publishing, 2015.
- [54] MCCULLOCH, Warren S.; PITTS, Walter. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 1943, 5: 115-133.
- [55] Hebb, D. O. (1949). *The organization of behavior; a neuropsychological theory*. Wiley.
- [56] Pinar Saygin, A., Cicekli, I., & Akman, V. (2000). Turing test: 50 years later. *Minds and machines*, 10(4), 463-518.
- [57] Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3), 210-229.
- [58] ROSENBLATT, Frank. The perceptron, a perceiving and recognizing automaton Project Para. Cornell Aeronautical Laboratory, 1957.
- [59] Bezdek, J. C., Chuah, S. K., & Leep, D. (1986). Generalized k-nearest neighbor rules. *Fuzzy Sets and Systems*, 18(3), 237-256.
- [60] Moravec, H. P. (1983). The Stanford cart and the CMU rover. *Proceedings of the IEEE*, 71(7), 872-884.
- [61] DEJONG, G. F. *Explanation based learning*. 1986.
- [62] Fradkov, A. L. (2020). Early history of machine learning. *IFAC-PapersOnLine*, 53(2), 1385-1390.
- [63] Newborn, M. (2012). *Kasparov versus Deep Blue: Computer chess comes of age*. Springer Science & Business Media.
- [64] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436-444.
- [65] Zhang, Z. (2012). Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2), 4-10.
- [66] Ferrucci, D., Levas, A., Bagchi, S., Gondek, D., & Mueller, E. T. (2013). Watson: beyond jeopardy!. *Artificial Intelligence*, 199, 93-105.
- [67] Helms, M., Ault, S. V., Mao, G., & Wang, J. (2018, March). An overview of google brain and its applications. In *Proceedings of the 2018 International Conference on Big Data and Education* (pp. 72-75).
- [68] ЕЛИСТРАТОВА, А. А.; КУКАРЦЕВ, В. В. Технология идентификации личности Deepface. *Решетневские чтения*, 2014, 2.18: 186-187.

- [69] Rastogi, R. (2018, June). Machine learning@ amazon. In The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (pp. 1337-1338).
- [70] Seide, F., & Agarwal, A. (2016, August). CNTK: Microsoft's open-source deep-learning toolkit. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 2135-2135).
- [71] WANG, Fei-Yue, et al. Where does AlphaGo go: From church-turing thesis to AlphaGo thesis and beyond. *IEEE/CAA Journal of Automatica Sinica*, 2016, 3.2: 113-120.
- [72] CERF, Vinton G. A comprehensive self-driving car test. *Communications of the ACM*, 2018, 61.2: 7-7.
- [73] FLORIDI, Luciano; CHIRIATTI, Massimo. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 2020, 30: 681-694.
- [74] AGGARWAL, Karan, et al. Has the future started? The current growth of artificial intelligence, machine learning, and deep learning. *Iraqi Journal for Computer Science and Mathematics*, 2022, 3.1: 115-123.
- [75] EBERBACH, E., & STRZALKA, D. (2022). In Search of Machine Learning Theory. In Proceedings of the Future Technologies Conference (FTC) 2021, Volume 1 (pp. 599-615). Springer International Publishing.
- [76] Bishop, C. M. (2006). "Pattern Recognition and Machine Learning." Springer.
- [77] Hastie, T.; Tibshirani, R.; Friedman, J. (2009). "The Elements of Statistical Learning." Springer.
- [78] Cortes, C.; Vapnik, V. (1995). "Support-Vector Networks." *Machine Learning*.
- [79] Goodfellow, I.; Bengio, Y.; Courville, A. (2016). "Deep Learning." MIT Press.
- [80] Kohavi, R. (1995). "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection." *IJCAI*.
- [81] Sokolova, M.; Lapalme, G. (2009). "A systematic analysis of performance measures for classification tasks." *Information Processing & Management*.
- [82] Jordan, M. I.; Mitchell, T. M. (2015). "Machine learning: Trends, perspectives, and prospects." *Science*.
- [83] Hinton, G. E. (2012). A practical guide to training restricted Boltzmann machines. In *Neural Networks: Tricks of the Trade: Second Edition* (pp. 599-619). Berlin, Heidelberg: Springer Berlin Heidelberg.
- [84] Comesaña-Campos, A., & Bouza-Rodríguez, J. B. (2016). An application of Hebbian learning in the design process decision-making. *Journal of intelligent manufacturing*, 27, 487-506.
- [85] Carpenter, G.A. & Grossberg, S. (1988). "The ART of adaptive pattern recognition by a self-organizing neural network" (PDF). *Computer*. 21 (3): 77-88. doi:10.1109/2.33. S2CID 14625094.
- [86] Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78(9), 1464-1480.
- [87] Sheather, S. J. (2004). Density estimation. *Statistical science*, 588-597.
- [88] Roman, V. (2019). *Unsupervised machine learning: Clustering analysis. Towards Data Science*.
- [89] Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M., & Telgarsky, M. (2014). Tensor decompositions for learning latent variable models. *Journal of machine learning research*, 15, 2773-2832.
- [90] Barlow, H. B. (1989). Unsupervised learning. *Neural computation*, 1(3), 295-311.
- [91] Hecht-Nielsen, R. (1992). Theory of the backpropagation neural network. In *Neural networks for perception* (pp. 65-93). Academic Press.
- [92] Storkey, A. J., & Valabregue, R. (1999). The basins of attraction of a new Hopfield learning rule. *Neural Networks*, 12(6), 869-876.
- [93] Cercignani, C., & Cercignani, C. (1988). *The boltzmann equation* (pp. 40-103). Springer New York.
- [94] Hinton, G. E., Dayan, P., Frey, B. J., & Neal, R. M. (1995). The "wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214), 1158-1161.
- [95] Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518), 859-877.
- [96] Huang, K., Yang, H., King, I., & Lyu, M. R. (2004, June). Learning classifiers from imbalanced data based on biased minimax probability machine. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. (Vol. 2, pp. II-II). IEEE.
- [97] Chesta, C., Siohan, O., & Lee, C. H. (1999). Maximum a posteriori linear regression for hidden Markov model adaptation. In *Sixth European Conference on Speech Communication and Technology*.
- [98] Venugopal, D., & Gogate, V. (2012). On lifting the gibbs sampling algorithm. *Advances in Neural Information Processing Systems*, 25.
- [99] You, L., Tan, Q., Kang, Y., Xu, C., & Lin, C. (2018). Reconstruction and prediction of capillary pressure curve based on Particle Swarm Optimization-Back Propagation Neural Network method. *Petroleum*, 4(3), 268-280.
- [100] Robert, C. P., & Titterton, D. M. (1998). Reparameterization strategies for hidden Markov models and Bayesian approaches to maximum likelihood estimation. *Statistics and Computing*, 8, 145-158.

- [101] Husmeier, D., & Husmeier, D. (1999). Improved Training Scheme Combining the Expectation Maximisation (EM) Algorithm with the RVFL Approach. *Neural Networks for Conditional Probability Estimation: Forecasting Beyond Point Predictions*, 99-119.
- [102] Wilson, A. C., Recht, B., & Jordan, M. I. (2016). A Lyapunov analysis of momentum methods in optimization. *arXiv preprint arXiv:1611.02635*.
- [103] Test, C. S., & Uniqueness, R. C. (2018). Principal component analysis.
- [104] Stone, J. V. (2002). Independent component analysis: an introduction. *Trends in cognitive sciences*, 6(2), 59-64.
- [105] Lee, D., & Seung, H. S. (2000). Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13.
- [106] Corberi, F., & Sarracino, A. (2019). Probability distributions with singularities. *Entropy*, 21(3), 312.
- [107] Bank, D., Koenigstein, N., & Giryas, R. (2020). Autoencoders. *arXiv preprint arXiv:2003.05991*.
- [108] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. (2018). Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1), 53-65.
- [109] Liu, H., & Abbeel, P. (2021). Behavior from the void: Unsupervised active pre-training. *Advances in Neural Information Processing Systems*, 34, 18459-18473.
- [110] Wiering, M. A., & Van Otterlo, M. (2012). Reinforcement learning. *Adaptation, learning, and optimization*, 12(3), 729.
- [111] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.
- [112] Nachum, Ofir, et al. Bridging the gap between value and policy based reinforcement learning. *Advances in neural information processing systems*, 2017, 30.
- [113] Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014, January). Deterministic policy gradient algorithms. In *International conference on machine learning* (pp. 387-395). Pmlr.
- [114] Tedrake, Russ; Zhang, Teresa Weirui; SEUNG, H. Sebastian. Stochastic policy gradient reinforcement learning on a simple 3D biped. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566). IEEE, 2004. p. 2849-2854.
- [115] Du, S. S., Kakade, S. M., Wang, R., & Yang, L. F. (2019). Is a good representation sufficient for sample efficient reinforcement learning?. *arXiv preprint arXiv:1910.03016*.
- [116] Clavera, I., Rothfuss, J., Schulman, J., Fujita, Y., Asfour, T., & Abbeel, P. (2018, October). Model-based reinforcement learning via meta-policy optimization. In *Conference on Robot Learning* (pp. 617-629). PMLR.
- [117] Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine learning*, 8, 279-292.
- [118] Barron, E. N., & Ishii, H. (1989). The Bellman equation for minimizing the maximum cost. *NONLINEAR ANAL. THEORY METHODS APPLIC.*, 13(9), 1067-1090.
- [119] Roderick, Melrose; Macglashan, James; TELLEX, Stefanie. Implementing the deep q-network. *arXiv preprint arXiv:1711.07478*, 2017.
- [120] Sehnke, F., Osendorfer, C., Rückstieß, T., Graves, A., Peters, J., & Schmidhuber, J. (2010). Parameter-exploring policy gradients. *Neural Networks*, 23(4), 551-559.
- [121] Zhang, J., Kim, J., O'Donoghue, B., & Boyd, S. (2021, May). Sample efficient reinforcement learning with REINFORCE. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 35, No. 12, pp. 10887-10895).
- [122] Konda, Vijay; TSITSIKLIS, John. Actor-critic algorithms. *Advances in neural information processing systems*, 1999, 12.
- [123] Schulman, John, et al. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [124] Schulman, John, et al. Trust region policy optimization. In: *International conference on machine learning*. PMLR, 2015. p. 1889-1897.
- [125] Zhang, S., Wu, T., Pan, M., Zhang, C., & Yu, Y. (2020, October). A-SARSA: A predictive container auto-scaling algorithm based on reinforcement learning. In *2020 IEEE international conference on web services (ICWS)* (pp. 489-497). IEEE.
- [126] Tesauro, G. (1995). Temporal difference learning and TD-Gammon. *Communications of the ACM*, 38(3), 58-68.
- [127] Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., ... & Silver, D. (2018, April). Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 32, No. 1).
- [128] Ceron, J. S. O., & Castro, P. S. (2021, July). Revisiting rainbow: Promoting more insightful and inclusive deep reinforcement learning research. In *International Conference on Machine Learning* (pp. 1373-1383). PMLR.
- [129] Duffield, Nick; Lund, Carsten; Thorup, Mikkel. Priority sampling for estimation of arbitrary subset sums. *Journal of the ACM (JACM)*, 2007, 54.6: 32-es.
- [130] Pan, J., Wang, X., Cheng, Y., & Yu, Q. (2018). Multisource transfer double DQN based on actor learning. *IEEE transactions on neural networks and learning systems*, 29(6), 2227-2238.

- [131] Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., ... & Levine, S. (2018). Soft actor-critic algorithms and applications. arXiv preprint arXiv:1812.05905.
- [132] HE, Yu-Lin, et al. Determining the optimal temperature parameter for Softmax function in reinforcement learning. *Applied Soft Computing*, 2018, 70: 80-85.
- [133] SZARSKI, Martin; CHAUHAN, Sunita. Composite temperature profile and tooling optimization via Deep Reinforcement Learning. *Composites Part A: Applied Science and Manufacturing*, 2021, 142: 106235.
- [134] DABNEY, Will, et al. Distributional reinforcement learning with quantile regression. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2018.
- [135] DOLL, Bradley B.; SIMON, Dylan A.; DAW, Nathaniel D. The ubiquity of model-based reinforcement learning. *Current opinion in neurobiology*, 2012, 22.6: 1075-1081.
- [136] NASTESKI, V. (2017). An overview of the supervised machine learning methods. *Horizons*. b, 4, 51-62.
- [137] Jindal, N., & Liu, B. (2007, May). Review spam detection. In *Proceedings of the 16th international conference on World Wide Web* (pp. 1189-1190).
- [138] Lu, Y., & Han, J. (2003). Cancer classification using gene expression data. *Information Systems*, 28(4), 243-268.
- [139] THOMPSON, Bruce; BORRELLO, Gloria M. The importance of structure coefficients in regression research. *Educational and psychological measurement*, 1985, 45.2: 203-209.
- [140] Goebel, K., Saha, B., Saxena, A., Celaya, J. R., & Christophersen, J. P. (2008). Prognostics in battery health management. *IEEE instrumentation & measurement magazine*, 11(4), 33-40.
- [141] DHAVAL, Bhatti; DESHPANDE, Anuradha. Short-term load forecasting with using multiple linear regression. *International Journal of Electrical and Computer Engineering*, 2020, 10.4: 3911-3917.
- [142] Yan, X., Zhang, W., Ma, L., Liu, W., & Wu, Q. (2018). Parsimonious quantile regression of financial asset tail dynamics via sequential learning. *Advances in neural information processing systems*, 31.
- [143] ZHOU, Zhi-Hua. A brief introduction to weakly supervised learning. *National science review*, 2018, 5.1: 44-53.
- [144] Raczkowski, K., & Sadowski, P. (1990). Real function continuity. *Formalized Mathematics*, 1(4), 787-791.
- [145] NUNEZ, Eduardo; STEYERBERG, Ewout W.; NUNEZ, Julio. Regression modeling strategies. *Revista Española de Cardiología (English Edition)*, 2011, 64.6: 501-507.
- [146] Kim, M., Lee, S., & Kim, J. (2020, February). A wide & deep learning sharing input data for regression analysis. In *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)* (pp. 8-12). IEEE.
- [147] Riu, J., & Rius, F. X. (1997). Method comparison using regression with uncertainties in both axes. *TrAC Trends in Analytical Chemistry*, 16(4), 211-216.
- [148] KENDALL, Alex; CIPOLLA, Roberto. Modelling uncertainty in deep learning for camera relocalization. In: *2016 IEEE international conference on Robotics and Automation (ICRA)*. IEEE, 2016. p. 4762-4769.
- [149] JALAYER, Fatemeh; DE RISI, Raffaele; MANFREDI, Gaetano. Bayesian Cloud Analysis: efficient structural fragility assessment using linear regression. *Bulletin of Earthquake Engineering*, 2015, 13: 1183-1203.
- [150] JIMÉNEZ-CHACÓN, J.; ALVAREZ-PRIETO, M. Modelling uncertainty in a concentration range. *Accreditation and quality assurance*, 2009, 14: 15-27.
- [151] Allison, P. D. (1990). Change scores as dependent variables in regression analysis. *Sociological methodology*, 93-114.
- [152] Long, J. S., & Freese, J. (2006). Regression models for categorical dependent variables using Stata (Vol. 7). Stata press.
- [153] Cox, D. R., & Wermuth, N. (2014). *Multivariate dependencies: Models, analysis and interpretation* (Vol. 67). CRC Press.
- [154] SCHIELZETH, Holger. Simple means to improve the interpretability of regression coefficients. *Methods in Ecology and Evolution*, 2010, 1.2: 103-113.
- [155] KIM, Hyunjoong, et al. Visualizable and interpretable regression models with good prediction power. *IIE Transactions*, 2007, 39.6: 565-579.
- [156] HUANG, Yunhu, et al. Deep patch learning algorithms with high interpretability for regression problems. *International Journal of Intelligent Systems*, 2022, 37.11: 8239-8276.
- [157] KOOK, Lucas, et al. Deep and interpretable regression models for ordinal outcomes. *Pattern Recognition*, 2022, 122: 108263.
- [158] OTTE, Clemens. Safe and interpretable machine learning: A methodological review. *Computational intelligence in intelligent data analysis*, 2013, 111-122.
- [159] ALLEN, Michael Patrick. *Understanding regression analysis*. Springer Science & Business Media, 2004.s
- [160] NIMON, Kim F.; OSWALD, Frederick L. Understanding the results of multiple linear regression: Beyond standardized regression coefficients. *Organizational Research Methods*, 2013, 16.4: 650-674.
- [161] ÜSTÜN, B.; MELSSSEN, W. J.; BUYDENS, L. M. C. Visualisation and interpretation of support vector regression models. *Analytica chimica acta*, 2007, 595.1-2: 299-309.

