

**POLITECHNIKA RZESZOWSKA**

im. Ignacego Łukasiewicza

Wydział Elektrotechniki i Informatyki

**ROZPRAWA DOKTORSKA**

mgr inż. Anna Czmił

Usprawnienie procesu diagnostyki medycznej  
przy użyciu metod sztucznej inteligencji

w formie cyklu publikacji naukowych

Promotor

dr. hab. inż. Damian Mazur, prof. PRz

Kopromotor

dr hab. n. med. Bogdan Obrzut, prof. UR

Rzeszów, czerwiec 2023



## Podziękowania

Pragnę serdecznie podziękować osobom, które przyczyniły się do powstania tej rozprawy. Prof. Damianowi Mazurowi, promotorowi mojej pracy doktorskiej, dziękuję za czuwanie nad realizacją poszczególnych zadań przez cały okres moich studiów doktorskich, za poświęcony czas i wszystkie cenne rady. Kopromotorowi, prof. Bogdanowi Obrzutowi dziękuję za wnikliwe uwagi, które przyczyniły się do ulepszenia pracy.

Chciałabym także serdecznie podziękować prof. Jackowi Klusce za dzielenie się wiedzą, doświadczeniem oraz cennymi uwagami, które pomogły mi w tworzeniu niniejszej pracy, a także pozwoliły zrozumieć wiele innych zagadnień. Dziękuję również wszystkim współautorom publikacji, dzięki którym powstał ten cykl.

Szczególne podziękowania składam mojemu kochanemu Mężowi za wszelką pomoc i okazane wsparcie. Dziękuję, że mogliśmy razem prowadzić badania naukowe, wymieniać się pomysłami oraz wspólnie przeżywać sukcesy i porażki.





# Spis treści

<b>1. Wprowadzenie</b> . . . . .	<b>7</b>
<b>2. Metody sztucznej inteligencji w diagnostyce medycznej</b> . . . . .	<b>13</b>
2.1. Diagnostowanie cukrzycy na podstawie aktywności fizycznej . . . . .	13
2.2. Zastosowanie głębokiej sieci neuronowej do rozpoznawania komórek krwi	17
2.3. Automatyzacja procesów oceny jakości oraz składania genomów prokariotycznych . . . . .	21
2.4. Zastosowanie programowania ekspresji genów do wydobywania metareguł z danych medycznych . . . . .	26
2.5. Porównanie rozmytych klasyfikatorów opartych na regułach i metaregułach . . . . .	31
<b>3. Podsumowanie i wnioski</b> . . . . .	<b>39</b>
<b>Literatura</b> . . . . .	<b>45</b>
<b>Dorobek naukowy autorki</b> . . . . .	<b>57</b>
<b>Wykaz stosowanych oznaczeń</b> . . . . .	<b>59</b>
<b>Artykuły naukowe wchodzące w skład cyklu (opublikowane w latach 2019-2023)</b> . . . . .	<b>61</b>
A method to detect type 1 diabetes based on physical activity measurements using a mobile device . . . . .	63
Automatic Detection and Counting of Blood Cells in Smear Images Using RetinaNet . . . . .	79
NanoForms: an integrated server for processing, analysis and assembly of raw sequencing data of microbial genomes, from Oxford Nanopore technology . . . . .	101
GPR: A Python implementation of an extremely simple classifier based on fuzzy logic and gene expression programming . . . . .	115
A Comparative Study of Rule-based Fuzzy Logic Classifiers for Medical Applications . . . . .	123
<b>Streszczenie w języku polskim</b> . . . . .	<b>143</b>
<b>Streszczenie w języku angielskim</b> . . . . .	<b>145</b>
<b>Oświadczenia współautorów</b> . . . . .	<b>147</b>



# 1. Wprowadzenie

Niniejsza rozprawa doktorska stanowi jednotematyczny cykl publikacji naukowych dotyczących zastosowania metod sztucznej inteligencji w celu usprawnienia procesu diagnostycznego w medycynie. Praca ma charakter interdyscyplinarny i zawiera konkretne przykłady zastosowania tychże metod, które zostały zaimplementowane i wdrożone oraz pozwoliły na rozwiązanie wielu problemów z obszaru diagnostyki medycznej. Należą do nich: wykrywanie cukrzycy typu 1 u dzieci i młodzieży, automatyczne zliczanie komórek krwi na podstawie zdjęć mikroskopowych z użyciem głębokiej sieci neuronowej, automatyzacja procesu oceny i składania sekwencji genomowych uzyskanych za pomocą nowych metod sekwencjonowania (next-generation sequencing, NGS), implementacja w języku Python algorytmu GPR, który pozwala na generowanie wysoce interpretowalnych metareguł oraz porównanie rozmytych algorytmów regułowych w zastosowaniach medycznych.

W skład prezentowanej rozprawy doktorskiej wchodzi pięć publikacji:

- [A-1] **Czmił, A.** Czmił, S., & Mazur, D. (2019). *A Method to Detect Type 1 Diabetes Based on Physical Activity Measurements Using a Mobile Device*. *Applied Sciences*, 9(12), 2555. doi:10.3390/app9122555. IF\_2019 = 2,474, IF\_2022 = 2,838, liczba punktów: 70, obecnie 100, wkład: 33,33%.
- [A-2] Drałus, G., Mazur, D., & **Czmił, A.** (2021). *Automatic Detection and Counting of Blood Cells in Smear Images Using RetinaNet*. *Entropy*, 23(11), 1522. doi:10.3390/e23111522. IF\_2021 = 2,738, liczba punktów: 100, wkład: 33,33%.
- [A-3] **Czmił, A.**, Wroński, M., Czmił, S., Sochacka-Piętal, M., Ćmił, M., Gawor, J., Wołkowicz, T., Plewczyński, D., Strzałka, D., & Piętal, M. (2022). *NanoForms: an integrated server for processing, analysis and assembly of raw sequencing data of microbial genomes, from Oxford Nanopore technology*. *PeerJ*, 10, e13056. doi:10.7717/peerj.13056. IF\_2022 = 3,061, liczba punktów: 100, wkład: 10%.
- [A-4] **Czmił, A.**, Kluska, J., & Czmił, S. (2023). *GPR: A Python implementation of an extremely simple classifier based on fuzzy logic and gene expression programming*. *SoftwareX*, 22, 101362. doi:10.1016/j.softx.2023.101362. IF\_2022 = 2,868, liczba punktów: 200, wkład: 33,33%.

[A-5] Czmił, A. (2023). *Comparative Study of Fuzzy Rule-Based Classifiers for Medical Applications*. *Sensors*, 23(2), 992. doi:10.3390/s23020992. IF\_2022: 3,847, liczba punktów: 100, wkład: 100%.

Wszystkie artykuły naukowe wchodzące w skład cyklu znajdują się w czasopiśmie wyszczególnionych na liście czasopism punktowanych Ministerstwa Nauki i Szkolnictwa Wyższego i zostały opublikowane w latach 2019-2023. Sumaryczny Impact Factor (zgodnie z rokiem ukazania się publikacji) wynosi 14,988, a liczba punktów MNiSW (punktacja czasopism naukowych zgodnie z wykazem MNiSW) wynosi 570.

## Motywacja oraz stan wiedzy

Koncepcja wykorzystania komputerów do symulacji inteligentnego zachowania i krytycznego myślenia została po raz pierwszy opisana przez Alana Turinga w 1950 r. W pracy [1] przedstawił on test, który później stał się znany jako „test Turinga”. Jego celem było ustalenie sposobu określania zdolności maszyny do posługiwania się językiem naturalnym, a pośrednio udowodnienie opanowania przez nią umiejętności myślenia w sposób podobny do ludzkiego [2]. Sześć lat później John McCarthy zaproponował termin sztuczna inteligencja (artificial intelligence, AI), którą zdefiniował jako naukę obejmującą inżynierię tworzenia inteligentnych maszyn, a zwłaszcza inteligentnych programów komputerowych. Definicja ta ma zarówno zwolenników, jak i przeciwników. Ci drudzy twierdzą, że zaawansowane zachowania i stany, takie jak miłość, kreatywność czy wybory moralne, zawsze będą poza zasięgiem jakiegokolwiek maszyny czy programu komputerowego [3].