- [162] Maniyar, D. M., & Nabney, I. T. (2004, September). Guiding local regression using visualisation. In International Workshop on Deterministic and Statistical Methods in Machine Learning (pp. 98-109). Berlin, Heidelberg: Springer Berlin Heidelberg.
- [163] GUEDJ, Benjamin; DESIKAN, Bhargav Srinivasa. Pycobra: A python toolbox for ensemble learning and visualisation. Journal of Machine Learning Research, 2018, 18.190: 1-5.
- [164] BRUNSDON, Chris. Visualisation of Regression Trees. Citeseer, 2007.
- [165] KARGAR, Mohammad Javad; HANIFIZADE, Alireza. Automation of regression test in microservice architecture. In: 2018 4th International Conference on Web Research (ICWR). IEEE, 2018. p. 133-137.
- [166] VANHEEL, Frank, et al. Automated linear regression tools improve rssi wsn localization in multipath indoor environment. EURASIP Journal on Wireless Communications and Networking, 2011, 2011.1: 1-27.
- [167] LYNCH, Jeffrey James. The automation of regression modeling in osteometric sorting: An ordination approach. Journal of forensic sciences, 2018, 63.3: 798-804.
- [168] MEMON, Atif M. Automated GUI regression testing using AI planning. In: Artificial Intelligence Methods In Software Testing. 2004. p. 51-99.
- [169] FELSING, Dennis, et al. Automating regression verification. In: Proceedings of the 29th ACM/IEEE international conference on Automated software engineering. 2014. p. 349-360.
- [170] CZAJKOWSKI, M., & KRETOWSKI, M. (2016). *The role of decision tree representation in regression problems—An evolutionary perspective*. Applied Soft Computing, 48, 458-475.
- [171] JORDAN, Michael I.; MITCHELL, Tom M. Machine learning: Trends, perspectives, and prospects. Science, 2015, 349.6245: 255-260.
- [172] TULABANDHULA, Theja; RUDIN, Cynthia. Machine learning with operational costs. Journal of Machine Learning Research, 2013, 14.7.
- [173] Quinlan, J. R. (1987). Simplifying decision trees. International journal of man-machine studies, 27(3), 221-234.
- [174] KOTSIANTIS, Sotiris B. Decision trees: a recent overview. Artificial Intelligence Review, 2013, 39: 261-283.
- [175] GRĄBCZEWSKI, Krzysztof; DUCH, Włodzisław. Heterogeneous forests of decision trees. In: International Conference on Artificial Neural Networks. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. p. 504-509.
- [176] MAULUD, Dastan; ABDULAZEEZ, Adnan M. A review on linear regression comprehensive in machine learning. Journal of Applied Science and Technology Trends, 2020, 1.4: 140-147.
- [177] LIVINGSTONE, David J.; SALT, David W. Judging the significance of multiple linear regression models. Journal of Medicinal Chemistry, 2005, 48.3: 661-663.
- [178] KHINE, Kyi Lai Lai; NYUNT, Thi Thi Soe. Predictive big data analytics using multiple linear regression model. In: Big Data Analysis and Deep Learning Applications: Proceedings of the First International Conference on Big Data Analysis and Deep Learning 1st. Springer Singapore, 2019. p. 9-19.
- [179] FAN, Jianqing; HAN, Fang; LIU, Han. Challenges of big data analysis. National science review, 2014, 1.2: 293-314.
- [180] JOBSON, J. D. Multiple linear regression. Applied multivariate data analysis: Regression and experimental design, 1991, 219-398.
- [181] SONG, Yunsheng, et al. An efficient instance selection algorithm for k nearest neighbor regression. Neurocomputing, 2017, 251: 26-34.
- [182] TER BRAAK, Cajo JF. Non-linear methods for multivariate statistical calibration and their use in palaeoecology: a comparison of inverse (k-nearest neighbours, partial least squares and weighted averaging partial least squares) and classical approaches. Chemometrics and Intelligent Laboratory Systems, 1995, 28.1: 165-180.
- [183] LAKSHMI, S. Venkata; PRABAKARAN, T. Edwin. Application of k-nearest neighbour classification method for intrusion detection in network data. International Journal of Computer Applications, 2014, 97.7.
- [184] SUN, Shiliang; HUANG, Rongqing. An adaptive k-nearest neighbor algorithm. In: 2010 seventh international conference on fuzzy systems and knowledge discovery. IEEE, 2010. p. 91-94.
- [185] SAMET, Hanan. K-nearest neighbor finding using MaxNearestDist. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 30.2: 243-252.
- [186] BEZDEK, James C.; CHUAH, Siew K.; LEEP, David. Generalized k-nearest neighbor rules. Fuzzy Sets and Systems, 1986, 18.3: 237-256.
- [187] LAROSE, Daniel T.; LAROSE, Chantal D. k-nearest neighbor algorithm. 2014.
- [188] SOZOU, Peter D., et al. Non-linear point distribution modelling using a multi-layer perceptron. Image and Vision Computing, 1997, 15.6: 457-463.

- [189] ROSSI, Fabrice; CONAN-GUEZ, Brieuc. Functional multi-layer perceptron: a non-linear tool for functional data analysis. *Neural networks*, 2005, 18.1: 45-60.
- [190] PARLOS, Alexander G.; CHONG, Kil To; ATIYA, Amir F. Application of the recurrent multilayer perceptron in modeling complex process dynamics. *IEEE Transactions on Neural Networks*, 1994, 5.2: 255-266.
- [191] ARENA, Paolo, et al. Multilayer perceptrons to approximate complex valued functions. *International Journal of Neural Systems*, 1995, 6.04: 435-446.
- [192] LUO, Xiong; CHANG, Xiaohui; BAN, Xiaojuan. Regression and classification using extreme learning machine based on L1-norm and L2-norm. *Neurocomputing*, 2016, 174: 179-186.
- [193] GOEMAN, Jelle; MEIJER, Rosa; CHATURVEDI, Nimisha. L1 and L2 penalized regression models. *cran. r-project. or*, 2012.
- [194] WANG, Haifeng; HU, Dejin. Comparison of SVM and LS-SVM for regression. In: *2005 International conference on neural networks and brain*. IEEE, 2005. p. 279-283.
- [195] SHAWE-TAYLOR, John; SUN, Shiliang. A review of optimization methodologies in support vector machines. *Neurocomputing*, 2011, 74.17: 3609-3618.
- [196] YUE, Yisong, et al. A support vector method for optimizing average precision. In: *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. 2007. p. 271-278.
- [197] LEE, Yuh-Jye; MANGASARIAN, Olvi L. RSVM: Reduced support vector machines. In: *Proceedings of the 2001 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 2001. p. 1-17.
- [198] FINE, Shai; SCHEINBERG, Katya. Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, 2001, 2.Dec: 243-264.
- [199] RUIZ, Alberto; LÓPEZ-DE-TERUEL, Pedro E. Nonlinear kernel-based statistical pattern analysis. *IEEE Transactions on Neural Networks*, 2001, 12.1: 16-32.
- [200] ABDIANSAH, Abdiansah; WARDOYO, Retantyo. Time complexity analysis of support vector machines (SVM) in LibSVM. *Int. J. Comput. Appl*, 2015, 128.3: 28-34.
- [201] JOACHIMS, Thorsten. Training linear SVMs in linear time. In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2006. p. 217-226.
- [202] RÜPING, Stefan. SVM kernels for time series analysis. Technical report, 2001.
- [203] AKBANI, Rehan; KWEK, Stephen; JAPKOWICZ, Nathalie. Applying support vector machines to imbalanced datasets. In: *Machine Learning: ECML 2004: 15th European Conference on Machine Learning, Pisa, Italy, September 20-24, 2004. Proceedings 15*. Springer Berlin Heidelberg, 2004. p. 39-50.
- [204] LAVALLEY, Michael P. Logistic regression. *Circulation*, 2008, 117.18: 2395-2399.
- [205] WRIGHT, Raymond E. Logistic regression. 1995.
- [206] NICK, Todd G.; CAMPBELL, Kathleen M. Logistic regression. *Topics in biostatistics*, 2007, 273-301.
- [207] LE CESSIE, Saskia; VAN HOUWELINGEN, J. C. Logistic regression for correlated binary data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 1994, 43.1: 95-108.
- [208] MAGDER, Laurence S.; HUGHES, James P. Logistic regression when the outcome is measured with uncertainty. *American journal of epidemiology*, 1997, 146.2: 195-203.
- [209] STOLTZFUS, Jill C. Logistic regression: a brief primer. *Academic emergency medicine*, 2011, 18.10: 1099-1104.
- [210] TRIPEPI, Giovanni, et al. Linear and logistic regression analysis. *Kidney international*, 2008, 73.7: 806-810.
- [211] LUO, Xiong; CHANG, Xiaohui; BAN, Xiaojuan. Regression and classification using extreme learning machine based on L1-norm and L2-norm. *Neurocomputing*, 2016, 174: 179-186.
- [212] LIU, Han; WANG, Lie; ZHAO, Tuo. Multivariate regression with calibration. *Advances in neural information processing systems*, 2014, 27.
- [213] DEMIR-KAVUK, Ozgur, et al. Prediction using step-wise L1, L2 regularization and feature selection for small data sets with large number of features. *BMC bioinformatics*, 2011, 12: 1-10.
- [214] OSTERTAGOVÁ, Eva. Modelling using polynomial regression. *Procedia Engineering*, 2012, 48: 500-506.
- [215] HAWKINS, Douglas M. The problem of overfitting. *Journal of chemical information and computer sciences*, 2004, 44.1: 1-12.
- [216] KOEHRSEN, Will. Overfitting vs. underfitting: A complete example. *Towards Data Science*, 2018, 405.
- [217] WAN, Xing. The effect of regularization coefficient on polynomial regression. In: *Journal of Physics: Conference Series*. IOP Publishing, 2019. p. 042054.
- [218] CHENG, Xi, et al. Polynomial regression as an alternative to neural nets. *arXiv preprint arXiv:1806.06850*, 2018.
- [219] BILBAO, Imanol; BILBAO, Javier. Overfitting problem and the over-training in the era of data: Particularly for Artificial Neural Networks. In: *2017 eighth international conference on intelligent computing and information systems (ICICIS)*. IEEE, 2017. p. 173-177.

- [220] PRICE, Eric; SILWAL, Sandeep; ZHOU, Samson. Hardness and algorithms for robust and sparse optimization. In: International Conference on Machine Learning. PMLR, 2022. p. 17926-17944.
- [221] QASIM, Muhammad; AMIN, Muhammad; AMANULLAH, Muhammad. On the performance of some new Liu parameters for the gamma regression model. *Journal of Statistical Computation and Simulation*, 2018, 88.16: 3065-3080.
- [222] BOUJENANE, Ismail; AIMANI, JALILA EL. Incidence and occurrence time of clinical mastitis in Holstein cows. *Turkish Journal of Veterinary & Animal Sciences*, 2015, 39.1: 42-49.
- [223] LUKMAN, Adewale F., et al. A new ridge-type estimator for the gamma regression model. *Scientifica*, 2021, 2021: 1-8.
- [224] ALGAMAL, Zakariya Yahya. Shrinkage estimators for gamma regression model. *Electronic Journal of Applied Statistical Analysis*, 2018, 11.1: 253-268.
- [225] AMIN, Muhammad, et al. Performance of some ridge estimators for the gamma regression model. *Statistical papers*, 2020, 61: 997-1026.
- [226] RAHAYU, Anita, et al. Multivariate gamma regression: Parameter estimation, hypothesis testing, and its application. *Symmetry*, 2020, 12.5: 813.
- [227] HANS, Chris. Elastic net regression modeling with the orthant normal prior. *Journal of the American Statistical Association*, 2011, 106.496: 1383-1393.
- [228] ZHANG, Zheng, et al. Discriminative elastic-net regularized linear regression. *IEEE Transactions on Image Processing*, 2017, 26.3: 1466-1481.
- [229] OGUTU, Joseph O.; SCHULZ-STREECK, Torben; PIEPHO, Hans-Peter. Genomic selection using regularized linear regression models: ridge regression, lasso, elastic net and their extensions. In: *BMC proceedings*. BioMed Central, 2012. p. 1-6.
- [230] TOWNSEND, Wilbur. ELASTICREGRESS: Stata module to perform elastic net regression, lasso regression, ridge regression. 2018.
- [231] HAO, Lingxin; NAIMAN, Daniel Q. Quantile regression. Sage, 2007.
- [232] KOENKER, Roger; HALLOCK, Kevin F. Quantile regression. *Journal of economic perspectives*, 2001, 15.4: 143-156.
- [233] YU, Keming; LU, Zudi; STANDER, Julian. Quantile regression: applications and current research areas. *Journal of the Royal Statistical Society Series D: The Statistician*, 2003, 52.3: 331-350.
- [234] KOENKER, Roger, et al. (ed.). *Handbook of quantile regression*. 2017.
- [235] KOENKER, Roger. Quantile regression: 40 years on. *Annual review of economics*, 2017, 9: 155-176.
- [236] KUDRYAVTSEV, Andrey A. Using quantile regression for rate-making. *Insurance: Mathematics and Economics*, 2009, 45.2: 296-304.
- [237] HAYAT, Matthew J.; HIGGINS, Melinda. Understanding poisson regression. *Journal of Nursing Education*, 2014, 53.4: 207-215.
- [238] CONSUL, PoC; FAMOYE, Felix. Generalized Poisson regression model. *Communications in Statistics-Theory and Methods*, 1992, 21.1: 89-109.
- [239] FROME, Edward L. The analysis of rates using Poisson regression models. *Biometrics*, 1983, 665-674.
- [240] FAMOYE, Felix; WULU, John T.; SINGH, Karan P. On the generalized Poisson regression model with an application to accident data. *Journal of Data Science*, 2004, 2.3: 287-295.
- [241] FAMOYE, Felix. Restricted generalized Poisson regression model. *Communications in Statistics-Theory and Methods*, 1993, 22.5: 1335-1354.
- [242] CHRISTIANSEN, Cindy L.; MORRIS, Carl N. Hierarchical Poisson regression modeling. *Journal of the American Statistical Association*, 1997, 92.438: 618-632.
- [243] SCHEMPER, Michael; HENDERSON, Robin. Predictive accuracy and explained variation in Cox regression. *Biometrics*, 2000, 56.1: 249-255.
- [244] STEL, Vianda S., et al. Survival analysis II: Cox regression. *Nephron Clinical Practice*, 2011, 119.3: c255-c260.
- [245] THOMAS, Laine; REYES, Eric M. Tutorial: survival estimation for Cox regression models with time-varying coefficients using SAS and R. *Journal of Statistical Software*, 2014, 61: 1-23.
- [246] BENÍTEZ-PAREJO, N.; DEL ÁGUILA, MM Rodríguez; PÉREZ-VICENTE, S. Survival analysis and Cox regression. *Allergologia et immunopathologia*, 2011, 39.6: 362-373.
- [247] LUNN, Mary; MCNEIL, Don. Applying Cox regression to competing risks. *Biometrics*, 1995, 524-532.
- [248] ZHANG, Zhongheng, et al. Time-varying covariates and coefficients in Cox regression models. *Annals of translational medicine*, 2018, 6.7.
- [249] LIN, D. Y. Cox regression analysis of multivariate failure time data: the marginal approach. *Statistics in medicine*, 1994, 13.21: 2233-2247.

- [250] D'AGOSTINO, Ralph B., et al. Relation of pooled logistic regression to time dependent Cox regression analysis: the Framingham Heart Study. *Statistics in medicine*, 1990, 9.12: 1501-1515.
- [251] RINALDO, Alessandro. Properties and refinements of the fused lasso. 2009.
- [252] BENTO, Jose; FURMANIAK, Ralph; RAY, Surjyendu. On the complexity of the weighted fused lasso. *IEEE Signal Processing Letters*, 2018, 25.10: 1595-1599.
- [253] CHEN, Xi, et al. Graph-structured multi-task regression and an efficient optimization method for general fused lasso. arXiv preprint arXiv:1005.3579, 2010.
- [254] HOEFLING, Holger. A path algorithm for the fused lasso signal approximator. *Journal of Computational and Graphical Statistics*, 2010, 19.4: 984-1006.
- [255] TIBSHIRANI, Robert, et al. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 2005, 67.1: 91-108.
- [256] DALLAKYAN, Aramays; POURAHMADI, Mohsen. Fused-lasso regularized Cholesky factors of large nonstationary covariance matrices of replicated time series. *Journal of Computational and Graphical Statistics*, 2023, 32.1: 157-170.
- [257] XIN, Bo, et al. Efficient generalized fused lasso and its application to the diagnosis of Alzheimer's disease. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2014.
- [258] HOEFLING, Holger. A path algorithm for the fused lasso signal approximator. *Journal of Computational and Graphical Statistics*, 2010, 19.4: 984-1006.
- [259] BETANCOURT, Brenda; RODRÍGUEZ, Abel; BOYD, Naomi. Bayesian fused lasso regression for dynamic binary networks. *Journal of Computational and Graphical Statistics*, 2017, 26.4: 840-850.
- [260] YERLIKAYA-OZKURT, Fatma; TAYLAN, Pakize; TEZ, Müjgan. Spline based Sparseness and Smoothness for Partially Nonlinear Model via C-fused Lasso. In: *Book of Abstracts*. p. 75.
- [261] SUN, Qiang; ZHOU, Wen-Xin; FAN, Jianqing. Adaptive huber regression. *Journal of the American Statistical Association*, 2020, 115.529: 254-265.
- [262] TSAKONAS, Efthymios, et al. Convergence of the huber regression m-estimate in the presence of dense outliers. *IEEE Signal Processing Letters*, 2014, 21.10: 1211-1214.
- [263] D'ORSI, Tommaso; NOVIKOV, Gleb; STEURER, David. Consistent regression when oblivious outliers overwhelm. In: *International Conference on Machine Learning*. PMLR, 2021. p. 2297-2306.
- [264] JOHANSEN, Søren; NIELSEN, Bent. Outlier detection in regression using an iterated one-step approximation to the Huber-skip estimator. *Econometrics*, 2013, 1.1: 53-70.
- [265] LI, Guoying. Robust regression. *Exploring data tables, trends, and shapes*, 1985, 281: U340.
- [266] ROUSSEEUW, Peter J.; LEROY, Annick M. Robust regression and outlier detection. John wiley & sons, 2005.
- [267] ALMA, Özlem Gürünlü. Comparison of robust regression methods in linear regression. *Int. J. Contemp. Math. Sciences*, 2011, 6.9: 409-421.
- [268] KLIVANS, Adam; KOTHARI, Pravesh K.; MEKA, Raghu. Efficient algorithms for outlier-robust regression. In: *Conference On Learning Theory*. PMLR, 2018. p. 1420-1430.
- [269] BERKEY, Catherine S., et al. A random-effects regression model for meta-analysis. *Statistics in medicine*, 1995, 14.4: 395-411.
- [270] NORBERG, Ragnar. Hierarchical credibility: analysis of a random effect linear model with nested classification. *Scandinavian Actuarial Journal*, 1986, 1986.3-4: 204-222.
- [271] MCNEISH, Daniel; STAPLETON, Laura M.; SILVERMAN, Rebecca D. On the unnecessary ubiquity of hierarchical linear modeling. *Psychological methods*, 2017, 22.1: 114.
- [272] ROTH, Tobias, et al. Hierarchical change-point regression models including random effects to estimate empirical critical loads for nitrogen using Bayesian Regression Models (brms) and JAGS. *MethodsX*, 2022, 9: 101902.
- [273] MCCOACH, D. Betsy. Hierarchical linear modeling. *The reviewer's guide to quantitative methods in the social sciences*, 2010, 123-140.
- [274] MENTRE, France; MALLET, Alain; BACCAR, Doha. Optimal design in random-effects regression models. *Biometrika*, 1997, 84.2: 429-442.
- [275] FRIEDMAN, Jerome H. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 2001, 1189-1232.
- [276] FREEMAN, Elizabeth A., et al. Random forests and stochastic gradient boosting for predicting tree canopy cover: comparing tuning processes and model performance. *Canadian Journal of Forest Research*, 2016, 46.3: 323-339.
- [277] BOLDINI, Davide, et al. Tuning gradient boosting for imbalanced bioassay modelling with custom loss functions. *Journal of Cheminformatics*, 2022, 14.1: 1-13.