Od połowy ubiegłego wieku naukowcy badają potencjalne zastosowania technik inteligentnych w wielu różnych dziedzinach medycyny. W latach 70. XX wieku na Uniwersytecie Stanford stworzono regułowy system ekspertowy MYCIN pozwalający na zdiagnozowanie bakteryjnej choroby krwi i zaproponowanie odpowiedniej terapii [4]. Koncepcja systemów ekspertowych opiera się na wykorzystaniu wiedzy i doświadczenia ekspertów w celu opracowania algorytmów wnioskowania, które pozwolą na podejmowanie decyzji bez bezpośredniego udziału ludzkiego eksperta. System ekspertowy może działać samodzielnie lub wspomagać eksperta w podejmowaniu decyzji, dostarczając mu alternatywnych rozwiązań problemów. Kolejnymi zaproponowanymi systemami ekspertowymi były PUFF i Icons, które pozwalają kolejno na diagnozowanie i leczenie chorób płuc oraz udzielanie porad dotyczących terapii antybiotykowej dla

chorych z oddziałów intensywnej terapii [5]. Innym przykładem systemu ekspertowego wykorzystywanego w medycynie jest CASNET, którego zadaniem jest diagnoza, interpretacja oraz terapia stanów chorobowych związanych z jaskrą [6]. Wśród systemów ekspertowych wykorzystywanych w obszarze medycyny zaproponowano także systemy, takie jak QMR, ELSA, AEGIS, HERMES czy AMIGO, których zadaniem jest również pomoc lekarzom w procesie diagnozowania pacjentów. Systemy ekspertowe mogą być także stosowane w autodiagnozie i autoleczeniu [5].

Postępy w dziedzinie uczenia maszynowego i sieci neuronowych umożliwiły wspieranie diagnostyki różnych chorób i stanów patologicznych. W 2007 roku firma IBM przedstawiła system Watson, który wykorzystuje zaawansowaną technologię DeepQA do analizy nieustrukturyzowanych danych, opartą na przetwarzaniu języka naturalnego oraz dużą ilość danych pochodzących z różnych dziedzin [7]. Wykorzystanie tej innowacyjnej technologii w połączeniu z elektroniczną dokumentacją medyczną pacjenta oraz innymi zasobami elektronicznymi otworzyło drogę do zastosowania jej w celu uzyskania odpowiedzi medycznych opierających się na dowodach. W 2017 roku IBM Watson został zastosowany do identyfikacji nowych białek wiążących RNA, które odgrywają kluczową rolę w patogenezie stwardnienia zanikowego bocznego [8]. Jego działanie przetestowano także w badaniach pilotażowych dotyczących identyfikacji celów leków czy zmiany ich przeznaczenia [7].

Zaproponowany w 2022 roku przez firmę Open AI Chat GPT to narzędzie cieszące się ogromną popularnością, także w kontekście zastosowań medycznych [9]. Dzięki wykorzystaniu uczenia maszynowego jest w stanie dyskutować z użytkownikiem, a także napisać tekst o różnej długości na podstawie dostarczonego przez użytkownika założenia. Połączenie algorytmów sztucznej inteligencji oraz olbrzymiej bazy danych umożliwia wykonywanie przez czat poleceń, takich jak tłumaczenie treści na języki obce czy generowanie kodu programu. Jego możliwości mogą zostać również wykorzystane w sektorze ochrony zdrowia w zakresie zapewnienia wsparcia w wypełnianiu dokumentacji medycznej, a także jej korygowania. Za pomocą czatu można się także zorientować, jakie dolegliwości czy choroby mogą się kryć za podanymi przez użytkownika objawami, a także poznać leki dostępne bez recepty lub naturalne sposoby leczenia, a w przypadku niepokojących objawów uzyskać informację o konieczności pilnej konsultacji z lekarzem. Narzędzie może również stanowić rodzaj innowacyjnego wsparcia dla personelu medycznego w podejmowaniu decyzji medycznych czy zdalnym monitoringu pacjenta.

Zastosowanie metod sztucznej inteligencji obejmuje różne dziedziny medycyny. Dalej omówiono kilka przykładów ich zastosowania. W radiologii sztuczna inteligencja jest wykorzystywana do analizy obrazów medycznych zarówno w celach diagnostycznych, jak i terapeutycznych. Wspomniane metody pozwalają na automatyczne rozpoznawanie złożonych wzorców na obrazach, dostarczają ilościowej oceny cech radiologicznych, umożliwiając wytyczenie obszaru zmian nowotworowych oraz odkrywanie cech choroby, których nie widać gołym okiem [10, 11]. W kardiologii sztuczna inteligencja jest stosowana w leczeniu oraz zapobieganiu chorobie wieńcowej. Pozwala na automatyczną interpretację zaburzeń rytmu serca, modelowanie ryzyka choroby czy przewidywanie rokowania pacjentów [12]. W dermatologii może pomóc w diagnozowaniu i leczeniu chorób skóry, takich jak łuszczyca, atopowe zapalenie skóry, grzybica paznokci czy nowotwory [13]. W onkologii jest stosowana w celu wspierania decyzji klinicznych w diagnostyce oraz badaniach przesiewowych w kierunku różnych rodzajów nowotworów, przetwarzania danych w celu wykrywania nowotworów lub oceny rokowania pacjentów [14]. W genomice szczególnie ważną rolę odgrywają głębokie sieci neuronowe, które są stosowane do przetwarzania dużych i złożonych zbiorów danych genetycznych. Mają także duży potencjał w przewidywaniu ryzyka wystąpienia chorób oraz medycynie spersonalizowanej [15]. W neurologii wykorzystuje się różne dziedziny sztucznej inteligencji, obejmujące uczenie nienadzorowane, systemy monitorujące ruchy i fazy drżenia, algorytmy pozwalające na analizę sygnałów z elektroencefalografii (EEG), ocenę funkcji ruchowych, identyfikację wzorców niestabilności autonomicznej czy przewidywanie wyników operacyjnego leczenia padaczki [16]. W diabetologii sztuczna inteligencja może być stosowana do diagnostyki powikłań, monitorowania glikemii w czasie rzeczywistym w celu osiągnięcia jej optymalnego wyrównania, predykcji stanu cukrzycy u pacjenta, może także wspierać prowadzenie zdrowego stylu życia i przestrzeganie zaleceń dotyczących przyjmowania leków [17]. Sztuczna inteligencja jest także stosowana w hematologii, przy rozpoznawaniu rodzajów komórek krwi, wspomaga diagnostykę nowotworów hematologicznych i stopnia ich zaawansowania oraz innych chorób hematologicznych [18].

Wykorzystanie sztucznej inteligencji w diagnostyce medycznej przynosi wiele obiecujących rezultatów, wciąż jednak istnieją pewne wyzwania i ograniczenia. Jednym z największych problemów wciąż pozostają kwestie etyczne. Istnieją również obawy, że niektóre decyzje medyczne podjęte przez narzędzia kliniczne oparte na sztucznej

inteligencji mogą działać na niekorzyść pacjenta, chociażby ze względu na ograniczenia w zakresie inteligencji emocjonalnej i empatii. Jednocześnie dostosowanie obecnych praktyk odpowiedzialności i bezpieczeństwa związanych z wykorzystaniem sztucznej inteligencji w medycynie jest wciąż niedostateczne. Pojawiają się również obawy dotyczące prywatności pacjentów i bezpieczeństwa danych medycznych, szczególnie tych przechowywanych w chmurze, które wymagają stosowania odpowiednich procedur bezpieczeństwa i przestrzegania odpowiednich przepisów. Kolejnym ograniczeniem jest to, że sztuczna inteligencja wymaga dużej ilości danych medycznych, aby dokładnie przewidywać wyniki leczenia. Konieczne jest zatem zapewnienie odpowiedniej jakości danych, ponieważ mają one bezpośredni wpływ na wyniki diagnostyczne [19, 20, 21]. Systemy oparte na sztucznej inteligencji mogą być stosowane do rozpoznawania konkretnego przypadku, do którego zostały zaprojektowane i wytrenowane, jednak ich zakres wiedzy jest limitowany. Ponadto, spektrum zadań poznawczych i społecznych sztucznej inteligencji w różnorodnych, nieprzewidzianych okolicznościach jest znacznie węższe od eksperta ludzkiego [22].

Reasumując, wykorzystanie sztucznej inteligencji w medycynie jest obiecujące i może pomóc poprawić jakość życia pacjentów, zwiększyć skuteczność diagnostyki i terapii, a także zoptymalizować koszty leczenia. Wciąż jednak istnieją liczne ograniczenia i wyzwania, którym trzeba sprostać. W związku z tym konieczne jest ciągłe doskonalenie istniejących metod, prowadzenie dalszych badań oraz proponowanie nowych rozwiązań dotyczących wykorzystania sztucznej inteligencji w medycynie.

## **Hipoteza badawcza oraz cele pracy**

Głównym celem pracy jest przedstawienie możliwości zastosowania metod sztucznej inteligencji do poprawy jakości i skuteczności procesu diagnostyki medycznej. Opierając się na tym założeniu, w pracy sformułowano hipotezę, która zakłada, że

*Możliwe jest wykorzystanie różnych metod sztucznej inteligencji do analizy danych medycznych i automatyzacji wybranych procesów diagnostycznych, pozwalające na uzyskanie interpretowalnych wyników z dokładnością i efektywnością nie gorszą niż innych istniejących metod znanych z literatury.*

W celu potwierdzenia postawionej hipotezy sformułowano następujące zadania szczegółowe:

- 1) Zastosowanie metod sztucznej inteligencji do klasyfikacji cukrzycy typu 1 na podstawie danych uzyskanych za pomocą nieinwazyjnych pomiarów aktywności fizycznej.
- 2) Opracowanie metody pozwalającej na automatyczne, jednoczesne rozpoznawanie i zliczanie czerwonych i białych krwinek oraz płytek krwi na podstawie zdjęć mikroskopowych z wykorzystaniem głębokich sieci neuronowych.
- 3) Opracowanie aplikacji umożliwiającej automatyzację procesu oceny, składania i identyfikacji sekwencji genomowych uzyskanych za pomocą nowych metod sekwencjonowania przy użyciu narzędzi korzystających z metod uczenia maszynowego.
- 4) Implementacja w języku Python klasyfikatora opartego na logice rozmytej i programowaniu ekspresji genów, służącego do generowania wysoce interpretowalnych reguł rozmytych.
- 5) Opracowanie narzędzia pozwalającego na eksperymentalne porównanie wybranych rozmytych algorytmów opartych na regułach do klasyfikacji danych medycznych.

Osiągnięcie celu dysertacji polega na realizacji wyszczególnionych zadań, które wymagają pozyskania, przetworzenia i analizy danych, sformułowania konkretnych zadań i wyboru najbardziej odpowiednich narzędzi informatycznych do ich rozwiązania, opracowania oprogramowania, przeprowadzenia testów zaproponowanych rozwiązań, interpretacji uzyskanych wyników oraz sformułowania wniosków.



## **2. Metody sztucznej inteligencji w diagnostyce medycznej**

W tym rozdziale przedstawiono proponowane zastosowania metod sztucznej inteligencji w diagnostyce medycznej. W kolejnych podrozdziałach opisano kolejno nieinwazyjną metodę wykrywania cukrzycy typu 1, poruszono problematykę rozpoznawania komórek krwi na zdjęciach z rozmazu krwi za pomocą głębokiej sieci neuronowej oraz przedstawiono możliwości automatyzacji procesów oceny jakości i składania genomów prokariotycznych. Ostatnie dwa podrozdziały dotyczą zastosowania programowania ekspresji genów do wydobywania metareguł z danych medycznych oraz porównania rozmytych klasyfikatorów opartych na regułach i metaregułach.

### **2.1. Diagnostowanie cukrzycy na podstawie aktywności fizycznej**

Cukrzyca (diabetes mellitus) stanowi grupę chorób metabolicznych charakteryzujących się hiperglikemią – zbyt wysokim stężeniem glukozy we krwi [23]. Typ 1 cukrzycy, w którym organizm nie produkuje wystarczającej ilości insuliny, jest szczególnie powszechny wśród dzieci i młodzieży [24, 25]. Współcześnie jest ona najczęściej diagnozowaną chorobą przewlekłą okresu dziecięcego, a wskaźnik zachorowań w Polsce wynosi obecnie 18-25 przypadków rocznie na 100 000 mieszkańców [26]. Co więcej, wzrasta liczba nowych przypadków cukrzycy wśród dzieci w wieku do szóstego roku życia [27]. Podstawowym sposobem leczenia tej choroby jest głównie podawanie insuliny, zmiana nawyków żywieniowych i wprowadzenie umiarkowanej aktywności fizycznej [28, 29].

Opóźnione rozpoznanie cukrzycy u dzieci może prowadzić do poważnych powikłań dotyczących układu nerwowego, takich jak neuropatia obwodowa i autonomiczna, układu krążenia, w tym miażdżycy, choroby wieńcowej i udaru mózgu, a także chorób narządów wewnętrznych, takich jak retinopatia cukrzycowa, nefropatia cukrzycowa czy uszkodzenie słuchu [30, 31]. Osoby cierpiące na cukrzycę są szczególnie narażone na depresję, choroby nerwicowe, niealkoholowe stłuszczenie wątroby, choroby przyzębia, utratę słuchu oraz powikłania podczas ciąży [32, 33, 34].

Diagnostyka cukrzycy wymaga pomiaru stężenia glukozy w osoczu krwi, które można wykonać w medycznym laboratorium diagnostycznym lub w warunkach domowych za pomocą glukometru. Można ją także rozpoznać po stwierdzeniu podwyższo-

nego stężenia hemoglobiny glikowanej (HbA1c) we krwi [35]. Aby wykonać badanie za pomocą glukometru, należy pobrać kroplę krwi z opuszki palca pacjenta i umieścić ją na specjalnym pasku testowym [36]. Aktualne kryteria rozpoznawcze cukrzycy uznawane zarówno przez Amerykańskie Stowarzyszenie Diabetyków (ADA), jak i Światową Organizację Zdrowia (WHO) obejmują stwierdzenie podwyższonego stężenia glukozy na czczo, natomiast WHO uznaje badanie obciążenia glukozą za dodatkowy czynnik diagnostyczny choroby [23].

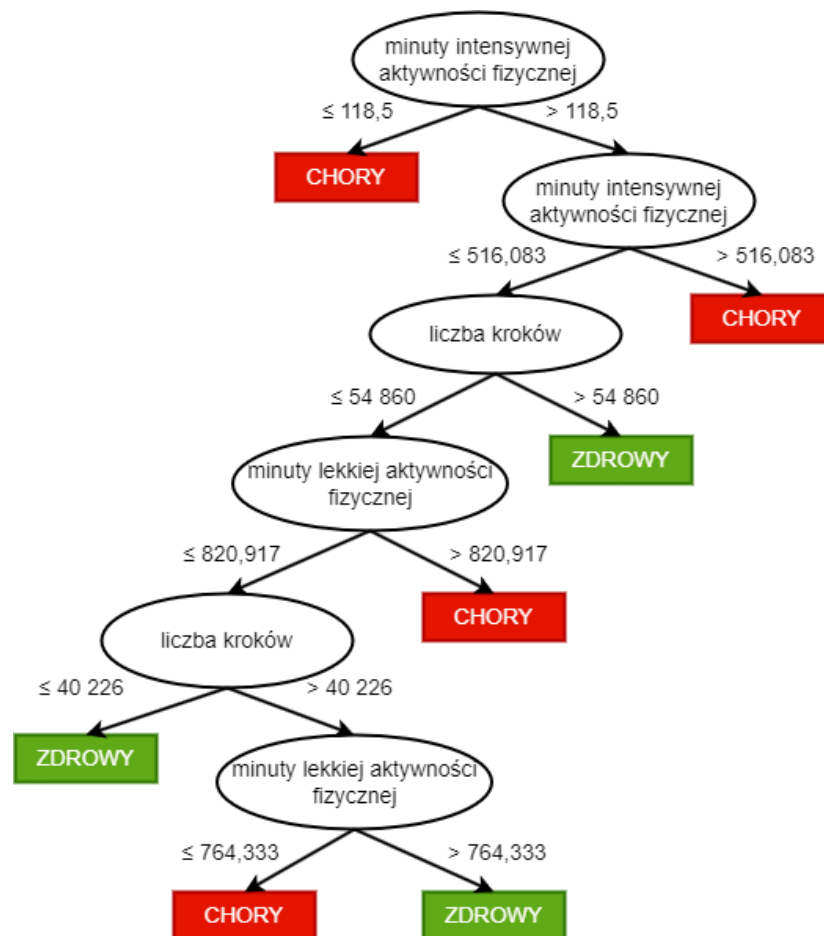
Celem pracy [A-1] wchodzącej w skład rozprawy doktorskiej było zaproponowanie nieinwazyjnej metody wykrywania cukrzycy typu 1, opartej na pomiarze aktywności fizycznej. Zostały do tego wykorzystane rzeczywiste dane pochodzące z badań przeprowadzonych w latach 2014–2016 przez Ewelinę Czenczek–Lewandowską [37]. Zbiór danych obejmował grupę 230 dzieci w wieku od 6 do 18 lat, w tym 115 dzieci zdrowych oraz 115 dzieci z cukrzycą typu 1, będących pod opieką Poradni Cukrzycowej dla dzieci w Klinicznym Szpitalu Wojewódzkim nr 2 w Rzeszowie. Każdy rekord danych zawierał dziewięć parametrów:

- wiek,
- płeć,
- wagę,
- wzrost,
- tygodniową liczbę kroków,
- tygodniową liczbę minut zajęć sedenteryjnych (bardzo lekkiej aktywności fizycznej),
- tygodniową liczbę minut lekkiej aktywności fizycznej,
- tygodniową liczbę minut umiarkowanej aktywności fizycznej,
- tygodniową liczbę minut intensywnej aktywności fizycznej,
- informację o występowaniu cukrzycy typu 1.