- [278] BARTZ-BEIELSTEIN, Thomas; CHANDRASEKARAN, Sowmya; REHBACH, Frederik. Case study II: tuning of gradient boosting (xgboost). In: *Hyperparameter Tuning for Machine and Deep Learning with R: A Practical Guide*. Singapore: Springer Nature Singapore, 2023. p. 221-234.
- [279] ALHAKEEM, Zaineb M., et al. Prediction of ecofriendly concrete compressive strength using gradient boosting regression tree combined with GridSearchCV hyperparameter-optimization techniques. *Materials*, 2022, 15.21: 7432.
- [280] CHEN, Yifei, et al. A gradient boosting algorithm for survival analysis via direct optimization of concordance index. *Computational and mathematical methods in medicine*, 2013, 2013.
- [281] LIU, Jiaming, et al. Interpreting the prediction results of the tree-based gradient boosting models for financial distress prediction with an explainable machine learning approach. *Journal of Forecasting*, 2023, 42.5: 1112-1137.
- [282] XIA, Yulu, et al. GBMVis: Visual analytics for interpreting gradient boosting machine. In: *Cooperative Design, Visualization, and Engineering: 18th International Conference, CDVE 2021, Virtual Event, October 24–27, 2021, Proceedings 18*. Springer International Publishing, 2021. p. 63-72.
- [283] PRETTENHOFER, Peter; LOUPPE, Gilles. Gradient boosted regression trees in scikit-learn. In: *PyData 2014*. 2014.
- [284] BUNEA, Florentina; TSYBAKOV, Alexandre B.; WEGKAMP, Marten H. Aggregation for Gaussian regression. 2007.
- [285] MARREL, Amandine, et al. An efficient methodology for modeling complex computer codes with Gaussian processes. *Computational Statistics & Data Analysis*, 2008, 52.10: 4731-4744.
- [286] GIBBS, Mark N. Bayesian Gaussian processes for regression and classification. 1998. PhD Thesis. University of Cambridge.
- [287] MUSCHINSKI, Thomas, et al. Cholesky-based multivariate Gaussian regression. *Econometrics and Statistics*, 2022.
- [288] SHUMWAY, Robert H., et al. ARIMA models. *Time series analysis and its applications: with R examples*, 2017, 75-163.
- [289] Xu, M., Watanachaturaporn, P., Varshney, P. K., & Arora, M. K. (2005). Decision tree regression for soft classification of remote sensing data. *Remote Sensing of Environment*, 97(3), 322-336.
- [290] Rathore, S. S., & Kumar, S. (2016). A decision tree regression based approach for the number of software faults prediction. *ACM SIGSOFT Software Engineering Notes*, 41(1), 1-6.
- [291] Pekel, E. (2020). Estimation of soil moisture using decision tree regression. *Theoretical and Applied Climatology*, 139(3-4), 1111-1119.
- [292] Tso, G. K., & Yau, K. K. (2007). Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks. *Energy*, 32(9), 1761-1768.
- [293] Thomas, E. H., & Galambos, N. (2004). What satisfies students? Mining student-opinion data with regression and decision tree analysis. *Research in Higher Education*, 45, 251-269.
- [294] Long, W. J., Griffith, J. L., Selker, H. P., & D'agostino, R. B. (1993). A comparison of logistic regression to decision-tree induction in a medical domain. *Computers and Biomedical Research*, 26(1), 74-97.
- [295] Czajkowski, M., & Kretowski, M. (2016). The role of decision tree representation in regression problems—An evolutionary perspective. *Applied soft computing*, 48, 458-475.
- [296] Ibrahim, Z., & Rusli, D. (2007, September). Predicting students' academic performance: comparing artificial neural network, decision tree and linear regression. In *21st Annual SAS Malaysia Forum*, 5th September.
- [297] Swetapadma, A., & Yadav, A. (2016). A novel decision tree regression-based fault distance estimation scheme for transmission lines. *IEEE transactions on power delivery*, 32(1), 234-245.
- [298] Westreich, D., Lessler, J., & Funk, M. J. (2010). Propensity score estimation: neural networks, support vector machines, decision trees (CART), and meta-classifiers as alternatives to logistic regression. *Journal of clinical epidemiology*, 63(8), 826-833.
- [299] Pathak, S., Mishra, I., & Swetapadma, A. (2018, November). An assessment of decision tree based classification and regression algorithms. In *2018 3rd International Conference on Inventive Computation Technologies (ICICT)* (pp. 92-95). IEEE.
- [300] Aiken, L. S., West, S. G., & Pitts, S. C. (2003). Multiple linear regression. *Handbook of psychology*, 481-507.
- [301] Eberly, L. E. (2007). Multiple linear regression. *Topics in Biostatistics*, 165-187.
- [302] Tranmer, M., & Elliot, M. (2008). Multiple linear regression. *The Cathie Marsh Centre for Census and Survey Research (CCSR)*, 5(5), 1-5.
- [303] Uyanik, G. K., & Güler, N. (2013). A study on multiple linear regression analysis. *Procedia-Social and Behavioral Sciences*, 106, 234-240.
- [304] Olive, D. J., & Olive, D. J. (2017). *Multiple linear regression* (pp. 17-83). Springer International Publishing.
- [305] Grégoire, G. (2014). *Multiple linear regression*. *European Astronomical Society Publications Series*, 66, 45-72.
- [306] Andrews, D. F. (1974). A robust method for multiple linear regression. *Technometrics*, 16(4), 523-531.
- [307] Jobson, J. D., & Jobson, J. D. (1991). *Multiple linear regression*. *Applied multivariate data analysis: Regression and experimental design*, 219-398.

- [308] Marill, K. A. (2004). Advanced statistics: linear regression, part II: multiple linear regression. *Academic emergency medicine*, 11(1), 94-102.
- [309] Bottenberg, R. A., & Ward, J. H. (1963). Applied multiple linear regression (Vol. 63, No. 6). 6570th Personnel Research Laboratory, Aerospace Medical Division, Air Force Systems Command, Lackland Air Force Base.
- [310] Song, Y., Liang, J., Lu, J., & Zhao, X. (2017). An efficient instance selection algorithm for k nearest neighbor regression. *Neurocomputing*, 251, 26-34.
- [311] Maltamo, M., & Kangas, A. (1998). Methods based on k-nearest neighbor regression in the prediction of basal area diameter distribution. *Canadian Journal of Forest Research*, 28(8), 1107-1115.
- [312] Hu, C., Jain, G., Zhang, P., Schmidt, C., Gomadam, P., & Goraka, T. (2014). Data-driven method based on particle swarm optimization and k-nearest neighbor regression for estimating capacity of lithium-ion battery. *Applied Energy*, 129, 49-55.
- [313] Lian, H. (2011). Convergence of functional k-nearest neighbor regression estimate with functional responses.
- [314] Kramer, O. (2011). Unsupervised K-nearest neighbor regression. arXiv preprint arXiv:1107.3600.
- [315] Nguyen, B., Morell, C., & De Baets, B. (2016). Large-scale distance metric learning for k-nearest neighbors regression. *Neurocomputing*, 214, 805-814.
- [316] Kramer, O. (2011, December). Dimensionality reduction by unsupervised k-nearest neighbor regression. In 2011 10th international conference on machine learning and applications and workshops (Vol. 1, pp. 275-278). IEEE.
- [317] Biau, G., Devroye, L., Dujmović, V., & Krzyżak, A. (2012). An affine invariant k-nearest neighbor regression estimate. *Journal of Multivariate Analysis*, 112, 24-34.
- [318] Farahnakian, F., Pahikkala, T., Liljeberg, P., & Plosila, J. (2013, December). Energy aware consolidation algorithm based on k-nearest neighbor regression for cloud data centers. In 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing (pp. 256-259). IEEE.
- [319] Huang, J., & Perry, M. (2016). A semi-empirical approach using gradient boosting and k-nearest neighbors regression for GEFCom2014 probabilistic solar power forecasting. *International Journal of Forecasting*, 32(3), 1081-1086.
- [320] Murtagh, F. (1991). Multilayer perceptrons for classification and regression. *Neurocomputing*, 2(5-6), 183-197.
- [321] Gaudart, J., Giusiano, B., & Huiart, L. (2004). Comparison of the performance of multi-layer perceptron and linear regression for epidemiological data. *Computational statistics & data analysis*, 44(4), 547-570.
- [322] Nassif, A. B., Ho, D., & Capretz, L. F. (2013). Towards an early software estimation using log-linear regression and a multilayer perceptron model. *Journal of Systems and Software*, 86(1), 144-160.
- [323] Agirre-Basurko, E., Ibarra-Berastegi, G., & Madariaga, I. (2006). Regression and multilayer perceptron-based models to forecast hourly O3 and NO2 levels in the Bilbao area. *Environmental Modelling & Software*, 21(4), 430-446.
- [324] Bagga, S., Goyal, A., Gupta, N., & Goyal, A. (2020). Credit card fraud detection using pipeling and ensemble learning. *Procedia Computer Science*, 173, 104-112.
- [325] You, J., Tulpan, D., Malpass, M. C., & Ellis, J. L. (2022). Using machine learning regression models to predict the pellet quality of pelleted feeds. *Animal Feed Science and Technology*, 293, 115443.
- [326] Manaswi, N. K., & Manaswi, N. K. (2018). Regression to MLP in Keras. *Deep Learning with Applications Using Python: Chatbots and Face, Object, and Speech Recognition With TensorFlow and Keras*, 69-89.
- [327] Mielniczuk, J., & Tyrcha, J. (1993). Consistency of multilayer perceptron regression estimators. *Neural Networks*, 6(7), 1019-1022.
- [328] Widyahastuti, F., & Tjhin, V. U. (2017, July). Predicting students performance in final examination using linear regression and multilayer perceptron. In 2017 10th International Conference on Human System Interactions (HSI) (pp. 188-192). IEEE.
- [329] Zhu, M., & Wang, L. (2010, July). Intelligent trading using support vector regression and multilayer perceptrons optimized with genetic algorithms. In The 2010 international joint conference on neural networks (IJCNN) (pp. 1-5). IEEE.
- [330] Gilroy, E. J., Hirsch, R. M., & Cohn, T. A. (1990). Mean square error of regression-based constituent transport estimates. *Water Resources Research*, 26(9), 2069-2077.
- [331] Chaudhuri, P., Huang, M. C., Loh, W. Y., & Yao, R. (1994). Piecewise-polynomial regression trees. *Statistica Sinica*, 143-167.
- [332] Razi, M. A., & Athappilly, K. (2005). A comparative predictive analysis of neural networks (NNs), nonlinear regression and classification and regression tree (CART) models. *Expert systems with applications*, 29(1), 65-74.
- [333] Zhang, B., Wei, Z., Ren, J., Cheng, Y., & Zheng, Z. (2018). An empirical study on predicting blood pressure using classification and regression trees. *IEEE access*, 6, 21758-21768.
- [334] Ghazali, F. M. M., Ahmad, W. M. A. W., Srivastava, K. C., Shrivastava, D., Noor, N. F. M., Akbar, N. A. N., ... & Alam, M. K. (2021). A study of creatinine level among patients with dyslipidemia and type 2 diabetes mellitus using multilayer perceptron and multiple linear regression. *Journal of pharmacy & bioallied sciences*, 13(Suppl 1), S795.
- [335] Rynkiewicz, J. (2012). General bound of overfitting for MLP regression models. *Neurocomputing*, 90, 106-110.

- [336] Browne, M. W. (1975). A comparison of single sample and cross-validation methods for estimating the mean squared error of prediction in multiple linear regression. *British Journal of Mathematical and Statistical Psychology*, 28(1), 112-120.
- [337] Williams, C. G., & Ojuri, O. O. (2021). Predictive modelling of soils' hydraulic conductivity using artificial neural network and multiple linear regression. *SN Applied Sciences*, 3, 1-13.
- [338] Bakar, N. M. A., & Tahir, I. M. (2009). Applying multiple linear regression and neural network to predict bank performance. *International Business Research*, 2(4), 176-183.s
- [339] Prayudani, S., Hizriadi, A., Lase, Y. Y., & Fatmi, Y. (2019, November). Analysis accuracy of forecasting measurement technique on random K-nearest neighbor (RKNN) using MAPE and MSE. In *Journal of Physics: Conference Series* (Vol. 1361, No. 1, p. 012089). IOP Publishing.
- [340] Airen, S., & Agrawal, J. (2022). Movie recommender system using k-nearest neighbors variants. *National Academy Science Letters*, 45(1), 75-82.
- [341] Mean Absolute Error. Retrieved September, 19, 2016.
- [342] Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE). *Geoscientific model development discussions*, 7(1), 1525-1534.
- [343] Hodson, T. O. (2022). Root-mean-square error (RMSE) or mean absolute error (MAE): When to use them or not. *Geoscientific Model Development*, 15(14), 5481-5487.
- [344] Karunasingha, D. S. K. (2022). Root mean square error or mean absolute error? Use their ratio as well. *Information Sciences*, 585, 609-629.
- [345] Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature. *Geoscientific model development*, 7(3), 1247-1250.
- [346] Cleger-Tamayo, S., Fernández-Luna, J. M., & Huete, J. F. (2012, September). On the Use of Weighted Mean Absolute Error in Recommender Systems. In *RUE@ RecSys* (pp. 24-26).
- [347] Camero, A., Toutouh, J., & Alba, E. (2019). A specialized evolutionary strategy using mean absolute error random sampling to design recurrent neural networks. *arXiv preprint arXiv:1909.02425*, 1-10.
- [348] Robeson, S. M., & Willmott, C. J. (2023). Decomposition of the mean absolute error (MAE) into systematic and unsystematic components. *Plos one*, 18(2), e0279774.
- [349] Langdon, W. B., Dolado, J., Sarro, F., & Harman, M. (2016). Exact mean absolute error of baseline predictor, MARPO. *Information and Software Technology*, 73, 16-18.
- [350] Yang, L., Liu, S., Tsoka, S., & Papageorgiou, L. G. (2017). A regression tree approach using mathematical programming. *Expert Systems with Applications*, 78, 347-357.
- [351] Saha, S., Arabameri, A., Saha, A., Blaschke, T., Ngo, P. T. T., Nhu, V. H., & Band, S. S. (2021). Prediction of landslide susceptibility in Rudraprayag, India using novel ensemble of conditional probability and boosted regression tree-based on cross-validation method. *Science of the total environment*, 764, 142928.
- [352] Suyono, H., Hasanah, R. N., Setyawan, R. A., Mudjirahardjo, P., Wijoyo, A., & Musirin, I. (2018). Comparison of solar radiation intensity forecasting using ANFIS and multiple linear regression methods. *Bulletin of Electrical Engineering and Informatics*, 7(2), 191-198.
- [353] Niazian, M., Sadat-Noori, S. A., & Abdipour, M. (2018). Modeling the seed yield of Ajowan (*Trachyspermum ammi* L.) using artificial neural network and multiple linear regression models. *Industrial Crops and Products*, 117, 224-234.
- [354] Monburinon, N., Chertchom, P., Kaewkiriya, T., Rungpheung, S., Buya, S., & Boonpou, P. (2018, May). Prediction of prices for used car by using regression models. In *2018 5th International Conference on Business and Industrial Research (ICBIR)* (pp. 115-119). IEEE.
- [355] Hatipoğlu, Ş., Belgrat, M. A., Degirmenci, A., & Karal, Ö. (2021, October). Prediction of Unemployment Rates in Turkey by k-Nearest Neighbor Regression Analysis. In *2021 Innovations in Intelligent Systems and Applications Conference (ASYU)* (pp. 1-5). IEEE.
- [356] Sinta, D., Wijayanto, H., & Sartono, B. J. A. M. S. (2014). Ensemble K-Nearest neighbors method to predict rice price in Indonesia. *Appl. Math. Sci*, 8(160), 7993-8005.
- [357] Zriaa, R., & Amali, S. (2021). A comparative study between k-nearest neighbors and k-means clustering techniques of collaborative filtering in e-learning environment. In *Innovations in Smart Cities Applications Volume 4: The Proceedings of the 5th International Conference on Smart City Applications* (pp. 268-282). Springer International Publishing.
- [358] Pham, B. T., Nguyen, M. D., Bui, K. T. T., Prakash, I., Chapi, K., & Bui, D. T. (2019). A novel artificial intelligence approach based on Multi-layer Perceptron Neural Network and Biogeography-based Optimization for predicting coefficient of consolidation of soil. *Catena*, 173, 302-311.