Parametry dotyczące aktywności fizycznej były rejestrowane przez siedem kolejnych dni badania w sposób nieinwazyjny za pomocą akcelerometru ActiGraph w wersji noszonej na nadgarstku. Warto także zaznaczyć, że akcelerometry są obecnie najdokładniejszymi i najbardziej obiektywnymi czujnikami ruchu służącymi do oceny aktyw-

ności fizycznej, ponieważ wykrywają przyspieszenia ruchu ciała, dając możliwość rzetelnego pomiaru intensywności, czasu trwania aktywności fizycznej, jak również liczby wykonanych kroków i czasu spędzonego w sposób bierny.

W wyniku selekcji cech uzyskanej metodą współczynnika korelacji najważniejszymi parametrami w wykrywaniu choroby okazały się kolejno: tygodniowa liczba kroków, liczba minut wysokiej oraz umiarkowanej aktywności fizycznej. Zastosowanie algorytmu drzewa decyzyjnego pozwoliło na graficzne przedstawienie procesu decyzyjnego i ułatwienie zrozumienia, na czym polega model podejmowania decyzji. Schemat drzewa decyzyjnego został przedstawiony na rysunku 2.1.



Rysunek 2.1: Schemat drzewa decyzyjnego

W rozwiązaniu zadania klasyfikacji binarnej dotyczącej występowania cukrzycy typu 1 wśród dzieci i młodzieży zastosowano dziesięć popularnych algorytmów sztucznej inteligencji, tj. metodę wektorów wspierających (support vector machines, SVM), probabilistyczną sieć neuronową (probabilistic neural network, PNN), perceptron wielowarstwowy (multilayer perceptron, MLP), metodę grupowania argumentów (group

method of data handling, GMDH), programowanie ekspresji genów (gene expression programming, GEP), regresję liniową (linear regression), radialną funkcję bazową (radial basis function network, RBF), regresję logistyczną (logistic regression), drzewo decyzyjne (decision tree, DT) oraz las losowy (random forest, RF).

Tabela 2.1: Wartości metryk wydajnościowych obliczonych dla wszystkich badanych algorytmów klasyfikacyjnych na zbiorze danych dotyczących cukrzycy typu 1

Algorytm	Acc (%)	Sen (%)	Spe (%)	Prec (%)	G-index	AUC
<b>RF</b>	86.09	87.83	84.35	84.87	0.1983	-
<b>PNN</b>	84.35	89.57	79.13	81.10	0.2333	0.926578
<b>SVM</b>	84.35	86.96	81.74	82.64	0.2244	0.909716
<b>DT</b>	83.48	86.09	80.87	81.82	0.2365	-
<b>GEP</b>	83.04	83.48	82.61	82.76	0.2399	0.830435
<b>Logistic regression</b>	82.61	84.35	80.87	81.51	0.2472	0.883478
<b>GMDH</b>	82.61	82.61	82.61	82.61	0.2460	0.905482
<b>RBF</b>	82.17	85.22	79.13	80.33	0.2557	0.905331
<b>MLP</b>	81.30	86.09	76.52	78.57	0.2729	0.897921
<b>Linear regression</b>	80.87	85.22	76.52	78.40	0.2774	0.884008

Oceny jakości klasyfikacji binarnej dokonano za pomocą metryk wydajnościowych, takich jak dokładność (accuracy, ACC), czułość (sensitivity, Sen), specyficzność (specificity, Spe), precyzja (precision, Pre), wskaźnik dobroci (goodness index, G-index) i pole pod krzywą charakterystyki działania odbiornika (area under ROC curve, AUC). Najlepsze wyniki dokładności (86,09%), specyficzności (84,35%), precyzji (84,87%) oraz wskaźnika zgodności (0,1983) uzyskano, stosując metodę lasu drzew decyzyjnych. Największą czułość osiągnęła probabilistyczna sieć neuronowa – 89,57%.

W celu wyodrębnienia konkretnych grup danych zastosowano także metodę klastrowania. Analiza wykazała, że uzyskane grupy są zbliżone do oryginalnie przypisanych klas. W kolejnym kroku wyselekcjonowano 215 rekordów pokrywających się z oryginalnymi klasami i na ich podstawie zaimplementowano model drzewa regresji jednoelementowej. Wyniki wskazują na ryzyko zachorowania na cukrzycę typu 1 u dzieci i młodzieży pokonujących mniej niż 60 837 kroków tygodniowo, a reguła ta jest spełniona w co najmniej 65% przypadków. Schemat drzewa decyzyjnego po zastosowaniu klastrowania został przedstawiony na rysunku 2.2. Przeprowadzone badania potwierdziły możliwość diagnostyki cukrzycy typu 1 za pomocą pomiaru aktywności fi-

zycznej, który jest znacznie niższy u dzieci i młodzieży z cukrzycą typu 1 w porównaniu ze zdrowymi rówieśnikami.



Rysunek 2.2: Schemat drzewa decyzyjnego po klastrowaniu

Udział własny autorki niniejszej rozprawy doktorskiej w przygotowaniu pracy [A-2] polegał na współautorstwie koncepcji artykułu, doborze algorytmów sztucznej inteligencji do eksperymentów z uwzględnieniem przyjętej metodologii, a także współudziale w przeprowadzeniu eksperymentów obliczeniowych, opracowaniu i analizie wyników, przygotowaniu grafik oraz współredakcji pracy.

## 2.2. Zastosowanie głębokiej sieci neuronowej do rozpoznawania komórek krwi

Głębokie sieci neuronowe odgrywają szczególną rolę w inżynierii biomedycznej i aktywnie wspomagają diagnostykę medyczną [38]. Dzięki możliwościom uczenia się z dużych zbiorów danych i automatycznego wykrywania wzorców te modele sztucznej inteligencji stanowią obiecujące rozwiązanie dla wielu trudnych problemów diagnostycznych. Jednym z nich jest zliczanie komórek krwi na obrazach mikroskopowych.

Morfologia krwi jest tanim, prostym do wykonania i łatwo dostępnym badaniem, które przynosi ważne informacje pomocne w diagnostyce wielu chorób [39]. Zawiera informacje o produkcji komórek krwi, a także o ich liczebności. Analiza wyników morfologii krwi pozwala na ocenę zdolności pacjenta do transportu tlenu oraz jego układu odpornościowego. W wyniku tego testu można wykryć anemię, sepsę, niektóre nowotwory, infekcje i wiele innych chorób, a także monitorować działania niepożądane leków [40, 41].

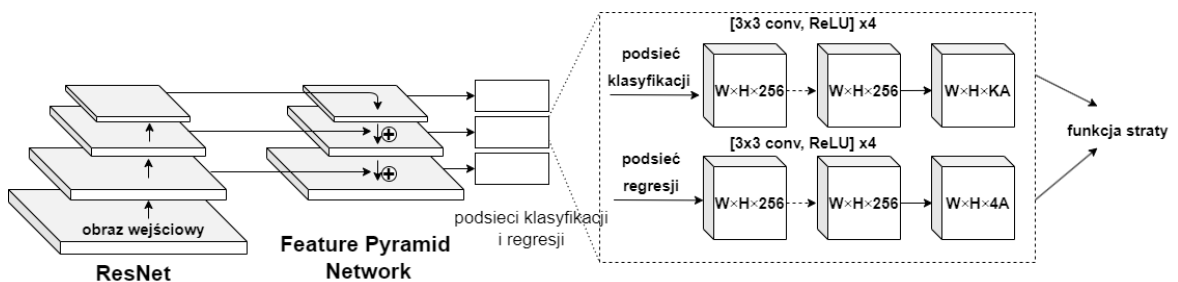
W laboratoriach medycznych codziennie pojawia się duża liczba próbek krwi, które muszą zostać szybko i skrupulatnie przebadane. Do właściwej interpretacji wyników badań, a także do precyzyjnej diagnostyki jest wymagana spora wiedza i doświadczenie personelu medycznego. Najczęściej stosowaną metodą analizy próbek krwi jest analiza mikroskopowa rozmazu krwi pod kątem liczby i jakości poszczególnych komór-

rek krwi, takich jak erytrocyty, leukocyty i trombocyty [42]. Alternatywą dla ręcznego liczenia komórek, polegającego na oglądaniu obrazów pod mikroskopem i manualnej identyfikacji komórek, są metody półautomatyczne i automatyczne. Metody półautomatyczne składają się najczęściej z kilku etapów i wymagają interakcji personelu laboratoryjnego. Metody automatyczne wykorzystują zaś zaawansowane oprogramowanie i sprzęt, które automatycznie wykrywają i zliczają komórki na obrazach.

Automatyczne metody liczenia komórek mają zarówno zalety, jak i wady. Wśród zalet można wymienić większą wydajność i dokładność w porównaniu z ręczną metodą. Należy jednak poświęcić dodatkowy czas na odpowiednie przygotowanie obrazów komórek. Wykrywanie komórek musi także zostać połączone z analizą ilościową komórek i uzyskaniem dokładnej liczby komórek na obrazie medycznym, co jest niezbędne do właściwej interpretacji wyników badań i precyzyjnej diagnostyki.

W ostatnich latach nastąpił dynamiczny rozwój dużych modeli głębokich sieci neuronowych, które ewoluowały z klasycznych sieci neuronowych. Algorytmy wykorzystujące sieci głębokie znalazły zastosowanie w analizie tekstu pisanego, syntezie mowy, a także w rozpoznawaniu mowy i obrazów, w tym również w kontekście diagnostyki medycznej [43]. Wiele niezależnych badań potwierdziło skuteczność i efektywność sieci uczenia głębokiego, zwłaszcza w zastosowaniach do automatycznego liczenia komórek krwi [44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54]. Pośród nich kilka artykułów opisywało klasyfikację i liczenie różnych typów krwinek białych i płytek krwi z wykorzystaniem głębokiej sieci neuronowej. Wciąż jednak brakowało dokładnej analizy dotyczącej doboru optymalnych parametrów w procesie uczenia, tj. określenia optymalnej liczby epok i progów, które pozwolą osiągnąć najlepsze wyniki. Ponadto, wyniki uzyskane we wcześniejszych pracach często były oceniane tylko na podstawie dokładności, która bez wątpienia jest ważną metryką, ale często niewystarczającą, np. w przypadku nierównomiernego rozkładu elementów w klasach. Z tego powodu wyniki powinny być również omówione w kontekście ważnych metryk wydajnościowych, takich jak czułość, precyzja i wynik F1. Wiele prac skupiało się na rozpoznawaniu komórek na małych obrazach (wycinkach), podczas gdy mikroskopowe obrazy medyczne często zawierają setki nakładających się na siebie i stykających ze sobą komórek, co stanowi dodatkowe wyzwanie. Celem artykułu [A-2] było opracowanie precyzyjnej i automatycznej metody jednoczesnego liczenia trzech różnych typów komórek na jednym obrazie z wykorzystaniem sieci RetinaNet.

Sieć RetinaNet to sieć neuronowa oparta na architekturze sieci konwolucyjnych (convolutional neural networks, CNN), która została zaprojektowana w 2017 roku przez badaczy z Facebook AI Research do wykrywania obiektów na obrazach [55]. Zasadniczo składa się z dwóch części: podstawy sieci (backbone) służącej do ekstrakcji cech, opartej na ResNet, oraz sieci Feature Pyramid (FPN) służącej do gromadzenia cech na różnych poziomach rozdzielczości. RetinaNet wykorzystuje także funkcję straty Focal Loss, która pozwala na radzenie sobie z problemem nierównowagi między tłem a wykrywanymi obiektami. Schemat architektury sieci przedstawiono na rysunku 2.3.



Rysunek 2.3: Architektura sieci RetinaNet

Do uczenia sieci od podstaw wykorzystano rzeczywisty zbiór danych składający się z 900 obrazów zawierających czerwone i białe krwinki oraz płytki krwi. Testy aplikacji przeprowadzono, korzystając ze zbioru obrazów leukocytów do segmentacji i klasyfikacji (Leukocyte Images for Segmentation and Classification, LISC) [56], zawierającego obrazy leukocytów do segmentacji i klasyfikacji o rozdzielczości  $720 \times 576$  uzyskane za pomocą mikroskopu świetlnego ze stukrotnym powiększeniem i rejestrowanych za pomocą aparatu cyfrowego. Przed rozpoczęciem uczenia zbiór uczący został opatrzony adnotacjami odnoszącymi się do trzech rodzajów komórek.

Następnie sieć RetinaNet ze szkieletem ResNet50 została wytrenowana do rozpoznawania i klasyfikacji komórek krwi. Sieć była uczona przez wykonywanie różnej liczby epok i zapisywanie poszczególnych modeli. Następnie dla modeli uczonych z liczbą 10, 15, 20, 25, 30, 35 i 40 epok zbadano wpływ liczby epok na spadki w funkcji straty. Wyniki uzyskane dla kolejnych epok uczenia przetestowano manualnie na 15 zdjęciach, obliczając średni wynik miary F1 dla każdej epoki. Metryka F1, będąca średnią harmoniczną precyzji i czułości, została wybrana jako wstępna miara oceny jakości detekcji i klasyfikacji komórek krwi przez model. Przy wyborze wzięto pod uwagę rozmiar wstępnego zbioru danych testowych, problem nierównoważonych danych oraz istotne

znaczenie obu tych metryk w kontekście medycznym. Na podstawie uzyskanych wyników wybrano modele trenowane za pomocą 10 i 30 epok, które osiągnęły wysokie wyniki miary F1 i stabilną funkcję straty.

Dla wybranych modeli RN10 i RN30 przeprowadzono szczegółowe badania na większej liczbie zdjęć. Z zestawu wybrano losowo 131 obrazów do liczenia białych krwinek, 64 obrazy do liczenia płytek krwi i 15 obrazów do liczenia krwinek czerwonych. Wybór różnej liczby obrazów do badań jest związany z liczbą pojedynczych komórek zawartych na każdym obrazie. Z uwagi na dużą liczbę krwinek czerwonych (średnio 121 na obraz) do ich zliczania i dalszych badań wybrano tylko 15 obrazów. Rozpoznane komórki zostały automatycznie oznaczone przez sieć za pomocą ramek ograniczających, zgodnie z reprezentowanym typem. Porównano także wyniki uzyskane dla modelu trenowanego z 10 epokami i 30 epokami.

Wyniki badań dotyczących sieci RetinaNet wykazały, że każdy typ komórki krwi ma swoją optymalną wartość progową, która pozwala na osiągnięcie najwyższej dokładności rozpoznawania i zliczenia danego typu komórek. Model RN10 po wykonaniu 10 epok uczenia był już stosunkowo dokładny w liczeniu krwinek, ale wykonanie 30 epok uczenia sprawiło, że wydajność modelu się zwiększyła. Model RN30 osiągnął dokładność liczenia krwinek czerwonych, krwinek białych i płytek krwi wynoszącą odpowiednio 99,7%, 98,6% i 97,8%. Po wykonaniu 40 epok zaobserwowano już oznaki przetrenowania modelu. W ramach badania ustalono również jeden optymalny próg wynoszący 0,45, który pozwolił na poprawne rozpoznawanie i zliczanie krwinek białych i płytek krwi oraz większości erytrocytów. Wyniki sugerują również, że wybór optymalnego modelu do liczenia komórek krwi jest trudnym zadaniem i zależy od wielu czynników, takich jak próg ufności, liczba epok oraz wybrane kryterium oceny jakości zliczania. Ze względu na mnogość dobieranych parametrów i brak jednoznacznych kryteriów, wybór optymalnego modelu do liczenia komórek krwi jest problemem otwartym.