- [359] Xu, Y., Li, F., & Asgari, A. (2022). Prediction and optimization of heating and cooling loads in a residential building based on multi-layer perceptron neural network and different optimization algorithms. *Energy*, 240, 122692.
- [360] Meshram, S. G., Meshram, C., Pourhosseini, F. A., Hasan, M. A., & Islam, S. (2022). A multi-layer perceptron (MLP)-Fire fly algorithm (FFA)-based model for sediment prediction. *Soft Computing*, 1-10.
- [361] AgriMetSoft (2019). Online Calculators. Available on: <https://agrimetsoft.com/calculators/R-squared%20correlation>
- [362] Sutrick, K. (2017). Teaching R² in Regression. *Business Education Innovation Journal*, 9(1).
- [363] Nathans, L. L., Oswald, F. L., & Nimon, K. (2012). Interpreting multiple linear regression: a guidebook of variable importance. *Practical assessment, research & evaluation*, 17(9), n9.
- [364] Gagné, P., & Dayton, C. M. (2002). Best regression model using information criteria. *Journal of Modern Applied Statistical Methods*, 1(2), 57.
- [365] Wang, G. C., & Jain, C. L. (2003). *Regression analysis: modeling & forecasting*. Institute of Business Forec.
- [366] Grace-Martin, K. (2012). Can a regression model with a small R-squared be useful. *The Analysis Factor*.
- [367] Menard, S. (2002). *Applied logistic regression analysis* (No. 106). Sage.
- [368] Helm, R., & Mark, A. (2012). Analysis and evaluation of moderator effects in regression models: state of art, alternatives and empirical example. *Review of Managerial Science*, 6, 307-332.
- [369] Bancel, F., Lathuille, Q., & Lhuissier, A. (2013). Why is “your” WACC necessarily wrong?. *ESCP Europe*, 19.
- [370] Bekkar, A., Hssina, B., Douzi, S., & Douzi, K. (2022, March). Air Quality Forecasting using decision trees algorithms. In *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)* (pp. 1-4). IEEE.
- [371] Guo, R., Fu, D., & Sollazzo, G. (2022). An ensemble learning model for asphalt pavement performance prediction based on gradient boosting decision tree. *International Journal of Pavement Engineering*, 23(10), 3633-3646.
- [372] El Guabassi, I., Bousalem, Z., Marah, R., & Qazdar, A. (2021). A recommender system for predicting students' admission to a graduate program using machine learning algorithms.
- [373] Zamouche, M., Chermat, M., Kermiche, Z., Tahraoui, H., Kebir, M., Bollinger, J. C., ... & Mouni, L. (2023). Predictive Model Based on K-Nearest Neighbor Coupled with the Gray Wolf Optimizer Algorithm (KNN_GWO) for Estimating the Amount of Phenol Adsorption on Powdered Activated Carbon. *Water*, 15(3), 493.
- [374] Fehrmann, L., & Kleinn, C. (2005, October). A k-Nearest Neighbor Approach for Estimation of Single-Tree Biomass. In *Proceedings of the Seventh Annual Forest Inventory and Analysis Symposium*; US Department of Agriculture, Forest Service: Washington, DC, USA.
- [375] Moon, T., Hong, S., Choi, H. Y., Jung, D. H., Chang, S. H., & Son, J. E. (2019). Interpolation of greenhouse environment data using multilayer perceptron. *Computers and Electronics in Agriculture*, 166, 105023.
- [376] Darbor, M., Faramarzi, L., & Sharifzadeh, M. (2019). Performance assessment of rotary drilling using non-linear multiple regression analysis and multilayer perceptron neural network. *Bulletin of Engineering Geology and the Environment*, 78, 1501-1513.
- [377] Šegota, S. B., Anđelić, N., Mrzljak, V., Lorencin, I., Kuric, I., & Car, Z. (2021). Utilization of multilayer perceptron for determining the inverse kinematics of an industrial robotic manipulator. *International Journal of Advanced Robotic Systems*, 18(4), 1729881420925283.

Bibliografia rozdziału szóstego

- [1] Mazurek J. A New Approach to a Derivation of a Priority Vector from an Interval Comparison Matrix in a Group AHP Framework. In: *International Conference on Intelligent Decision Technologies*. Springer, Cham, 2016: 193-201.
- [2] Liang F., Brunelli M., and Rezaei J. Consistency issues in the best worst method: Measurements and thresholds. *Omega* 96 (2020): 102175.
- [3] Saaty T. L. A scaling method for priorities in hierarchical structures. *Journal of mathematical psychology* 15.3 (1977): 234-281.
- [4] Saaty T. L. *The analytic hierarchy process* McGraw-Hill. New York, 1980, 324.
- [5] Caflisch R. E. Monte carlo and quasi-monte carlo methods. *Acta numerica*, 1998, 7: 1-49.
- [6] Zeschui X, Cuiping Wei. A consistency improving method in the analytic hierarchy process. *European journal of operational research*, 1999, 116.2: 443-449.
- [7] Cao D., Leung L. C., Law J. S. Modifying inconsistent comparison matrix in analytic hierarchy process: A heuristic approach. *Decision Support Systems*, 2008, 44.4: 944-953.

- [8] Ergu D., Kou G., Peng Y., & Shi Y. A simple method to improve the consistency ratio of the pairwise comparison matrix in ANP. *European Journal of Operational Research*, 213(1), 2021: 246-259.
- [9] Benítez J., Delgado-Galván X., Gutiérrez J. A., Izquierdo J. Balancing consistency and expert judgment in AHP. *Mathematical and Computer Modelling*, 2011, 54.7-8: 1785-1790.
- [10] Benítez J., Delgado-Galván X., Izquierdo J., Pérez-García R. Improving consistency in AHP decision-making processes. *Applied Mathematics and Computation*, 2012, 219.5: 2432-2441.
- [11] Kułakowski K., Juszczyk R., Ernst S. A concurrent inconsistency reduction algorithm for the pairwise comparisons method. In: *International Conference on Artificial Intelligence and Soft Computing*. Springer, Cham, 2015: 214-222.
- [12] Szybowski J. The improvement of data in pairwise comparison matrices. *Procedia Computer Science*, 2018, 126: 1006-1013.
- [13] Abel E., Mikhailov L., Keane J. Inconsistency reduction in decision making via multi-objective optimisation. *European Journal of Operational Research*, 2018, 267.1: 212-226.
- [14] Mazurek J., Perzina R., Strzałka D., Kowal B. A new step-by-step (SBS) algorithm for inconsistency reduction in pairwise comparisons. *IEEE Access*, 2020, 8: 135821-135828.
- [15] Mazurek J., Perzina R., Strzałka D., Kowal B., Kuraś P. A numerical comparison of iterative algorithms for inconsistency reduction in pairwise comparisons. *IEEE Access*, 2021, 9: 62553-62561.
- [16] https://reduce.prz.edu.pl/pc_matrices_generator/
- [17] Brans J., Vincke P., Mareschal B. How to select and how to rank projects: The PROMETHEE method." *European journal of operational research* 24.2 (1986): 228-238.
- [18] Brans, J. P., & De Smet, Y. (2016). PROMETHEE methods. In *Multiple criteria decision analysis* (pp. 187-219). Springer, New York, NY.
- [19] Smarzewski, R., Kozera R. Constructive Consistent Approximations in Pairwise Comparisons. *Advances in Science and Technology Research Journal* 16.4 (2022): 243-255.
- [20] Koczkodaj WW, Orłowski M. An orthogonal basis for computing a consistent approximation to a pairwise comparisons matrix. *Computers and Mathematics with Applications*. 1997;34:41-47.
- [21] Koczkodaj WW, Szarek SJ. On distance-based consistency reduction algorithms for pairwise comparisons. *Logic Journal of the IGPL*. 2010;18:859-869.
- [22] Koczkodaj WW, Szwarc R. On axiomatization of inconsistency indicators for pairwise comparisons. *Fundamenta Informaticae*. 2014;132:485-500.
- [23] Koczkodaj WW, Smarzewski R, Szybowski J. On orthogonal projections on the space of consistent pairwise comparisons matrices. *Fundamenta Informaticae*. 2020;172:379-397.
- [24] Cavallo B, Brunelli M. A general unified framework for interval pairwise comparisons matrices. *International Journal of Approximate Reasoning*. 178-198;93:2018.
- [25] Hansen P., Ombler F. A new method for scoring additive multi-attribute value models using pairwise rankings of alternatives. *Journal of Multi-Criteria Decision Analysis* 15.3-4 (2008): 87-107.
- [26] Opricovic, S., Tzeng G. H. Extended VIKOR method in comparison with outranking methods. *European journal of operational research* 178.2 (2007): 514-529.
- [27] Alkhairi P., Purba L. P., Eryzha A., Windarto A. P., Wanto A. (2019, August). The Analysis of the ELECTREE II Algorithm in Determining the Doubts of the Community Doing Business Online. *Journal of Physics: Conference Series*. Vol. 1255. No. 1. IOP Publishing, 2019.

Spis rysunków

Rys. 2.1. Rozkład $d%$, $n = 4$, $0.10 < CR < 0.30$	Błąd! Nie zdefiniowano zakładki.
Rys. 2.2. Rozkład D , $n = 4$, $0.10 < CR < 0.30$	Błąd! Nie zdefiniowano zakładki.
Rys. 2.3. Rozkład $d%$, $n = 4$, $0.30 \leq CR < 0.80$	Błąd! Nie zdefiniowano zakładki.
Rys. 2.4. Rozkład D , $n = 4$, $0.30 \leq CR < 0.80$	Błąd! Nie zdefiniowano zakładki.
Rys. 2.5. Rozkład $d%$, $n = 8$, $0.10 < CR < 0.30$	Błąd! Nie zdefiniowano zakładki.
Rys. 2.6. Rozkład D , $n = 8$, $0.10 < CR < 0.30$	Błąd! Nie zdefiniowano zakładki.
Rys. 2.7. Rozkład $d%$, $n = 8$, $0.30 \leq CR < 0.80$	Błąd! Nie zdefiniowano zakładki.
Rys. 2.8. Rozkład D , $n = 8$, $0.30 \leq CR < 0.80$	Błąd! Nie zdefiniowano zakładki.
Rys. 2.9. Narzędzie PCM_CR	Błąd! Nie zdefiniowano zakładki.
Rys. 3.2. Plik matrix.csv dostępny w repozytorium [20].....	65
Rys. 3.3. Wizualizacja wyników pomiaru czasowego algorytmów redukcji niespójności ..	83
Rys. 3.4. Wizualizacja wyników zużycia pamięci algorytmów redukcji niespójności	85
Rys. 3.5. Wyjście przykładowej aplikacji.....	88
Rys. 4.1. Struktura aplikacji internetowej REDUCE.prz.edu.pl.....	91
Rys. 4.2. Narzędzie REDUCE.prz.edu.pl – główne menu.....	93
Rys. 4.3. Narzędzie REDUCE.prz.edu.pl – drugie menu	94
Rys. 4.4 Wygląd sekcji loga i nagłówka	100
Rys. 4.5. Wygląd wyboru rozmiaru macierzy, progu redukcji CR oraz wyboru algorytmu redukcji niespójności.....	101
Rys. 4.6. Interfejs aplikacji REDUCE	101
Rys. 4.7. Wygląd przycisków do generowania losowej macierzy oraz resetowania wpisanych wartości.....	101
Rys. 4.8 Wygląd sekcji wprowadzania macierzy	102
Rys. 4.9 Zmiana rozmiaru wprowadzanej macierzy	102
Rys. 4.10. Parametry wprowadzonej macierzy oraz przycisk Reduce!	103
Rys. 4.12. Widok strony mobilnej REDUCE na prawdziwym urządzeniu z systemem Android korzystając z narzędzie BitBar Browser Live Testing	125
Rys. 4.13 Panel ze statystykami z odbytego testu na urządzeniu mobilnym Android	125
Rys. 4.14 Zawartość pliku console.log na urządzeniu iPhone.....	126
Rys. 4.15 Test aplikacji w przeglądarce Chrome z systemem Windows 10	127
Rys. 4.16 Test aplikacji w przeglądarce Firefox z systemem Windows 11.....	127
Rys. 4.17 Test aplikacji w przeglądarce Safari z systemem macOS 12	128
Rys. 4.18 Sumaryczna ocena zastosowanego certyfikatu TLS/SSL.....	133
Rys. 4.19 Sekcja „Sever Key and Certificate #2”	133
Rys. 4.20 Sekcja „Additional Certificates”	134
Rys. 4.21 Sekcja „Certification Paths”	134
Rys. 4.22 Sekcje „Protocols” oraz „Cipher Suites”	135
Rys. 4.23 Rezultaty symulacji protokołu handshake dla certyfikatu.....	136
Rys. 4.24 Szczegóły obsługiwanego protokołów	137
Rys. 4.25 Informacje dodatkowe na temat certyfikatu TLS/SSL	138
Rys. 4.26. Ocena ogólna wydajności witryny REDUCE.prz.edu.pl.....	143
Rys. 4.27. Metryki wydajności witryny REDUCE.prz.edu.pl.....	143
Rys. 4.28. Struktura wpływu poszczególnych elementów na długość ładowania witryny	144
Rys. 4.29. Wykres wodospadu pokazujący czasy ładowania poszczególnych elementów	
Rys. 4.30. Szczegółowe statystyki na temat użytych rodzajów elementów strony	145
Rys. 4.31. Wyniki walidacji HTML, cz. 1	150
Rys. 4.32. Wyniki walidacji HTML, cz. 2	151
Rys. 4.33. Wyniki walidacji HTML, cz. 3	152
Rys. 4.34 Wyniki walidacji CSS.....	153
Rys. 4.35 Wynik audytu WCAG 2.1	154

Rys. 4.36. Wyniki audytu w kategorii „Clickables”	155
Rys. 4.37. Wyniki audytu w kategorii „Titles”	155
Rys. 4.38. Wyniki audytu w kategorii „Menus”	155
Rys. 4.39. Wyniki audytu w kategorii „Orientation”	156
Rys. 4.40. Wyniki audytu w kategorii „Pagination”	156
Rys. 4.41. Wyniki audytu w kategorii „Usability”	156
Rys. 4.42 Wyniki audytu w kategorii „Graphics”	159
Rys. 4.43 Wyniki audytu w kategorii „Forms”	160
Rys. 5.1. Schemat działania algorytmu Szybowskiego (2018)	169
Rys. 5.2. Proces badań	186
Rys. 5.3. Wyniki w plikach CSV	188
Rys. 5.4. Schemat działania skryptu	188
Rys. 5.4. Ostateczna ewaluacja modeli	194
Rys. 5.5. Dystrybucja wartości dla pomiarów błędów MSE , MAE oraz R^2	195
Rys. 5.6. Porównanie przewidywanej wartości CR z rzeczywistą wartością CR	197
Rys. 6.1. Schemat blokowy algorytmu generowania macierzy porównań z danym CR	203
Rys. 6.2. Schemat blokowy ulepszonych algorytmu CR	204
Rys. 6.3. Czas generowania 10 000 losowych macierzy o różnych rozmiarach	205
Rys. 6.4 Porównanie wydajności ulepszonych algorytmu "Xu i Wei" ze zwykłym algorytmem - wartość na wykresie wskazuje wartość procentowej poprawy wydajności (wartość * 100%).	207
Rys. 6.5. Porównanie wydajności ulepszonych algorytmu "Szybowski" ze standardowym algorytmem - wartość na wykresie wskazuje wartość procentowej poprawy wydajności (wartość * 100%).	207
Rys. 6.6 Interfejs aplikacji "PC MATRICES GENERATOR"	208
Rys. 6.7 Dostępne opcje w aplikacji "PC MATRICES GENERATOR"	209
Rys. 6.8 Przykładowe dane wyjściowe aplikacji "PC Matrices Generator" - plik CSV z losowymi macierzami w podanym zakresie CR	209

Spis listingów (dt. Appendixu A)