Następnie otrzymane wyniki porównano do wyników uzyskanych przez innych autorów ([51, 52, 53, 54]) zajmujących się tematyką liczenia krwinek czerwonych, białych i płytek krwi. Wyniki porównania wskazują, że zaproponowane podejście znacznie poprawia dokładność liczenia komórek. Są one bardzo satysfakcjonujące, biorąc pod uwagę jednoczesne rozpoznawanie i liczenie trzech rodzajów komórek. Ponadto, obrazy medyczne nie wymagają dodatkowej obróbki, a wyniki są uzyskiwane po jednorazowej



prezentacji obrazu. Opracowana w tym celu aplikacja posiada potencjał do zastąpienia manualnej identyfikacji i liczenia komórek krwi.

W ramach prac dotyczących opracowania artykułu [A-2] wkład własny autorki polegał na udziale w implementacji oprogramowania służącego do identyfikacji i zliczania komórek na obrazach z rozmazów krwi, współpracowaniu metodologii i przeprowadzeniu części eksperymentów obliczeniowych, współpracowaniu i analizie wyników, przygotowaniu rysunków oraz tabel, współredakcji pracy.

### **2.3. Automatyzacja procesów oceny jakości oraz składania genomów prokariotycznych**

Sekwencjonowanie DNA jest techniką odczytywania kolejności par nukleotydowych w cząsteczce DNA, wykonywanego głównie za pomocą zautomatyzowanych sekwenatorów. Sekwencjonowanie za pomocą technologii Oxford Nanopore (ONT) jest techniką sekwencjonowania następnej generacji (NGS), sekwencjonowania trzeciej generacji (3GS), polegającą na elektroforetycznym transporcie kwasów nukleinowych przez kanały złożone z białek (nanoporów) oraz identyfikacji ich sekwencji na podstawie zmian mierzonego sygnału elektrycznego. Kwas deoksyrybonukleinowy (DNA) jest zbudowany z czterech rodzajów nukleotydów, różniących się zasadami azotowymi: adeniny (A) i tyminy (T) oraz cytozyny (C) i guaniny (G). Każda z nich ma inną masę i strukturę chemiczną, a w związku z tym każda z zasad generuje unikatowy sygnał prądowy, co umożliwia jednoznaczne przypisanie sekwencji DNA.

Technologia Oxford Nanopore charakteryzuje się wysoką jakością uzyskiwanych danych oraz prostotą obsługi, a także stosunkowo krótkim czasem sekwencjonowania. Umożliwiła również wydłużenie odczytów sekwencyjnych oraz bezpośrednie sekwencjonowanie cząsteczek DNA i RNA w ich naturalnym stanie. W ciągu kilku lat od wprowadzenia pierwszego sekwenatora nanoporowego technologia Oxford Nanopore stała się jedną z przodujących metod sekwencjonowania [57]. Ze względu na swoją prostotę, dostępność oraz przystępną cenę sekwencjonowanie nanoporowe jest coraz częściej stosowane w badaniach epidemiologicznych do szybkiej identyfikacji i monitorowania rozprzestrzeniania się chorobotwórczych bakterii. Technologia Oxford Nanopore może się także przyczynić do rozwoju Internetu Rzeczy Żywych (Internet of Living Things, IoLT), ponieważ sekwencjonowanie DNA w czasie rzeczywistym pozwoliłoby na monitorowanie mikroorganizmów w glebie lub w wodzie, badanie różnorodności genetycznej

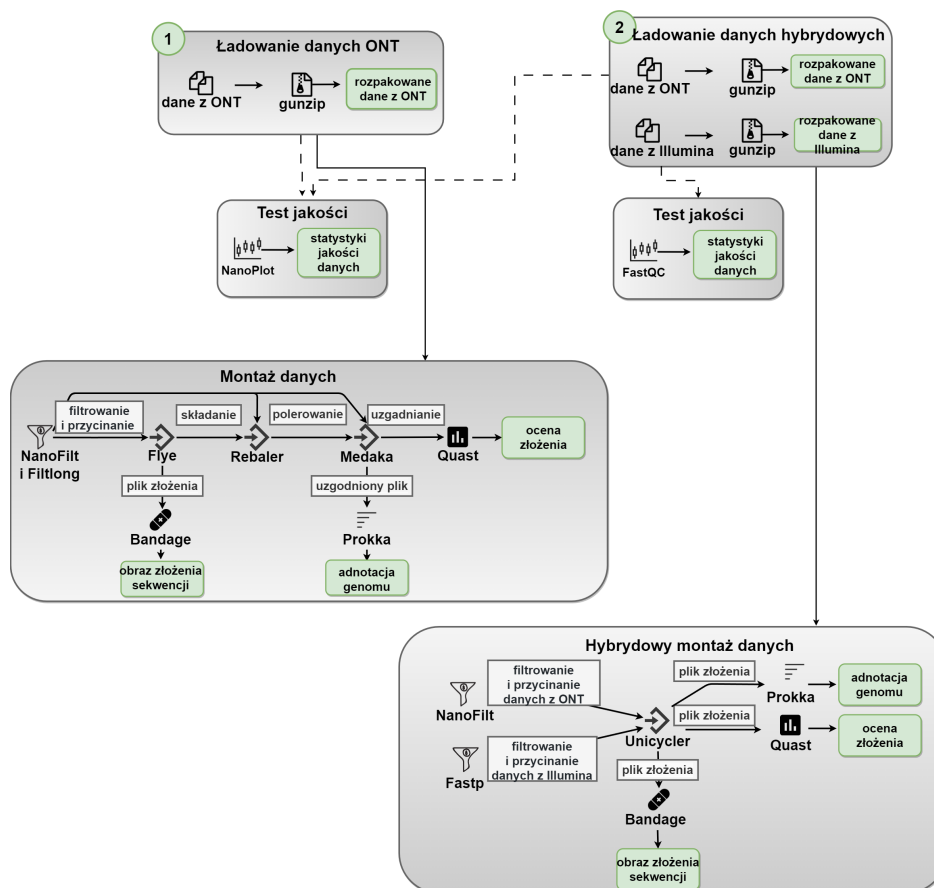
populacji zwierząt lub roślin, a także diagnozowanie chorób zakaźnych w terenie.

Analiza sekwencji DNA bakterii jest istotnym elementem diagnostyki medycznej, ponieważ umożliwia automatyczną klasyfikację bakterii, identyfikację chorobotwórczych gatunków, określanie wrażliwości na antybiotyki, analizę mikrobiomu jelitowego, przewidywanie ryzyka wystąpienia chorób związanych z określonymi gatunkami bakterii czy wczesne wykrywanie infekcji bakteryjnych na podstawie analizy sekwencji genomów [58]. Metody uczenia maszynowego pozwalają na przewidywanie jakości sekwencji nukleotydów, ich składanie, filtrowanie oraz korekcję błędów w sekwencjach. Aplikacja-serwer NanoForms opisana w artykule [A-3] pozwala na automatyzację procesów oceny jakości i składania genomów prokariotycznych. Umożliwia analizę danych genomów osobom bez specjalistycznej wiedzy bioinformatycznej lub informatycznej. Wykorzystanie metod uczenia maszynowego do wczesnego wykrywania infekcji bakteryjnych na podstawie analizy sekwencji genomów może znacznie usprawnić skuteczne leczenie i zapobieganie rozwojowi poważnych chorób oraz epidemii [59, 60].

Niemniej jednak, po zakończeniu standardowego eksperymentu za pomocą ONT badacze stają przed wyzwaniem przetworzenia ogromnej ilości surowych danych. Na rynku istnieje wiele narzędzi bioinformatycznych, które służą do klasyfikacji taksonomicznej, jednak ich zastosowanie może być trudne dla niedoświadczonych użytkowników z powodu złożoności narzędzi, liczby funkcjonalności, ustawień parametrów oraz problemów z instalacją i aktualizacją. Mimo dostępności badań porównawczych, które dostarczają zaleceń dotyczących najlepszych narzędzi i ich uruchamiania, korzystanie z tych narzędzi wciąż może wymagać doświadczenia i zaawansowanej wiedzy bioinformatycznej.

W celu rozwiązania wymienionych problemów opracowano aplikację NanoForms. Została ona zaimplementowana przy użyciu języka Python, frameworka Django, systemu operacyjnego Linux/UNIX/BSD, Workflow Description Language (WDL). Wykorzystano także Cromwell, Crontab, Docker i BioContainers oraz niestandardowy zestaw narzędzi bioinformatycznych, takich jak Bandage, Fastp, FastQC, Filtlong, Flye, Kraken 2, Kraken Tools, Krona, Medaka, Nanofilt, NanoPlot, Prokka, Rebaler, Unicycler czy QUAST. Niektóre z narzędzi, m.in. Kraken 2 czy Medaka, wykorzystują metody sztucznej inteligencji w swoim działaniu. Kraken 2 stosuje metody uczenia maszynowego do klasyfikacji genomów lub metagenomów do określonych taksonów przy użyciu sekwencji referencyjnych [61, 62]. Medaka to narzędzie do tworzenia konsensusowych

sekwencji i wariantów wywołań z danych uzyskanych z sekwencjonowania nanoporowego. Do realizacji tego zadania Medaka wykorzystuje sieci neuronowe, które operują na zestawie pojedynczych odczytów sekwencjonowania [63]. Diagram przepływu danych w aplikacji NanoForms został przedstawiony na rysunku 2.4.



Rysunek 2.4: Diagram przepływu danych w aplikacji NanoForms

Aplikacja NanoForms jest dostępna bezpłatnie do celów akademickich. Kod źródłowy aplikacji jest udostępniony publicznie (na licencji GPLv3) do użytku niekomercyjnego i jest dostępny pod adresem internetowym <https://github.com/czmilanna/NanoForms>. Pozwala na prostą instalację lokalnej wersji NanoForms. Działanie aplikacji ograniczono do analizy małych genomów prokariotycznych (sekwencje o wielkości do 15 Mb i do 15 GB rozmiaru pliku), ponieważ w przypadku ludzkiego genomu typowy surowy zestaw danych z sekwencjonowania Oxford Nanopore przekracza 1 TB. Powoduje to utrudnienia z przesyłaniem surowych danych przez Internet, nawet przy dużej przepustowości, a także wymaga znacznych nakładów finansowych na zasoby obliczeniowe.

Użytkownicy nieposiadający konta w systemie mają możliwość dostępu do przy-

kładowych zbiorów danych oraz wyników testów jakości i składania sekwencji w celu zapoznania się z funkcjonalnościami oferowanymi przez aplikację. Aby korzystać z głównych funkcjonalności aplikacji, wymagana jest rejestracja i zalogowanie się w systemie, co jest związane z zapewnieniem bezpieczeństwa systemu.

W celu założenia konta w systemie należy wypełnić formularz rejestracyjny, a następnie potwierdzić rejestrację za pomocą wiadomości wysłanej na podany przy rejestracji adres e-mail. Po rejestracji użytkownik ma możliwość zalogowania się na swoje konto. Zalogowani użytkownicy mogą dodawać własne zbiory danych do późniejszej analizy. Posiadają także dostęp do kilku publicznych zestawów danych pochodzących z Europejskiego Archiwum Nukleotydów (ENA; <http://www.ebi.ac.uk/ena>). Użytkownicy mogą wykorzystywać te zbiory danych do przeprowadzania testów jakości lub składania sekwencji danych przy użyciu odpowiednich formularzy dostępnych w NanoForms.

Przeprowadzenie testu jakości w NanoForms pozwala użytkownikowi zdecydować, czy kontynuować analizę, czy też powrócić do laboratorium w celu poprawy jakości danych. Aplikacja oferuje także dwie główne strategie składania genomów bakteryjnych przy użyciu sekwencjonowania z długim odczytem:

- składanie de novo (de novo assembly) – polega na odtworzeniu badanej sekwencji przez sklepanie długich odczytów nakładających się na siebie bez konieczności stosowania genomu referencyjnego,
- składanie hybrydowe (hybrid assembly) – łączy odczyty długie z odczytami krótkimi, co ma na celu zapewnienie większej dokładności składania. Pierwszym krokiem jest wykorzystanie odczytów długich do zbudowania wstępnej wersji genomu, a następnie wykorzystanie odczytów krótkich do poprawienia błędów i uzupełnienia luk w sekwencji. Składanie hybrydowe jest stosowane szczególnie w przypadku organizmów o wysokiej złożoności genomowej.

W celu przeprowadzenia składania hybrydowego należy załadować dwa zestawy danych: jeden zawierający długie odczyty uzyskane przy użyciu sekwencjonowania Oxford Nanopore, a drugi – referencyjny – składający się z krótkich odczytów z sekwencjonatora Illumina. NanoForms oferuje również opcje wyboru niektórych parametrów dla zaawansowanych użytkowników. Proces składania genomów jest czasochłonną operacją, dlatego też podczas wykonywania analizy użytkownik ma podgląd procesu przetwarza-

nia zadania w czasie rzeczywistym. Po ukończeniu procesu analizy system automatycznie informuje użytkownika o zakończeniu operacji, przesyłając wiadomość na podany podczas rejestracji adres mailowy.

Końcowym wynikiem działania NanoForms jest złożony genom w formacie FASTA, a także raport zawierający pliki adnotacji Prokka i diagram z programu Bandage, pozwalający na prostą ocenę graficzną kompletności złożenia. Użytkownik ma możliwość pobrania zarówno samego pliku złożenia, jak i folderu zawierającego raporty z działania poszczególnych narzędzi wraz z wynikami cząstkowymi. Aplikacja zapewnia także zakładkę z listą najczęściej zadawanych pytań (frequently-asked questions, FAQ), na której można znaleźć liczne wskazówki dotyczące sposobu korzystania z serwera.

W artykule dokonano również szczegółowej analizy oraz porównania aplikacji i narzędzi podobnych do NanoForms, takich jak CGE [64], Enterobase [65], Galaxy Tools [66], NanoGalaxy [67], Patric [68], EPI2ME [69], NanoPipe [70] z przedstawieniem ich możliwości oraz wad i zalet. Wśród wymienionych usług NanoForms wyróżnia się prostotą w uruchamianiu analiz połączonych w strumień komend. Jest on dostępny bezpłatnie dla wszystkich naukowców, łączy szybkie składanie genomu prokariotycznego z intuicyjnym, interaktywnym interfejsem. Do uzyskania sekwencji próbki wystarczy posiadać średniej klasy laptop oraz urządzenie MinION firmy Oxford Nanopore. Żadne dodatkowe zasoby nie są potrzebne, a użytkownik może kontynuować analizy genomiczne z wykorzystaniem usługi NanoForms.

Aplikacja NanoForms zautomatyzowała proces oceny, składania i identyfikacji sekwencji genomowych uzyskanych za pomocą nowych metod sekwencjonowania z zastosowaniem narzędzi korzystających z metod uczenia maszynowego. Powstała ona podczas realizacji projektu *Technologia Oxford Nanopore: optymalizacja enzymów oraz analizy danych genomicznych pod kątem zastosowań komercyjnych*, realizowanego w ramach programu grantowego na prace B+R jednostek naukowych w ramach projektu *Podkarpackiego Centrum Innowacji*. Serwer wzbudził spore zainteresowanie różnych instytucji badawczych, uniwersytetów oraz placówek medycznych z całego świata, w tym z Japonii, Kolumbii, Niemiec, Tajwanu, Indii i Stanów Zjednoczonych, a także organizacji zajmujących się zdrowiem publicznym, genetyką i technologią biometryczną.

Wkład własny autorki niniejszej rozprawy doktorskiej w powstanie publikacji [A-3] obejmował współredakcję pracy, przygotowanie architektury aplikacji, implementację kluczowych funkcjonalności w aplikacji Nanoforms obejmujących ładowanie

danych, przygotowanie skryptów za pomocą Workflow Description Language (WDL) pozwalających na ocenę jakości danych z sekwencjonowania nanoporowego Oxford Nanopore oraz Illumina, a także umożliwiających składanie genomów bakteryjnych metodami de novo i hybrydową. Autorka była również odpowiedzialna za analizę danych genomowych uzyskanych podczas sekwencjonowania, a także przygotowanie widoków aplikacji oraz opracowanie i analizę wyników.