Listing. 3.1. Plik dockerfile.....	263
Listing. 3.2. Funkcja import_pc_matrix_from_csv().....	263
Listing. 3.3. Funkcja create_pc_matrix().....	263
Listing. 3.4. Funkcja return_random_number().....	264
Listing. 3.5. Funkcja return_max_eigenvalue().....	264
Listing. 3.6. Funkcja calc_vecs().....	264
Listing. 3.7. Funkcja calc_geo_mean().....	265
Listing. 3.8. Funkcja calc_eij().....	265
Listing. 3.9. Funkcja return_ri().....	265
Listing. 3.10. Funkcja dynamic_vectors().....	265
Listing. 3.11. Funkcja gmm_vectors.....	266
Listing. 3.12. Funkcja convert.....	266
Listing. 3.13. Funkcja xu_and_wei_cr().....	266
Listing. 3.14. Funkcja szybowski_cr().....	268
Listing. 3.15. Funkcja cao_cr().....	268
Listing. 3.16. Funkcja koczkodaj_index().....	268
Listing. 3.17. Funkcja golden_wang_index().....	268
Listing. 3.18. Funkcja palaez_lamata_index().....	268
Listing. 3.19. Funkcja geometric_consistency_index().....	269
Listing. 3.20. Funkcja triads_geometric_consistency_index().....	269
Listing. 3.21. Funkcja relative_error_index().....	268
Listing. 3.22. Funkcja harmonic_consistency_index().....	270
Listing. 3.23. Plik test_auxillary_and_indexes.py.....	271
Listing. 3.24. Plik cao_test.py().....	268
Listing. 3.25. Plik szybowski_test.py.....	271
Listing. 3.26. Plik xu_and_wei_test.py.....	272
Listing. 3.27. Wypis z terminala podczas uruchomienia testów.....	272
Listing. 3.28. Program testujący wydajność czasową algorytmów.....	272
Listing. 3.29. Program testujący zużycie pamięci algorytmów.....	273
Listing. 3.30. Program testujący zbieżność algorytmów.....	273
Listing. 3.31. Przykład ilustrujący zastosowanie modułu.....	274
Listing 4.1. Sekcja loga i nagłówek w kodzie HTML.....	275
Listing 4.2. Sekcja instrukcji użytkownika.....	275
Listing 4.3. Sekcja wyboru rozmiaru macierzy.....	275
Listing 4.4. Elementy umożliwiające wprowadzanie macierzy.....	276
Listing 4.5. Parametry wprowadzonej macierzy.....	276
Listing 4.6. Kod odpowiedzialny za przycisk „Reduce!.....	277
Listing 4.7. Globalne resetowanie stylów CSS.....	277
Listing 4.8. Strukturalne elementy.....	277
Listing 4.9. Stylowanie ciała strony.....	278
Listing 4.10. Stylowanie ciała strony.....	279
Listing 4.11. Stylowanie tytułów i opisów.....	279
Listing 4.12. Stylowanie przycisków.....	280
Listing 4.13. Stylowanie list oraz przełączników.....	281
Listing 4.14. Funkcja setMatrixVisibility().....	281
Listing 4.15. Funkcja setMatrixSize().....	281
Listing 4.16. Funkcja getParam().....	282
Listing 4.17. Funkcja setMatrixValue().....	282
Listing 4.18. Funkcja generateRandomMatrix().....	282

Listing 4.19. Funkcja resetMatrix()	241
Listing 4.20. Funkcja setReduceButton()	283
Listing 4.21. Funkcja getCICR()	284
Listing 4.22. Funkcja sendMatrix()	284
Listing 4.23. Funkcja setNewMatrix()	285
Listing 4.24. Funkcja checkMatrix()	285
Listing 4.25. Funkcja checkMatrixItem()	286
Listing 4.26. Funkcja \$(document).ready()	287
Listing 4.27. Import bibliotek back-endowych	288
Listing 4.28. Funkcja computeCI_CR()	288
Listing 4.29. Funkcja reduceButton()	289
Listing 4.30. Plik konfiguracyjny usługi Unicorn	290
Listing 4.31. Plik konfiguracyjny NGINX	290
Listing 4.32. Polecenia uruchamiające NGINX	290
Listing 4.33. Polecenia instalacyjne narzędzia Certbot	291
Listing 4.34. Generowanie certyfikatu za pomocą narzędzia Certbot	291
Listing 4.35. Polecenie crontab okresowo odnawiające certyfikat	291
Listing 4.36. Konfiguracja Uncomplicated Firewall	291
Listing 4.37. Polecenia instalacyjne Snorta	291
Listing 4.38. Reguły bezpieczeństwa dla aplikacji REDUCE	291
Listing 4.39. Uruchomienie Snorta	292
Listing 4.40. Definiowanie interfejsów firewalla Cisco ASA 5505	294
Listing 4.41. Konfiguracja reguł NAT i ACL w firewallu Cisco ASA 5505	292
Listing 4.42. Inspekcja ruchu ICMP w firewallu Cisco ASA 5505	292
Listing 4.43. Bieżąca konfiguracja firewalla Cisco ASA 5505	293
Listing 4.44. Wyniki skanowania IP aplikacji dla najbardziej powszechnych portów	294
Listing 4.45. Wyniki skanowania IP aplikacji dla kolejnych popularnych portów	294
Listing 4.46. Logi aplikacji Snort w przypadku wykrycia ataku XSS	294
Listing 4.47. Logi aplikacji Snort w przypadku wykrycia ataku SQL Injection	295
Listing 4.48. Logi aplikacji Snort w przypadku wykrycia ataku DDoS	295
Listing 5.1. Plik CR_calc.py	271
Listing 5.2. Plik FileLoader.py	272
Listing 5.3. Plik helpers.py	274
Listing 5.4. Plik main.py – sekcja importów	274
Listing 5.5. Plik main.py – klasa Calc	276

Spis tabel

Tabela 1.1. Skala Saaty’ego. Źródło [10].	32
Tabela 1.2. Średnia losowa niespójność (<i>RI</i>). Źródła [62],[63],[64]	34
Tabela 2.1. Wydajność AIR, średnie wart. dla $n = 4$ i początkowe $0,10 \leq CR < 0,30$,...50	
Tabela 2.2. Wydajność AIR, średnie wart. dla $n = 4$ i początkowe $0,30 \leq CR < 0,80$50	
Tabela 2.3. Wydajność AIR, średnie wart. dla $n = 8$ i początkowe $0,10 < CR < 0,30$51	
Tabela 2.4. Wydajność AIR, średnie wart. dla $n = 8$ i początkowe $0,30 \leq CR < 0,80$51	
Tabela 3.1. Złożoność obliczeniowa poszczególnych funkcji modułu Reduce.py.....	76
Tabela 3.2. Średnia pomiarów wydajności czasowej	83
Tabela 3.3. Średnia pomiarów zużycia pamięci	84
Tabela 3.4. Średnia pomiarów badania szybkości zbieżności algorytmów	87
Tabela 4.1. Porównanie cech frameworków Flask oraz Django.....	96
Tabela 4.2. Porównanie cech systemów Ubuntu Server oraz Windows Server	96
Tabela 4.3. Porównanie cech języków Python i R.....	96
Tabela 4.4. Porównanie cech jQuery, Vanilla JavaScript oraz React.....	96
Tabela 4.5. Porównanie cech NGINX i Apache	97
Tabela 4.6. Porównanie Let’s Encrypt oraz płatnych certyfikatów SSL	97
Tabela 5.1. Cechy różnych grup ML w kontekście redukcji niespójności	177
Tabela 5.2. Cechy pożądane do badania potencjału w technikach regresji	180
Tabela 5.3. Fragment pliku CSV przeznaczonego do treningu algorytmu	187
Tabela 5.3. Średnie wartości błędów <i>MSE</i> , <i>MAE</i> i R^2	195
Tabela 5.4. Średnie parametry zachowania preferencji po regresji drzewa decyzyjnego ..	197
Tabela 5.5. Porównanie wyników z iteracyjnymi metodami zmniejszania niespójności..	197
Tabela 6.1. Czas generacji 10 000 macierzy – alg. standardowy	205
Tabela 6.2 Czas generacji 10 000 macierzy – alg. ulepszony (Xu and Wei).....	206
Tabela 6.3 Czas generacji 10 000 macierzy- alg. ulepszony (Szybowski).....	206
Tabela 6.4 Najlepsza metoda dla każdego rozmiaru macierzy i zakresu <i>CR</i>	207

Redukcja niespójności w macierzach porównań parami i nowy algorytm generowania losowych macierzy porównań parami o zadanym zakresie współczynników niespójności - FAST-PCM

Streszczenie

W niniejszej rozprawie doktorskiej skupiono się na problematyce redukcji niespójności w macierzach porównań parami, które stanowią kluczowy komponent w wielokryterialnych metodach podejmowania decyzji. Centralnym punktem badawczym jest wprowadzenie i analiza nowego algorytmu FAST-PCM. Algorytm ten, bazujący na istniejących metodach redukcji niespójności, ma na celu znaczącą optymalizację procesu generowania losowych macierzy porównań z określonym zakresem współczynników niespójności.

Rozpoczynając od pogłębionego przeglądu literatury, autor analizuje koncepcję macierzy porównań parami, ich zastosowania oraz wyzwania związane z niespójnością. Ten teoretyczny fundament stanowi bazę dla empirycznych badań przedstawionych w kolejnych rozdziałach. W rozdziale drugim jest dokonana szczegółowa analiza różnych iteracyjnych algorytmów redukcji niespójności, korzystając z metodologii symulacji Monte Carlo. W trzecim rozdziale przedstawiona jest biblioteka REDUCE.py - narzędzie programistyczne stworzone w języku Python, dedykowane do minimalizacji niespójności w porównaniach parami. Czwarty rozdział koncentruje się na aplikacji webowej REDUCE, która służy do automatycznej redukcji niespójności w macierzach porównań parami. W piątym rozdziale autor bada potencjał wykorzystania technik uczenia maszynowego w kontekście redukcji niespójności, łącząc tradycyjne podejścia z nowoczesnymi technologiami analizy danych. Kluczowym elementem pracy jest szósty rozdział, w którym omówiony jest algorytm FAST-PCM oraz narzędzie online PC MATRICES GENERATOR. Te innowacje nie tylko potwierdzają tezę rozprawy, ale również wskazują na nowe możliwości w dziedzinie wielokryterialnego podejmowania decyzji.

Podsumowując, niniejsza rozprawa doktorska stanowi naukowy wkład w dziedzinę wielokryterialnego podejmowania decyzji. Poprzez syntezę teorii i praktyki oraz wprowadzenie innowacyjnych narzędzi i metod, praca przyczynia się do głębszego zrozumienia i optymalizacji procesu redukcji niespójności w macierzach porównań parami.

Słowa kluczowe:

Algorytmy redukcji niespójności, macierze porównań parami, wielokryterialne podejmowanie decyzji, symulacje Monte Carlo, narzędzia programistyczne, uczenie maszynowe, aplikacje webowe, niespójność, biblioteka REDUCE.py, Przemysł 4.0, technologie decyzyjne, metody numeryczne, fundamentalna skala Saaty'ego, wektor priorytetu, algorytm FAST-PCM, generowanie macierzy, współczynnik niespójności, AHP

Inconsistency Reduction in Pairwise Comparison Matrices and a Novel Algorithm for Generating Random Pairwise Comparison Matrices within a Specified Inconsistency Coefficient Range: FAST-PCM

Abstract

In this doctoral dissertation, the focus is placed on the issue of inconsistency reduction in pairwise comparison matrices, which serve as a pivotal component in multi-criteria decision-making methods. The research's central point is the introduction and analysis of a novel algorithm, FAST-PCM. This algorithm, grounded on existing inconsistency reduction methods, aims to significantly optimize the process of generating random pairwise comparison matrices within a specified inconsistency coefficient range.

Beginning with a comprehensive literature review, the author delves into the concept of pairwise comparison matrices, their applications, and the challenges associated with inconsistency. This theoretical foundation sets the stage for the empirical investigations presented in subsequent chapters. The second chapter offers a detailed analysis of various iterative inconsistency reduction algorithms, utilizing the Monte Carlo simulation methodology. The third chapter introduces the REDUCE.py library - a Python-based software tool designed to minimize inconsistency in pairwise comparisons. The fourth chapter centers on the REDUCE web application, facilitating the automatic reduction of inconsistency in pairwise comparison matrices. In the fifth chapter, the author explores the potential of machine learning techniques in the context of inconsistency reduction, merging traditional approaches with modern data analysis technologies. The work's pivotal element is the sixth chapter, discussing the FAST-PCM algorithm and the online tool PC MATRICES GENERATOR. These innovations not only validate the dissertation's thesis but also indicate new horizons in the field of multi-criteria decision-making.

In conclusion, this doctoral dissertation represents a scholarly contribution to the domain of multi-criteria decision-making. Through the synthesis of theory and practice and the introduction of innovative tools and methods, the work advances a deeper understanding and optimization of the inconsistency reduction process in pairwise comparison matrices.

Keywords:

Inconsistency reduction algorithms, pairwise comparison matrices, multi-criteria decision-making, Monte Carlo simulations, software tools, machine learning, web applications, inconsistency, REDUCE.py library, Industry 4.0, decision-making technologies, numerical methods, Saaty's fundamental scale, priority vector, FAST-PCM algorithm, matrix generation, inconsistency coefficient, AHP.

Appendix A – listingi programów, procedur i poleceń użytych w pracy doktorskiej

```
# Użycie obrazu bazowego z Pythonem 3.9
FROM python:3.9

# Instalacja niezbędnych narzędzi i zależności
RUN pip install poetry black isort mypy pytest

# Kopiowanie kodu źródłowego do kontenera
COPY . /app

# Ustawienie ścieżki pracy
WORKDIR /app

# Instalacja zależności projektu za pomocą Poetry
RUN poetry install

# Instrukcja uruchomienia skryptu lub aplikacji

CMD ["python", "reduce.py"]
```

Listing. 3.1. Plik dockerfile

```
def import_pc_matrix_from_csv(filename):
    reader = csv.reader(open(filename, "r"), delimiter=";")
    x = list(reader)
    imported_matrix = np.array(x).astype("float")

    matrix_size = len(x)
    matrix = eye(matrix_size)
    for n in range(matrix_size):
        for m in range(matrix_size):
            matrix[n, m] = imported_matrix[n][m]

    return matrix
```

Listing. 3.2. Funkcja import_pc_matrix_from_csv()

```
def create_pc_matrix(size):
    n_matrix = eye(size)

    for x in range(size):
        for y in range(size):
            if (n_matrix[x, y]) == 0:
                n_matrix[x, y] = return_random_number()

    for x in range(
        size):
        for y in range(x + 1):
            n_matrix[x, y] = 1 / n_matrix[y, x]

    return n_matrix
```

Listing. 3.3. Funkcja create_pc_matrix()

```

def return_random_number():
    choice_list = [Rational(1, 9), Rational(1, 8), Rational(1, 7),
                  Rational(1, 6), Rational(1, 5), Rational(1, 4),
                  Rational(1, 3), Rational(1, 2),
                  1, 2, 3, 4, 5, 6, 7, 8, 9]

    return random.choice(choice_list)

```

Listing. 3.4. Funkcja return_random_number()

```

def return_max_eigenvalue(n_matrix):
    nump_table = np.array(n_matrix).astype(np.float64)
    evals, evecs = la.eig(nump_table)

    evals = evals.real
    eigen_value_max = max(evals)

    return eigen_value_max

```

Listing. 3.5. Funkcja return_max_eigenvalue()

```

def calc_vecs(n_matrix):
    nump_table = np.array(n_matrix).astype(np.float64)
    evals, evecs = la.eig(nump_table)
    abs_real_evecs = np.absolute(evecs.real)

    sum_of_vec = 0.0
    sum_of_all_vec = 0.0

    for evec in abs_real_evecs:
        sum_of_vec = sum_of_vec + evec

    for sevec in sum_of_vec:
        sum_of_all_vec = sum_of_all_vec + sevec

    i = 0
    list_of_vecs = []

    while i < len(abs_real_evecs):
        list_of_vecs.append(abs_real_evecs[i][0] / sum_of_vec[i])
        i += 1

    return list_of_vecs

```

Listing. 3.6. Funkcja calc_vecs()


```

def calc_geo_mean(n_matrix):
    nump_table = np.array(n_matrix).astype(np.float64)
    list_of_geo_mean = []
    matrix_size = nump_table.shape[0]

    for n in range(matrix_size):
        w = gmean(nump_table[n, :])
        list_of_geo_mean.append(w)

    return list_of_geo_mean

```

Listing. 3.7. Funkcja calc_geo_mean()

```

def calc_eij(n_matrix):
    tmp_mac = np.array(n_matrix).astype(np.float64)
    l_vector = calc_geo_mean(tmp_mac)
    matrix_size = tmp_mac.shape[0]
    list_of_e = eye(matrix_size)

    for i in range(matrix_size):
        for j in range(matrix_size):
            if i < j:
                list_of_e[i, j] = np.abs(np.log(tmp_mac[i, j]
                    * (l_vector[j] / l_vector[i])))
            else:
                list_of_e[i, j] = 0

    return list_of_e

```

Listing. 3.8. Funkcja calc_eij()

```

def return_ri(size):
    ri_values = {
        3: 0.5247, 4: 0.8815, 5: 1.1086, 6: 1.2479, 7: 1.3417,
        8: 1.4056, 9: 1.4499, 10: 1.4854, 11: 1.5140, 12: 1.5365,
        13: 1.5551, 14: 1.5713
    }

    return ri_values.get(size, 0)

```

Listing. 3.9. Funkcja return_ri()

```

def dynamic_vectors(n_matrix):
    nump_table = np.array(n_matrix).astype(np.float64)
    evals, evecs = la.eig(nump_table)
    abs_real_evecs = np.absolute(evecs.real)
    vector_sum = 0, list_of_vecs = []
    for i in range(len(n_matrix)):
        vector_sum = vector_sum + abs_real_evecs[i][0]
    for i in range(len(n_matrix)):
        list_of_vecs.append(abs_real_evecs[i][0] / vector_sum)
    return list_of_vecs

```

Listing. 3.10. Funkcja dynamic_vectors()

```

def gmm_vectors(matrix):
    suma = 0, geo_tmp = [], tmp = 0, vectors = []
    matrix_tmp = convert(matrix).astype('float64')
    size = int((len(matrix)))
    for i in range(size):
        geo = gmean(matrix_tmp[i], axis=0)
        geo_tmp.append(geo)

    for i in range(size):
        suma = suma + geo_tmp[i]

    for i in range(size):
        tmp = geo_tmp[i] / suma
        vectors.append(tmp)
    return vectors

```

Listing. 3.11. Funkcja gmm_vectors

```

def convert(imp_matrix):
    matrix = np.squeeze(np.asarray(imp_matrix))

    return matrix

```

Listing. 3.12. Funkcja convert

```

def xu_and_wei_cr(matrix, lambd, threshold):
    tmp_mac = matrix
    matrix_size = int(math.sqrt(len(tmp_mac)))

    ci = (return_max_eigenvalue(tmp_mac) - matrix_size) /
        (matrix_size - 1)
    ri = return_ri(matrix_size)
    cr = ci / ri
    z = 0

    while cr >= threshold:

        z = z + 1
        l_vector = calc_vecs(tmp_mac)

        for n in range(matrix_size):
            for m in range(matrix_size):
                tmp_mac[n, m] = (pow(tmp_mac[n, m], lambd)) *
                    (pow((l_vector[n] / l_vector[m]), (1 - lambd)))

        ci = (return_max_eigenvalue(tmp_mac) - matrix_size) /
            (matrix_size - 1)
        cr = ci / ri

    return tmp_mac, ci, cr

```

Listing. 3.13. Funkcja xu_and_wei_cr()

```

def cao_cr(matrix, lamdb, threshold):
    tmp_mac = matrix
    matrix_size = int(math.sqrt(len(tmp_mac)))

    ci = (return_max_eigenvalue(tmp_mac) - matrix_size) /
          (matrix_size - 1)
    ri = return_ri(matrix_size)
    cr = ci / ri, z = 0

    while cr >= threshold:
        z = z + 1
        l_vector = calc_vecs(tmp_mac)
        for n in range(matrix_size):
            for m in range(matrix_size):
                tmp_mac[n, m] = (l_vector[n] / l_vector[m]) * (
                    lamdb * tmp_mac[n, m] * (l_vector[m] /
                    l_vector[n]) + (1 - lamdb))
        ci = (return_max_eigenvalue(tmp_mac) - matrix_size) /
              (matrix_size - 1)
        cr = ci / ri

    return tmp_mac, ci, cr

```

Listing. 3.14. Funkcja cao_cr()

```

def szybowski_cr(matrix, threshold):

    tmp_mac = matrix
    matrix_size = int(math.sqrt(len(tmp_mac)))
    ci = (return_max_eigenvalue(tmp_mac) - matrix_size) /
          (matrix_size - 1)
    ri = return_ri(matrix_size)
    cr = ci / ri
    z = 0

    while cr > threshold:

        z = z + 1
        list_of_eij = calc_eij(tmp_mac)
        l_vector = calc_geo_mean(tmp_mac)
        max_value = 0

        p = 0, q = 0
        i = 0, j = 0

        for i in range(matrix_size):
            for j in range(matrix_size):
                if i < j:
                    if max_value < list_of_eij[i, j]:
                        max_value = list_of_eij[i, j]
                        p = i
                        q = j

        tmp_mac[p, q] = l_vector[p] / l_vector[q]
        tmp_mac[q, p] = tmp_mac[p, q] ** -1

        ci = (return_max_eigenvalue(tmp_mac) - matrix_size) /
              (matrix_size - 1)
        cr = ci / ri

    return tmp_mac, ci, cr

```

Listing. 3.15. Funkcja szybowski_cr()

```

def koczkodaj_index(matrix):
    min_list = []
    for i in range(len(matrix)):
        for j in range(len(matrix)):
            for k in range(len(matrix)):
                min_a = abs(1 - matrix[i][j] / (matrix[i][k] *
                    matrix[k][j]))
                min_b = abs(1 - (matrix[i][k] * matrix[k][j]) /
                    matrix[i][j])
                min_list.append(min(min_a, min_b))

    del matrix

    return max(min_list)

```

Listing. 3.16. Funkcja koczkodaj_index

```

def golden_wang_index(matrix):
    sum_gw = []
    gw_index = 0

    for n in range(len(matrix)):
        sum_gw = 0
        for m in range(len(matrix)):
            sum_gw = sum_gw + matrix[m][n]
        sum_gw.append(sum_gw)

    for n in range(len(matrix)):
        for m in range(len(matrix)):
            matrix[m][n] = matrix[m][n] / sum_gw[n]

    vectors = dynamic_vectors(matrix)

    for n in range(len(matrix)):
        for m in range(len(matrix)):
            gw_index = gw_index + abs(matrix[n][m] - vectors[n])

    gw_index = (1 / (len(matrix) * len(matrix))) * gw_index

    del matrix

    return gw_index

```

Listing. 3.17. Funkcja golden_wang_index()

```

def pelaez_lamata_index(matrix):
    pli = 0
    for i in range(len(matrix) - 2):
        for j in range(len(matrix) - 1):
            for k in range(len(matrix)):
                if j > i:
                    if k > j:
                        pli = pli +
                            ((matrix[i][k] / (matrix[i][j]
                                * matrix[j][k])) + ((matrix[i][j] *
                                    matrix[j][k]) / matrix[i][k]) - 2)
    n = len(matrix)
    pli = pli * (6 / (n * (n - 1) * (n - 2)))
    del matrix
    return pli

```

Listing. 3.18. Funkcja palaez_lamata_index()

```

def geometric_consistency_index(matrix):
    gci = 0
    vectors = gmm_vectors(matrix)

    for i in range(len(matrix) - 1):
        for j in range(len(matrix)):
            if j > i:
                gci = gci + (log((matrix[i][j]) *
                                (vectors[j] / vectors[i]))) * (
                                log((matrix[i][j]) * (vectors[j] /
                                vectors[i])))

    n = len(matrix)
    gci = gci * (2 / ((n - 1) * (n - 2)))

    del matrix

    return gci

```

Listing. 3.19. Funkcja geometric_consistency_index()

```

def triads_geometric_consistency_index(matrix):
    tgci = 0
    n = len(matrix)

    for k in range(len(matrix)):
        for j in range(len(matrix)):
            for i in range(len(matrix)):
                if k > j:
                    if j > i:
                        tgci = tgci + (log(matrix[i][j] * matrix[j][k]
                                            * matrix[k][i])) * (log(matrix[i][j] *
                                            matrix[j][k] * matrix[k][i]))

    tgci = 2 * tgci
    tgci = tgci / (n * (n - 1) * (n - 2))

    del matrix

    return tgci

```

Listing. 3.20. Funkcja triads_geometric_consistency_index()

```

def relative_error_index(matrix):
    a,b,rei = 0
    vectors = gmm_vectors(matrix)

    for i in range(len(matrix)):
        for j in range(len(matrix)):
            a = a + (log(matrix[i][j]) - log((vectors[i] /
                                                vectors[j]))) * (log(matrix[i][j]) -
                                                log((vectors[i] / vectors[j])))
            b = b + log(matrix[i][j]) * log(matrix[i][j])
    rei = a / b

    del a,b,matrix
    return rei

```

Listing. 3.21. Funkcja relative_error_index()

```

def harmonic_consistency_index(matrix):
    sj = []
    for j in range(len(matrix)):
        tmp = 0
        for i in range(len(matrix)):
            tmp = matrix[i][j] + tmp
        sj.append(tmp)
    hm = 0
    for j in range(len(matrix)):
        hm = hm + (1 / sj[j])
    hm = (len(matrix)) / hm

    hci = ((hm - len(matrix)) * (len(matrix) + 1)) /
           (len(matrix) * (len(matrix) - 1))

    del matrix
    return hci

```

Listing. 3.22. Funkcja harmonic_consistency_index()

```

import pytest
import numpy as np
from reduce import *

def test_create_pc_matrices():
    matrix = create_pc_matrices(3)
    assert matrix.shape == (3, 3)

def test_return_random_number():
    number = return_random_number()
    assert number >= 1/9 and number <= 9

def test_return_max_eigenvalue():
    matrix = np.array([[1, 2], [3, 4]])
    max_eigenvalue = return_max_eigenvalue(matrix)
    assert max_eigenvalue == pytest.approx(5.3722813232690143)

def test_calc_vecs():
    matrix = np.array([[1, 2], [3, 4]])
    vecs = calc_vecs(matrix)
    assert len(vecs) == 2

def test_calc_geo_mean():
    matrix = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
    geo_mean = calc_geo_mean(matrix)
    assert len(geo_mean) == 3

...

def test_golden_wang_index():
    matrix = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
    gwi = golden_wang_index(matrix)
    assert gwi >= 0

def test_pelaez_lamata_index():
    matrix = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
    pli = pelaez_lamata_index(matrix)
    assert pli >= 0

def test_geometric_consistency_index():
    matrix = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
    gci = geometric_consistency_index(matrix)
    assert gci >= 0

```

```

def test_triads_geometric_consistency_index():
    matrix = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
    tgci = triads_geometric_consistency_index(matrix)
    assert tgci >= 0

def test_relative_error_index():
    matrix = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
    rei = relative_error_index(matrix)
    assert rei >= 0

def test_harmonic_consistency_index():
    matrix = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
    hci = harmonic_consistency_index(matrix)
    assert hci >= 0

```

Listing. 3.23. Plik test_auxillary_and_indexes.py

```

def test_cao_cr():

    matrix = np.random.rand(4, 4)

    matrix_size = matrix.shape[0]
    ci_initial = (return_max_eigenvalue(matrix) - matrix_size)
                / (matrix_size - 1)
    ri_initial = return_ri(matrix_size)
    cr_initial = ci_initial / ri_initial

    lambd = 0.5
    threshold = 0.1
    new_matrix, ci_final, cr_final = cao_cr(matrix, lambd, threshold)

    assert cr_final < threshold
    assert cr_final < cr_initial
    assert new_matrix.shape == matrix.shape

```

Listing. 3.24. Plik cao_test.py

```

def test_szybowski_cr():

    matrix = np.random.rand(4, 4)

    matrix_size = matrix.shape[0]
    ci_initial = (return_max_eigenvalue(matrix) - matrix_size)
                / (matrix_size - 1)
    ri_initial = return_ri(matrix_size)
    cr_initial = ci_initial / ri_initial

    threshold = 0.1
    new_matrix, ci_final, cr_final = szybowski_cr(matrix, threshold)

    assert cr_final < threshold
    assert cr_final < cr_initial
    assert new_matrix.shape == matrix.shape

```

Listing. 3.25. Plik szybowski_test.py

```

def test_szybowski_cr():

    matrix = np.random.rand(4, 4)

    matrix_size = matrix.shape[0]
    ci_initial = (return_max_eigenvalue(matrix) - matrix_size) /
(matrix_size - 1)
    ri_initial = return_ri(matrix_size)
    cr_initial = ci_initial / ri_initial

    threshold = 0.1
    new_matrix, ci_final, cr_final = szybowski_cr(matrix, threshold)

    assert cr_final < threshold
    assert cr_final < cr_initial
    assert new_matrix.shape == matrix.shape

```

Listing. 3.26. Plik xu_and_wei_test.py

```

$ pytest test_cao.py test_szybowski.py test_xu_and_wei.py
test_auxillary_and_indexes.py

===== test session starts =====

platform linux -- Python 3.8.5, pytest-6.2.1, pluggy-0.13.1
collected 16 items

test_cao.py . [ 6%]
test_szybowski.py . [ 12%]
test_xu_and_wei.py . [ 18%]
test_auxillary_and_indexes.py ..... [100%]

===== 16 passed in 0.12s =====

```

Listing. 3.27. Wypis z terminala podczas uruchomienia testów

```

import timeit

def test_time_performance(algorithm, size, count=10000):
    start_time = timeit.default_timer()
    for _ in range(count):
        matrix = generate_random_matrix(size)
        algorithm(matrix, 0.5, 0.1)
    elapsed_time = timeit.default_timer() - start_time
    return elapsed_time

sizes = range(3, 11)
for size in sizes:
    print(f"Testing time performance for size {size}x{size}:")
    print(f"xu_and_wei: {test_time_performance(xu_and_wei_cr, size)} s")
    print(f"cao_cr: {test_time_performance(cao_cr, size)} s")
    print(f"szybowski: {test_time_performance(szybowski_cr, size)} s")

```

Listing. 3.28. Program testujący wydajność czasową algorytmów


```

from memory_profiler import memory_usage

def test_memory_usage(algorithm, size, count=10000):
    mem_usages = [memory_usage((algorithm,
                                generate_random_matrix(size), 0.5, 0.1)),
                  (generate_random_matrix(size), 0.5, 0.1)),
                  max_usage=True) for _ in range(count)]
    avg_mem_usage = sum(mem_usages) / len(mem_usages)

    return avg_mem_usage

sizes = range(3, 11)
for size in sizes:

    print(f"Testing memory usage for size {size}x{size}:")
    print(f"xu_and_wei: {test_memory_usage(xu_and_wei_cr, size)} MiB")
    print(f"cao_cr: {test_memory_usage(cao_cr, size)} MiB")
    print(f"szybowski_cr: {test_memory_usage(szybowski_cr, size)} MiB")

```

Listing. 3.29. Program testujący zużycie pamięci algorytmów

```

def xu_and_wei_cr(matrix, lambd, threshold):
    # ... reszta kodu ...
    return tmp_mac, ci, cr, z # Dodaj z jako liczbę iteracji
def cao_cr(matrix, lambd, threshold):
    # ... reszta kodu ...
    return tmp_mac, ci, cr, z # Dodaj z jako liczbę iteracji
def szybowski_cr(matrix, threshold):
    # ... reszta kodu ...
    return tmp_mac, ci, cr, z # Dodaj z jako liczbę iteracji

def test_convergence_speed_and_iterations(matrix, lambd, threshold):

    # Zmierz czas zbieżności i iteracje dla algorytmu xu_and_wei_cr
    start_time = time.time()
    _, _, _, xu_and_wei_iterations = xu_and_wei_cr(matrix, lambd, threshold)
    xu_and_wei_time = time.time() - start_time

    # Zmierz czas zbieżności i iteracje dla algorytmu cao_cr
    start_time = time.time()
    _, _, _, cao_iterations = cao_cr(matrix, lambd, threshold)
    cao_time = time.time() - start_time

    # Zmierz czas zbieżności i iteracje dla algorytmu szybowski_cr
    start_time = time.time()
    _, _, _, szybowski_iterations = szybowski_cr(matrix, threshold)
    szybowski_time = time.time() - start_time

    return (xu_and_wei_time, xu_and_wei_iterations),
           (cao_time, cao_iterations),
           (szybowski_time, szybowski_iterations)

# Wywołaj funkcję testującą
xu_and_wei_result, cao_result, szybowski_result =
test_convergence_speed_and_iterations(matrix, lambd, threshold)

print("Czas zbieżności i liczba iteracji:")
print("Algorytm xu_and_wei_cr: czas =", xu_and_wei_result[0] * 1000,
      "ms, iteracje =", xu_and_wei_result[1])
print("Algorytm cao_cr: czas =", cao_result[0] * 1000,
      "ms, iteracje =", cao_result[1])
print("Algorytm szybowski_cr: czas =", szybowski_result[0] * 1000,
      "ms, iteracje =", szybowski_result[1])

```

Listing. 3.30. Program testujący zbieżność algorytmów

```

import reduce
from auxiliary_functions import multiple_append, save_matrices_to_csv

# Pusta lista macierzy, która będzie zawierała zredukowane macierze
list_of_matrices = []
# Pusta lista współczynnika spójności zredukowanych macierzy
list_of_ci = []
# Pusta lista wskaźnika spójności zredukowanych macierzy
list_of_cr = []
# Puste listy dla wskaźników z biblioteki REDUCE
ki, gwi, pli, gci, tgci, rei, hci = []
# Zmienna określająca rozmiar generowanej macierzy
size_of_matrix = 6

# Generowanie losowej macierzy o podanym rozmiarze
matrix = create_pc_matrix(size_of_matrix)

matrix_xw, cr_xw, ci_xw = xu_and_wei_cr(matrix, 0.9, 0.1)
# Funkcja redukcji CR generuje trzy wyjścia,
# podczas gdy wejście potrzebuje tablicy,
# parametru lambda, któremu nadajemy domyślną wartość,
# czyli 0,9, oraz progę parametru CR,
# czyli 0,1, jak opisano w artykule

# Podobnie jak powyżej
matrix_c, cr_c, ci_c = cao_cr(matrix, 0.9, 0.1)

# Algorytm Szybowskiego nie ma parametru lambda
matrix_s, cr_s, ci_s = szybowski_cr(matrix, 0.9, 0.1)

# funkcja pomocnicza, która dodaje powyższe dane do list
multiple_append(list_of_matrices, list_of_ci, list_of_cr,
                matrix_xw, matrix_c, matrix_s, cr_xw, cr_c,
                cr_s, ci_xw, ci_c, ci_s)

# dodawanie wszystkich wskaźników do listy wskaźników
for mx in list_of_matrices:
    # Indeks Koczkodaja
    ki.append(koczkodaj_index(mx))
    # Indeks Golden Wang
    gwi.append(golden_wang_index(mx))
    # Indeks Palaeza Lamaty
    pli.append(palaez_lamata_index(mx))
    # Indeks Geometrii Spójności
    gci.append(geometric_consistency_index(mx))
    # Indeks Geometrii Spójności Triad
    tgci.append(triads_geometric_consistency_index(mx))
    # Indeks Błędu Względnego
    rei.append(relative_error_index(mx))
    # Indeks Spójności Harmonicznej
    hci.append(harmonic_consistency_index(mx))

# funkcja pomocnicza, która zapisuje do pliku csv
save_matrices_to_csv(list_of_matrices, list_of_cr,
                    list_of_ci, ki, gwi, pli, gci, tgci, rei, hci)

```

Listing. 3.31. Przykład ilustrujący zastosowanie modułu

```

<div class="header_logos">
<p>
  <a href="https://w.prz.edu.pl/en/" target="_blank">
  </a>
</p>

  <h1 class="title">REDUCE</h1>
  <p class="description">
    A simple ...</p>
  <p color="red">
    IMPORTANT NOTE: If ...
    The paper is available by clicking on this
    <a href="https://www.researchgate.net/
    ..._
    multiplicative_pairwise_comparisons"
    target="_blank">LINK</a>.</p>
</div>

```

Listing 4.1. Sekcja loga i nagłówek w kodzie HTML

```

<div class="how_to_use_it">
  <p class="toggle js-toggle">How to use it?</p>
  <ol class="list js-list" style="display:none;">
    <li>Select ...tio).</li>
    <li>Select the threshold of reducing CR:
      <ul style="padding-top: 10px">- < 0.1 ...,</ul>
      <ul style="padding-top: 10px">- 0 - ...</ul>
    </li>
    <li>Select the algorithm for reducing CR:
      <ul style="padding-top: 10px">- Xu and...,</ul>
      <ul style="padding-top: 10px">- Szybowski.</ul>
    </li>
    <li>Type in values to a matrix:</li>
      <ul style="padding-top: 10px">- Values have ...</ul>
      <ul style="padding-top: 10px">- The diagona...,</ul>
      <ul style="padding-top: 10px">- Values at ...</ul>
      <ul style="padding-top: 20px; padding-bottom: 20px">
        You could ... </ul>
    </li>
    <li>After filling ...</li>
    <li>Click the ...eet.</li>
  </ol>
</div>

```

Listing 4.2. Sekcja instrukcji użytkownika

```

<div class="size_of_a_matrix">
  <label for="matrixSize">Size of a matrix:</label>
  <select name="matrixSize" id="matrixSize"
  class="js-clear-reduced-matrix">
    <option value="3" selected>3</option>
    ... </option></select></div>
<div class="reduce_threshold">
  <label for="reduceThreshold">Threshold of reducing CR:</label>
  <select name="reduceThreshold" id="reduceThreshold"
  class="js-clear-reduced-matrix">
    <option value="1" selected>&LT; 0.1</option>
    ...</select></div>
<div class="algorithm_selection"> .....

```

Listing 4.3. Sekcja wyboru rozmiaru macierzy, progę redukcji CR oraz wyboru algorytmu redukcji niespójności

```

<div class="matrix section center">
  <ul class="matrix_row">
    <li class="matrix_row_item">
      <input name="matrix[1][1]" value="1"
        class="js-matrix-item" readonly> </li>
    <li class="matrix_row_item">
      <input name="matrix[1][2]" class="js-matrix-item"></li>
    <li class="matrix_row_item">
      <input name="matrix[1][3]" class="js-matrix-item"></li>
    <li class="matrix_row_item">
      <input name="matrix[1][4]" class="js-matrix-item"></li>
    <li class="matrix_row_item">
      <input name="matrix[1][5]" class="js-matrix-item"></li>
    <li class="matrix_row_item">
      <input name="matrix[1][6]" class="js-matrix-item"></li>
    <li class="matrix_row_item">
      <input name="matrix[1][7]" class="js-matrix-item"></li>
    <li class="matrix_row_item">
      <input name="matrix[1][8]" class="js-matrix-item"></li>
    <li class="matrix_row_item">
      <input name="matrix[1][9]" class="js-matrix-item"></li>
    <li class="matrix_row_item">
      <input name="matrix[1][10]" class="js-matrix-item"></li>
  </ul>
  <ul class="matrix_row">
    <li class="matrix_row_item">
      <input name="matrix[2][1]" class="js-matrix-item"></li>
    <li class="matrix_row_item">
      <input name="matrix[2][2]" value="1"
        class="js-matrix-item" readonly></li>
    <li class="matrix_row_item">
      <input name="matrix[2][3]" class="js-matrix-item"></li>
    ...
    <li class="matrix_row_item">
      <input name="matrix[10][6]" class="js-matrix-item"></li>
    <li class="matrix_row_item">
      <input name="matrix[10][7]" class="js-matrix-item"></li>
    <li class="matrix_row_item">
      <input name="matrix[10][8]" class="js-matrix-item"></li>
    <li class="matrix_row_item">
      <input name="matrix[10][9]" class="js-matrix-item"></li>
    <li class="matrix_row_item">
      <input name="matrix[10][10]" value="1"
        class="js-matrix-item" readonly></li>
  </ul></div>

```

Listing 4.4 Elementy umożliwiające wprowadzanie macierzy

```

<p style="margin-top:16px;">
CI: <span class="js-CI"></span><br>
CR: <span class="js-CR"></span><br>
W: <span class="js-EV"></span><br>
KI: <span class="js-KI"></span><br>
GC: <span class="js-GC"></span><br>
GW: <span class="js-GW"></span><br>
PLI: <span class="js-PLI"></span><br>
TGCI: <span class="js-TGCI"></span><br>
RE: <span class="js-RE"></span><br>
HC: <span class="js-HC"></span></p>

```

Listing 4.5 Elementy umożliwiające wprowadzanie macierzy

```

<div class="section center">
  <button type="submit"
    class="reduceButton js-reduce-button"
    disabled="disabled">Reduce!</button>
</div>
<div class="matrix section center js-reduced-matrix">
</div>

```

Listing 4.6. Kod odpowiedzialny za przycisk „Reduce!” oraz przedstawienie zredukowanej macierzy

```

html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td,
article, aside, canvas, details, embed,
figure, figcaption, footer, header, hgroup,
menu, nav, output, ruby, section, summary,
time, mark, audio, video {
  margin: 0;
  padding: 0;
  border: 0;
  font-size: 100%;
  font: inherit;
  vertical-align: baseline; }

```

Listing 4.7. Globalne resetowanie stylów CSS

```

article, aside, details, figcaption, figure,
footer, header, hgroup, menu, nav, section
{display: block; }

```

Listing 4.8. Strukturalne elementy

```

body { line-height: 1;}
ol, ul { list-style: none;}
html, body { height: 100%; margin: 0;
              font-family: 'Lato', sans-serif; }
label{
  width: 250px;
  display: inline-block;
  padding-top: 10px;
}

.container {
  min-height: 100%;
  margin-bottom: -62px;
  padding: 0px 40px;
  max-width: 800px;
  margin-left: auto;
  margin-right: auto;
}

.footer{
  height: 41px;
  font-size: 12px;
  line-height: 14px;
  color: #b0b1b2;
  padding: 10px 20px;
  border-top: 1px solid #b0b1b2;
}

.push{
  height: 121px;
  font-size: 12px;
  line-height: 14px;
}

```

Listing 4.9. Stylowanie ciała strony

```

.matrix td{
    display: inline-block;
    height: 30px;
    width: 60px;
    line-height: 30px;
    font-size:15px;
}
.matrix_row{
    font-size:0;
    padding:0;
}
.matrix_row_item{
    display:none;
    font-size: 15px;
}
.matrix_row_item.show{
    display:inline-block;
}

.matrix_row_item input{
    width: 56px;
    height: 28px;
    line-height: 28px;
    text-align: center;
    border: 1px solid #8f8f8f;
    font-size: 15px;

    -webkit-transition: border-color 0.4s ease;
    -moz-transition: border-color 0.4s ease;
    -o-transition : border-color 0.4s ease;
    transition: border-color 0.4s ease;
}

.matrix_row_item input:focus{
    outline: 0;
    border: 1px solid #3f51b5;
}

```

Listing 4.10. Stylowanie ciała strony

```

.title{
    font-size: 60px;
    color: #3F51B5;
    font-weight: bold;
    border-top: 1px solid #b0b1b2;
    padding-top: 40px;
}

.description{
    padding: 10px 0px 20px;
    font-weight: bold;
    font-size: 20px;
}

```

Listing 4.11. Stylowanie tytułów i opisów

```

button{
  border: 1px solid #3F51B5;
  background-color: #3F51B5;
  border-radius: 4px;
  padding: 8px 16px;
  color: white;
  font-size: 20px;
  cursor:pointer;

  -webkit-transition: background-color 0.4s ease, color 0.4s ease;
  -moz-transition: background-color 0.4s ease, color 0.4s ease;
  -o-transition : background-color 0.4s ease, color 0.4s ease;
  transition: background-color 0.4s ease, color 0.4s ease;
}

button:hover{
  background-color: #1f339d;
  outline: none;
}

button:focus{
  outline: none;
}

.generateButton{
  padding: 4px 8px;
  font-size: 14px;
  margin-top: 16px;
}

.resetButton{
  padding: 4px 8px;
  font-size: 14px;
  margin-top: 16px;
}

button:disabled,
button[disabled],
button[disabled]:hover,
button[disabled]:focus{
  background-color: #e3e5e8;
  color: #b0b1b2;
  border: 1px solid #b0b1b2;
  outline: none;
}

```

Listing 4.12. Stylowanie przycisków


```

.list{
    list-style: decimal;
    padding: 15px 15px 15px 35px;
    background-color: #f0f8ff;
}

.list li{
    padding: 5px 10px;
}

.toggle{
    margin-bottom: 20px;
    cursor: pointer;
    padding-bottom: 5px;
    border-bottom: 2px solid #3f51b5;
    display: inline-block;
    color: #000;

    -webkit-transition: color 0.4s ease;
    -moz-transition: color 0.4s ease;
    -o-transition : color 0.4s ease;
    transition: color 0.4s ease;
}

.toggle:hover{
    color: #3f51b5;
}

```

Listing 4.13. Stylowanie list oraz przełączników

```

function setMatrixVisibility(item, size){
    var name = item.attr('name'),
        param = getParam(name);

    item.parent().removeClass('show');
    if(parseInt(param.row) <= size && parseInt(param.column)
        <= size){
        item.parent().addClass('show'); }; }

```

Listing 4.14. Funkcja setMatrixVisibility()

```

function setMatrixSize() {
    var size = $('#matrixSize').val(),
        items = $('.js-matrix-item');

    items.each(function() {
        var item = $(this);
        setMatrixVisibility(item, parseInt(size));
    });
}

```

Listing 4.15. Funkcja setMatrixSize()

```

function getParam(name) {
    var temp = name.indexOf('['),
        row = name.substring(name.indexOf('[')+1, temp),
        column = name.substring(temp+2, name.length-1);
    return {
        row: row,
        column: column }
}

```

Listing 4.16. Funkcija getParam()

```

function setMatrixValue(item) {
    var name = item.attr('name'),
        value = item.val(),
        param = getParam(name),
        itemInv = $('[name="matrix[' + param.column
            + '][' + param.row + ']"'),
        algorithm = ($.js-algorithm).val(),
        allowDec = (algorithm !== '3'),
        result;
    if(value === ''){
        result = '';
    }
    else if(value.length === 1){
        result = (value === '1')? '1' : '1/' + value;
    }
    else if(value.indexOf('/') > -1){
        result = value.substring(value.indexOf('/')+1, value.length);
    }
    else if(value.indexOf('.') > -1){
        var inv = 1 / parseFloat(value),
            invStr = inv.toString();

        result = (invStr.length > 6)? inv.toFixed(4) : inv;
    }
    itemInv.val(result);
}

```

Listing 4.17. Funkcija setMatrixValue()

```

function generateRandomMatrix() {
    var item = ($.js-matrix-item),
        values = Array('1', '2', '3', '4', '5', '6', '7',
            '8', '9', '1/2', '1/3', '1/4', '1/5',
            '1/6', '1/7', '1/8', '1/9');
    ($.js-matrix-item:not([readonly])).val('');
    item.each(function() {
        var elem = $(this),
            value = elem.val();
        if(value === '' && !elem.prop('readonly')){
            var rand = values[Math.floor(Math.random() * values.length)];
            elem.val(rand);
            setMatrixValue(elem);
        }
    });
    setReduceButton();
}

```

Listing 4.18. Funkcija generateRandomMatrix()

```

function resetMatrix() {
  $(' .js-matrix-item:not([readonly])').val('');

  const classesToReset = [ '.js-CI', '.js-CR', '.js-EV',
                           '.js-KI', '.js-GC', '.js-GW',
                           '.js-PLI', '.js-TGCI', '.js-RE', '.js-HC'];
  for (let className of classesToReset) {
    $(className).text('');
  }
}

```

Listing 4.19. Funkcja resetMatrix()

```

function setReduceButton() {
  var disabled = false;

  $(' .js-matrix-item').each(function() {
    var item = $(this);

    if(item.parent().hasClass('show')){
      if(item.val() === ''){
        disabled = true;
        return false;
      }
    }
  });

  $(' .js-reduce-button').prop('disabled', disabled);

  if(!disabled){
    getCICR();
  }
  else {

  const classesToReset = [ '.js-CI', '.js-CR', '.js-EV',
                           '.js-KI', '.js-GC', '.js-GW',
                           '.js-PLI', '.js-TGCI', '.js-RE',
                           '.js-HC'];
  for (let className of classesToReset) {
    $(className).text('');
  }

  $(' .js-reduced-matrix').html('');
}
}

```

Listing 4.20. Funkcja setReduceButton()

```

function getCICR() {
    var form = $('#matrixForm'),
        serializedData = form.serialize();

    $.ajax({
        url: '/computeCI_CR',
        data: serializedData,
        type: 'POST',
        success: function(response) {
            $.each(response, function( index, value ) {
                if (index != 'EV'){
                    var item = $('.js-'+index);

                    if(item.length > 0){
                        var temp = (value !== undefined &&
                            value !== '')?
                            value.toFixed(4) : '';
                        item.html(temp);}

                    else {
                        $('.js-EV').text(value);}
                });},
        error: function(error) {
            console.log(error);
        });
    });
}

```

Listing 4.21. Funkcja getCICR()

```

function sendMatrix() {
    var form = $('#matrixForm'),
        serializedData = form.serialize();
    $.ajax({
        url: '/reduceButton',
        data: serializedData,
        type: 'POST',
        success: function(response) {
            setNewMatrix(response);
        },
        error: function(error) {
            console.log(error);    });});
}

```

Listing 4.22. Funkcja sendMatrix()

```

function setNewMatrix(data) {
    var wrapper = $('#js-reduced-matrix');
    var size    = $('#matrixSize').val(),
        temp    = size * size,
        matrix  = '';
    $.each(data, function( index, value ) {

        if(index <= temp){
            var open  = (index%size === 1)? '<tr>' : '';
            var close = (index%size === 0)? '</tr>' : '';
            var v      = (value === parseInt(value))?
                value : value.toFixed(4);
            matrix = matrix + open + '<td>' +
                v + '</td>' + close;});

    var ci  = (data[1000] !== undefined && data[1000] !== '')?
        data[1000].toFixed(4) : '';
    var cr  = (data[1001] !== undefined && data[1001] !== '')?
        data[1001].toFixed(4) : '';
    var ev  = data[1002]; var ki  = data[1003]
    var gc  = data[1004]; var gw  = data[1005]
    var pl  = data[1006]; var tgci = data[1007]
    var re  = data[1008]; var hc  = data[1009]

    wrapper.html
    ('<table>' + matrix + '</table>
    <p style="margin-top:16px;">CI: '
    +ci+'<br>CR: '+cr+'<br>W: '+ev+'<br>KI: '
    +ki+'<br>GC: '+gc+'<br>GW: '+gw+'<br>PLI: '
    +pl+'<br>TGCI: '+tgci+'<br>RE: '+re+'<br>HC: '
    +hc+'</p>');
}

```

Listing 4.23. Funkcja setNewMatrix()

```

function checkMatrix() {
    var algorithm = $('#js-algorithm').val(),
        allowDec  = (algorithm !== '3');
    if(!allowDec){
        $('#js-matrix-item').each(function(){
            var item = $(this),
                value = item.val();
            if(value.indexOf('.') > -1){
                item.val('');
            }
        });
        setReduceButton();
    }
}

```

Listing 4.24. Funkcja checkMatrix()

```

function checkMatrixItem(item) {
    var value = item.val(),
        algorithm = $('<div>.js-algorithm').val(),
        allowDec = (algorithm !== '3');

    if(allowDec){
        var f = parseFloat(value),
            i = parseInt(f),
            inv = 1 / f;

        if(inv >= 9){
            item.val('');
        }

        else if(value === 0){
            item.val('');
        }

        else if(value.length === 2 && value.indexOf('.') === 1){
            item.val('');
        }

        else if(f === i && value.indexOf('.') > -1){
            item.val(i);
        }

        else if(value.length > 6){
            item.val('');
        }

    }

    else if(value.length > 3){
        item.val('');
    }

    if(value === '1/1'){
        item.val('1');
    }

    else if(value.length === 2 && value.indexOf('/') === 1){
        item.val('');
    }

    else if(value.indexOf('.') === 0 || value.indexOf('/') === 0){
        item.val('');
    }

    else if(value.indexOf('.') > 1 || value.indexOf('/') > 1){
        item.val('');
    }

    setMatrixValue(item);
    setReduceButton();
}

```

Listing 4.25. Funkcja checkMatrixItem()

```

$(document).ready(function() {

    setMatrixSize(); setReduceButton();

    $('#matrixSize').on('change', function() {
        setMatrixSize(); setReduceButton();
        $(window).resize();});

    $('.js-matrix-item').on('change', function() {
        var item = $(this); checkMatrixItem(item);});

    $('.js-matrix-item').on('keypress', function(e) {
        var keyCode = e.keyCode || e.which,
            value = $(this).val(),
            algorithm = $('.js-algorithm').val(),
            allowDec = (algorithm !== '3');

        if(allowDec) {

            if (((keyCode<48||keyCode > 57)&&keyCode!==47&&keyCode!==46) ||
                (value.length!== 1&&(keyCode===47||keyCode===46)) ||
                (value.length=== 1&&(keyCode!==47&&keyCode!==46)) ||
                (value.length=== 1&& value !=='1'&&keyCode===47) ||
                (value.length > 2 && value.indexOf('/') > -1) ||
                value.length > 5 ||
                value === '9'){
                e.preventDefault();

                return false;}}
            else{

            if (((keyCode < 49 || keyCode > 57) && keyCode !== 47) ||
                (value.length !== 1 && keyCode === 47) ||
                (value.length === 1 && keyCode !== 47) ||
                (value.length === 1 && value !== '1') ||
                value.length > 2){
                e.preventDefault();
                return false;
            }
        });

    $('#matrixForm').on('submit', function(e) {
        e.preventDefault();
        sendMatrix();});

    $('.js-generate-button').on('click', function(e) {
        generateRandomMatrix();
        $('.js-reduced-matrix').html('');});

    $('.js-reset-button').on('click', function(e) {
        resetMatrix();
        $('.js-reduced-matrix').html('');});

    $('.js-clear-reduced-matrix').on('change', function() {
        $('.js-reduced-matrix').html('');});

    $('.js-toggle').on('click', function(e) {
        $('.js-list').slideToggle("slow");});

    $('.js-algorithm').on('change', function(e) {
        checkMatrix();}); });

```

Listing 4.26. Funkcja \$(document).ready()

```

if __name__ == "__main__":
    app.run()
from flask import Flask, render_template, json, request
import reduce
import random
import numpy as np
import scipy.linalg as la
from scipy.stats.mstats import gmean
from sympy import *
import csv
from operator import itemgetter
app = Flask(__name__)

@app.route("/")
def main():
    return render_template('index.html')
if __name__ == "__main__":
    app.run()

```

Listing 4.27. Import bibliotek back-endowych

```

@app.route('/computeCI_CR', methods=['POST'])
def computeCI_CR():
    _matrixSize_val = int(request.form['matrixSize'])

    # Tworzenie macierzy mx o odpowiednim rozmiarze
    mx = Matrix([[1] * _matrixSize_val for _ in range(_matrixSize_val)])

    # Wypełnianie macierzy wartościami z POST'a
    for i in range(_matrixSize_val):
        for j in range(_matrixSize_val):

            mx[i, j] = convertValueToFloat(request.form[f'matrix[{i + 1}]
                                                         [{j + 1}]]')

    CI = (return_max_eigenvalue(mx) -
          _matrixSize_val) / (_matrixSize_val - 1)
    RI = return_RI(_matrixSize_val)
    CR = CI / RI

    EV = derivation(mx, _matrixSize_val)
    KI = Koczkodaj_index(mx)
    GC = geometric_consistency(mx)
    GW = Golden_Wang(mx)

    PLI = Pelaez_Lamata(mx)
    TGCI = Tirads_geometric_consistency(mx)

    RE = Relative_error(mx)
    HC = Harmonic_Consistency(mx)

    send_dict = {"CI": CI, "CR": CR, "EV": EV, "KI": KI,
                 "GC": GC, "GW": GW, "PLI": PLI, "TGCI": TGCI,
                 "RE": RE, "HC": HC}

    return send_dict

```

Listing 4.28. Funkcja computeCI_CR()


```

@app.route('/reduceButton', methods=['POST'])

def reduceButton():

    _matrixSize_val = int(request.form['matrixSize'])
    _reduceThreshold = request.form['reduceThreshold']
    _selectedAlgorithm = request.form['selectAlgorithm']

    # Tworzenie macierzy mx na podstawie wartości _matrixSize_val

    mx = Matrix([[1] * _matrixSize_val for _ in range(_matrixSize_val)])

    # Wypełnianie macierzy wartościami z formularza POST

    for i in range(_matrixSize_val):
        for j in range(_matrixSize_val):

            if i < 3 and j < 3:

                mx[i, j] = convertValueToFloat(request.form[f'matrix[{{i+1}}
                                                                    [{{j+1}}]'])

            elif i < _matrixSize_val and j < _matrixSize_val:

                mx[i, j] = convertValueToFloat(request.form[f'matrix[{{i+1}}
                                                                    [{{j+1}}]'])

    reduceThreshold_val = [0.1, 0.05, 0][int(_reduceThreshold) - 1]

    if _selectedAlgorithm == "1":

        mx_final, CI, CR = Xu_Wei(mx, reduceThreshold_val,
                                   _matrixSize_val, 0.9)
    elif _selectedAlgorithm == "2":

        mx_final, CI, CR = Xu_Wei(mx, reduceThreshold_val,
                                   _matrixSize_val, 0.5)
    elif _selectedAlgorithm == "3":

        mx_final, CI, CR = Szybowski(mx, reduceThreshold_val,
                                      _matrixSize_val)

    EV = derivation(mx_final, _matrixSize_val)
    GC = geometric_consistency(mx)
    KI = Koczkodaj_index(mx)
    GW = Golden_Wang(mx)
    PLI = Pelaez_Lamata(mx)
    TGCI = Tirads_geometric_consistency(mx)
    RE = Relative_error(mx)
    HC = Harmonic_Consistency(mx)

    dict_1 = dict(enumerate(mx_final.flatten(), 1))
    dict_2 = {1000: CI, 1001: CR, 1002: EV, 1003: KI,
              1004: GC, 1005: GW, 1006: PLI, 1007: TGCI,
              1008: RE, 1009: HC}

    dict_1.update(dict_2)

    return dict_1

```

Listing 4.29. Funkcja reduceButton()

```

import multiprocessing
bind = "127.0.0.1:8000"
# Adres IP i port, na którym Gunicorn będzie nasłuchiwał
workers = multiprocessing.cpu_count() * 2 + 1
# Liczba workerów oparta na liczbie rdzeni CPU
timeout = 120
# Maksymalny czas oczekiwania na odpowiedź od workera
worker_class = 'sync'
# Typ workera
loglevel = 'info'
# Poziom logowania
errorlog = '/var/log/gunicorn/error.log'
# Ścieżka do pliku z logami błędów
accesslog = '/var/log/gunicorn/access.log'
# Ścieżka do pliku z logami dostępu

```

Listing 4.30. Plik konfiguracyjny usługi Gunicorn

```

server {
    listen 80;
    server_name reduce.prz.edu.pl;
    location / {
        return 301 https://$host$request_uri;
    }
}

server {
    listen 443 ssl;
    server_name reduce.prz.edu.pl;

    ssl_certificate
        /etc/letsencrypt/live/reduce.prz.edu.pl/fullchain.pem;
    ssl_certificate_key
        /etc/letsencrypt/live/reduce.prz.edu.pl/privkey.pem;

    location /static/ {
        alias /reduce.prz.edu.pl/static/;
        expires 30d;
    }

    location / {
        proxy_pass http://127.0.0.1:8000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }

    error_log /var/log/nginx/reduce.prz.edu.pl.error.log;
    access_log /var/log/nginx/reduce.prz.edu.pl.access.log;
}

```

Listing 4.31. Plik konfiguracyjny NGINX

```

sudo ln -s /etc/nginx/sites-available/reduce /etc/nginx/sites-enabled
sudo nginx -t
sudo systemctl restart nginx

```

Listing 4.32. Polecenia uruchamiające NGINX

```
sudo apt-get update
sudo apt-get install software-properties-common
sudo add-apt-repository ppa:certbot/certbot
sudo apt-get update
sudo apt-get install certbot
```

Listing 4.33. Polecenia instalacyjne narzędzia Certbot

```
sudo certbot --nginx -d reduce.prz.edu.pl
```

Listing 4.34. Generowanie certyfikatu za pomocą narzędzia Certbot

```
0 */12 * * * sudo certbot renew --quiet
```

Listing 4.35. Polecenie crontab okresowo odnawiające certyfikat

```
sudo ufw enable

sudo ufw allow 80/tcp
sudo ufw allow 443/tcp

sudo ufw allow 22/tcp

sudo ufw default deny incoming
sudo ufw default allow outgoing

sudo ufw status
```

Listing 4.36. Konfiguracja Uncomplicated Firewall

```
sudo apt-get update
sudo apt-get install snort
```

Listing 4.37. Polecenia instalacyjne Snorta

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS
(msg:"SQL Injection Attempt"; pcre:"/(\%27)|(\')|(\-\-\)|(\%23)|(\#)/i";
 classtype:web-application-attack; sid:1000001;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS
(msg:"XSS Attack Attempt"; pcre:"/<script>/i";
 classtype:web-application-attack; sid:1000002;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS
(msg:"Potential DDoS Attack"; flags:S; threshold:type threshold,
 track by_src, count 100, seconds 10; classtype:attempted-dos;
 sid:1000005;)
```

Listing 4.38. Reguły bezpieczeństwa dla aplikacji REDUCE

```
sudo snort -c /etc/snort/snort.conf -i localhost
```

Listing 4.39. Uruchomienie Snorta

```
nat (inside,outside) after-auto source dynamic any interface  
access-list outside_access_in extended permit tcp  
any interface outside eq 80  
access-list outside_access_in extended permit tcp  
any interface outside eq 443  
access-list outside_access_in extended permit tcp  
any interface outside eq 22  
access-group outside_access_in in interface outside
```

Listing 4.40. Konfiguracja reguł NAT i ACL w firewallu Cisco ASA 5505

```
policy-map global_policy  
class inspection_default  
inspect icmp
```

Listing 4.41. Inspekcja ruchu ICMP w firewallu Cisco ASA 5505

```

asa5505# show running-config
: Saved
:
ASA Version 9.2(4)
!
hostname asa5505

enable password 8Ry2YjIyt7RRXU24 encrypted
names

!
interface Ethernet0/0
  switchport access vlan 2

!
interface Ethernet0/1

!
interface Vlan1

  nameif inside
  security-level 100
  ip address 192.168.1.1 255.255.255.0

!
interface Vlan2

  nameif outside
  security-level 0
  ip address 203.0.113.2 255.255.255.248

!
ftp mode passive
object network obj_any
  subnet 0.0.0.0 0.0.0.0

access-list outside_access_in extended permit
tcp any object obj_any eq www

access-list outside_access_in extended permit
tcp any object obj_any eq https

pager lines 24
logging asdm informational

mtu inside 1500
mtu outside 1500

no failover
icmp unreachable rate-limit 1 burst-size 1

no asdm history enable
arp timeout 14400

no arp permit-nonconnected

...

```

Listing 4.42. Bieżąca konfiguracja firewala Cisco ASA 5505

```

interface Vlan1
  nameif inside
  security-level 100
  ip address 192.168.1.1 255.255.255.0
interface Vlan2
  nameif outside
  security-level 0
  ip address [public_IP] 255.255.255.0

```

Listing 4.43. Definiowanie interfejsów firewala Cisco ASA 5505

```

Starting Nmap 7.40 ( https://nmap.org ) at 2023-08-15 12:54 UTC
Nmap scan report for 89.188.213.253
Host is up (0.11s latency).
PORT      STATE      SERVICE
21/tcp    filtered  ftp
22/tcp    filtered  ssh
23/tcp    filtered  telnet
80/tcp    open      http
110/tcp   filtered  pop3
143/tcp   filtered  imap
443/tcp   open      https
3389/tcp  filtered  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 2.17 seconds

```

Listing 4.44. Wyniki skanowania IP aplikacji reduce.prz.edu.pl dla najbardziej powszechnych portów

```

Starting Nmap 7.40 ( https://nmap.org ) at 2023-08-15 13:05 UTC
Nmap scan report for 89.188.213.253
Host is up (0.11s latency).

PORT      STATE      SERVICE
25/tcp    closed    smtp
53/tcp    closed    domain
135/tcp   closed    msrpc
139/tcp   closed    netbios-ssn
3306/tcp  closed    mysql
8080/tcp  closed    http-proxy
8081/tcp  closed    blackice-icecap
12345/tcp closed    netbus
20000/tcp closed    dnp

Nmap done: 1 IP address (1 host up) scanned in 2.10 seconds

```

Listing 4.45. Wyniki skanowania IP aplikacji reduce.prz.edu.pl dla kolejnych popularnych portów

```

[**] [1:1000002:1] Possible XSS Attack Detected [**]
[Classification: Web Application Attack] [Priority: 1]
08/15-13:30:25.14354 192.168.1.10:80 -> 89.188.213.253:80
TCP TTL:64 TOS:0x0 ID:54465 IpLen:20 DgmLen:100 DF
***AP*** Seq: 0x4568745 Ack: 0x90D4FC Win: 0x5748 TcpLen: 20
[Xref => http://www.snort.org/search/sid/1000001]

```

Listing 4.46. Logi aplikacji Snort w przypadku wykrycia ataku XSS

```

[**] [1:1000001:1] Possible SQL Injection Attack Detected [**]
[Classification: Web Application Attack] [Priority: 1]
08/15-14:15:30.32441 192.168.1.20:80 -> 89.188.213.253:80
TCP TTL:64 TOS:0x0 ID:43523 IpLen:20 DgmLen:150 DF
***AP*** Seq: 0x9064544 Ack: 0x2865387 Win: 0x5911 TcpLen: 20
[Xref => http://www.snort.org/search/sid/1000002]

```

Listing 4.47. Logi aplikacji Snort w przypadku wykrycia ataku SQL Injection

```

[**] [1:1000005:1] Possible DDoS Attack Detected [**]
[Classification: Attempted Denial of Service] [Priority: 1]
08/15-15:45:45.785212 192.168.1.30:80 -> 89.188.213.253:80
TCP TTL:64 TOS:0x0 ID:54634 IpLen:20 DgmLen:200 DF
***AP*** Seq: 0x9634512 Ack: 0x5434634 Win: 0x7254 TcpLen: 20
[Xref => http://www.snort.org/search/sid/1000005]

```

Listing 4.48. Logi aplikacji Snort w przypadku wykrycia ataku DDoS

```

import numpy as np
import scipy.linalg as la
from scipy.stats.mstats import gmean
from sympy import *

class WrongMatrixShape(Exception):
    pass

class CrCalc:
    def __init__(self, m: np.matrix):
        if m.shape[0] != m.shape[1]:
            raise WrongMatrixShape(f"Shape {m.shape} is not proper.")
        self.shape = m.shape
        self.matrix = m
        self.SIZE = m.shape[0]

    def max_eigenvalue(self):
        m = np.array(self.matrix).astype(np.float64)
        evals, evecs = la.eig(m)
        evals = evals.real
        eigen_value_max = max(evals)
        return eigen_value_max

    @staticmethod
    def __select_ri(size):
        val = { 3: 0.5247, 4: 0.8815, 5: 1.1086, 6: 1.2479,
              7: 1.3417, 8: 1.4056, 9: 1.4499, 10: 1.4854,
              11: 1.5140, 12: 1.5365, 13: 1.5551, 14: 1.5713}
        if size in val.keys():
            return val[size]
        return 0

    def calc(self):
        max_vals = self.max_eigenvalue()
        ci = (max_vals - self.SIZE) / (self.SIZE - 1)
        ri = self.__select_ri(self.SIZE)
        cr = ci / ri
        return cr

```

Listing 5.1. Plik CR_calc.py

```

import pandas as pd
import numpy as np

class Reader():
    @staticmethod
    def load(file, sep=';'):
        return pd.read_csv(file, sep=sep)

```

Listing 5.2. Plik FileLoader.py

```

import numpy as np
from CR_calc import CrCalc

def calc_cr(data):
    # A11,A12,A13,A14,A21,A22,A23,A24,A31,A32,A33,A34,A41,A42,A43,CR
    m1 = [data["A11"], data["A12"], data["A13"], data["A14"]]
    m2 = [data["A21"], data["A22"], data["A23"], data["A24"]]
    m3 = [data["A31"], data["A32"], data["A33"], data["A34"]]
    m4 = [data["A41"], data["A42"], data["A43"], data["A44"]]
    m = np.matrix([m1, m2, m3, m4])
    max_vals = self.max_eigenvalue()
    ci = (max_vals - self.SIZE) / (self.SIZE - 1)
    ri = self.__select_ri(self.SIZE); cr = ci / ri;
    return cr()

```

Listing 5.3. Plik helpers.py

```

import pandas
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import cross_val_score, train_test_split
from sklearn.neural_network import MLPRegressor
from sklearn.tree import DecisionTreeClassifier, DecisionTreeRegressor
from sklearn import svm
from sklearn import tree
from sklearn.neighbors import KNeighborsRegressor

import FileLoader
from CR_calc import CrCalc
from helpers import calc cr

```

Listing 5.4. Plik main.py – sekcja importów


```

class Calc():
    def __init__(self):

self.scores = []
self.predictions = {}

self.X = FileLoader.Reader.load("matrix_with_vals2.csv",
sep=',')
self.y = FileLoader.Reader.load("matrix_szyb.csv", sep=',')

self.X_train, self.X_test, self.y_train, self.y_test =
train_test_split(self.X, self.y, test_size=0.2, random_state=42)

self.X_train.to_csv(f"in/_X_train.csv")
self.X_test.to_csv(f"in/X_test.csv")
self.y_train.to_csv(f"in/y_train.csv")
self.y_test.to_csv(f"in/y_test.csv")

self.create_classifiers_list()
self.fit_classifiers_list()
self.calc_real_cr()

def create_classifiers_list(self):
    self.classifiers = [
        LinearRegression(),
        KNeighborsRegressor(),
        DecisionTreeRegressor(),
        MLPRegressor(),]

def fit_classifiers_list(self):
    for clf in self.classifiers:
        _ = clf.fit(self.X_train, self.y_train)
        predictions = clf.predict(self.X_test)
        predictions = pandas.DataFrame(predictions,
columns = self.y.columns)
        self.predictions[str(clf)] = predictions

def calc_real_cr(self):
    for clf, val in self.predictions.items():
        val["cr_real"] = val.apply(calc_cr, axis=1)
        val.to_csv(f"out/{clf}_Y_predicted.csv")

```

Listing 5.5. Plik main.py – klasa Calc