## 2.4. Zastosowanie programowania ekspresji genów do wydobywania metareguł z danych medycznych

W ostatnich latach rozwój nowych technologii oraz coraz większa ilość dostępnych danych medycznych przyczyniły się do poszukiwania nowych rozwiązań w zakresie diagnostyki i sposobów leczenia pacjentów. W celu uzyskania bardziej precyzyjnych, spójnych i szybkich wyników badań coraz częściej korzysta się z narzędzi opartych na sztucznej inteligencji. Dzięki tym rozwiązaniom lekarze mogą szybciej przewidzieć rozwój choroby, a także skuteczniej zapobiegać postępowi chorób przewlekłych [71, 72]. Wyjaśnialna sztuczna inteligencja (explainable AI, XAI) to podejście do projektowania i stosowania sztucznej inteligencji, którego celem jest ułatwienie lekarzom i pacjentom zrozumienia, w jaki sposób system wygenerował określone wyniki i prognozy. W tym celu często są stosowane algorytmy generujące reguły, ponieważ ich działanie jest proste do zrozumienia i interpretacji. Projektowanie klasyfikatorów, które są jednocześnie dokładne i pozwalają na zrozumienie mechanizmu klasyfikacji danych, jest bardzo ważnym problemem.

Implementacja algorytmu GPR w języku Python opisana w artykule [A-4] zapewnia otwarty dostęp do kodu źródłowego w celu umożliwienia użytkownikom korzystania z algorytmu bez żadnych komercyjnych narzędzi programistycznych. Algorytm GPR został zaproponowany i dokładnie zbadany w artykule [73]. Dalej zamieszczono jedynie jego krótki opis, zakładając najciekawszy przypadek, gdy oryginalny zbiór danych zawiera  $n$ -wymiarowe rzeczywiste wektory wejściowe (rekordy danych) ze współrzędnymi (cechami), na przykład  $y_k$  ( $k = 1, \dots, n$ ), należącymi do skończonych przedziałów. Wszystkie wektory wejściowe należy przekształcić w nowe punkty z hipersześcianu  $I^n = [0, 1]^n$ . Załóżmy, że rozmyty system oparty na regułach P1-TS modeluje zbiór danych i składa się z kilku rozmytych metareguł „jeżeli-to” [74, 75, 76]. Metareguła jest odpowiednikiem wielu pojedynczych reguł rozmytych „jeżeli-to”. Poprzednik dowolnej

pojedynczej reguły odnosi się do wszystkich zmiennych wejściowych  $y_1, \dots, y_n$ , natomiast poprzednik metareguły odnosi się do właściwego podzbioru zbioru zmiennych wejściowych  $\{y_1, \dots, y_n\}$ . Każda cecha  $y_k$  wykorzystywana w regule lub metaregule odnosi się do jednego z dwóch zbiorów rozmytych (zmiennych lingwistycznych). Funkcja przynależności pierwszego zbioru rozmytego została zdefiniowana jako  $P_k(y_k) = y_k$ , natomiast drugiego jako  $N_k(y_k) = 1 - P_k(y_k)$ , dla  $k = 1, \dots, n$ .

Jeżeli wszystkie następniki metareguł rozpatrywanego systemu P1-TS pochodzą ze zbioru  $\{0, 1\}$ , to system regułowy reprezentuje model wyrażony w logice wielowartościowej. Na przykład, jeśli  $y_k \leq \theta$ , gdzie  $\theta$  jest wartością progową (zwykle  $\theta = 0,5$ ), to  $y_k$  jest interpretowane jako „stwierdzenie prawie fałszywe” (*Low* lub *L*); w przeciwnym razie etykieta tej zmiennej jest interpretowana jako „stwierdzenie prawie prawdziwe” (*High* lub *H*). Zgodnie z twierdzeniem przedstawionym w artykule [73], dla wejść  $y_k \in [0, 1]$  układu P1-TS, po przekształceniu wszystkich zmiennych wejściowych ze zbioru  $\{y_1, \dots, y_n\}$  do zestawu nowych wejść:  $\{x_1, \dots, x_{2n}\}$ , takich że  $x_{2k-1} = y_k$  i  $x_{2k} = 1 - y_k$ , dla  $k = 1, \dots, n$ , nierozmyte wyjście  $S$  tego systemu można wyrazić za pomocą sumy iloczynów zmiennych „ $x_{(\cdot)}$ ” w następujący sposób:

$$S = \sum_{r=1}^M \prod_{k \in K_r} x_k \quad (2.1)$$

gdzie  $\prod_{k \in K_r} x_k$  jest iloczynem zmiennych reprezentujących cechy ciągłe dla rekordów danych, które odpowiadają etykietce klasy „1”, natomiast  $K_1, \dots, K_M \subset \{1, \dots, 2n\}$  są podzbiórmi zbioru indeksów i zazwyczaj  $1 \leq M \leq 2^{2n} - 1$ . Dodatkowo, dowolne wyrażenie algebraiczne w postaci zależności (2.1) może być interpretowane jako system metareguł, który definiuje system oparty na regułach P1-TS. Głównym problemem, który należy rozwiązać, jest znalezienie wyrażenia w postaci równania (2.1) dla danego zbioru danych. Należy zauważyć, że liczba możliwych rozwiązań tego problemu w postaci równania (2.1) jest ogromna, dlatego do jego rozwiązania zaproponowano użycie algorytmu GEP [77]. Ponadto, zbiór danych może zawierać rekordy z atrybutami kategorycznymi (etykietami), nie omówiono jednak tego (prostsze) przypadku, gdyż szczegóły znajdują się w artykule [73]. Minusem pierwotnego podejścia ([73]) była implementacja algorytmu przy użyciu komercyjnych narzędzi GeneXproTools i GeneXproServer wchodzących w skład oprogramowania do modelowania predykcyjnego oferowanego przez firmę Gepsoft Limited. Uniemożliwiało to otwarty dostęp do algorytmu. Dodatkowo, w odniesieniu do hiperparametrów GEP, wciąż istniał jeszcze potencjał

do uzyskania większej kontroli nad rozmiarem i złożonością wynikowych metareguł.

Implementacja klasyfikatora GPR w języku Python opisana w artykule [A-4] stanowi alternatywę dla narzędzi komercyjnych oraz zawiera następujące ulepszenia w stosunku do oryginalnego algorytmu:

- zautomatyzowano proces generowania metareguł językowych „jeżeli-to” – zadaniem użytkownika jest tylko udostępnienie rekordów danych zawierających liczby rzeczywiste z przedziału  $[0,1]$  oraz etykiet ze zbioru  $0,1$ ,
- do tej pory algorytm korzystał z dwóch rozmytych pojęć, takich jak „niski” ((*Low* lub *L*) i „wysoki” ((*High* lub *H*), występujących w poprzednikach reguł. W implementacji dodano jeszcze poprzedniki „średni” ((*Medium* lub *M*) oraz określenie intensywności „bardzo” (*very*), które są logicznie interpretowalne,
- dla każdej reguły jest obliczane jej wsparcie (współczynnik ufności).

W artykule zostały dokładnie opisane mechanizmy działania i implementacji algorytmu GPR. Przedstawiono również jego główne funkcjonalności oraz omówiono fragmenty kodu i przykładowe wyniki w celu zademonstrowania możliwości ustawienia algorytmu, a także sposobu zwracania przez niego wyników. Kod algorytmu jest dostępny pod adresem <https://github.com/czmilanna/gpr-algorithm>, natomiast dokumentacja algorytmu jest dostępna pod adresem <https://gpr-algorithm.readthedocs.io/en/latest/>. Do implementacji algorytmu użyto bibliotek Deap i Geppy dedykowanych do obliczeń ewolucyjnych, jak również pakietu numerycznego NumPy [78, 79, 80]. Podstawowa funkcjonalność algorytmu jest realizowana przez klasę GPR. Algorytm posiada metody `fit()` i `predict()`, które są zgodne z interfejsem biblioteki Scikit-learn stanowiącej obecnie jedną z najpopularniejszych i najbardziej przystępnych darmowych bibliotek uczenia maszynowego [81]. Służą one odpowiednio do uczenia i klasyfikacji przy użyciu wcześniej wyszkolonego modelu. Dzięki temu, że wyniki są zwracane w sposób analogiczny do użytego w bibliotece Scikit-learn, możliwe jest skorzystanie z funkcji modułu `sklearn.metrics` do oceny jakości klasyfikacji.

W celu przedstawienia sposobu generowania metareguł przez algorytm GPR przeprowadzono badanie zależności pomiędzy tygodniową liczbą kroków a częstotliwością występowania cukrzycy typu 1. W badaniu użyto tego samego zbioru danych, który opisano w artykule [A-1]. Parametr `max_n_of_rules` ustawiono na 1, aby algorytm



wygenerował tylko jedną główną regułę, tak jak zostało to pokazane na przykładzie zamieszczonym w Listingu 1.

Listing 1: Przykładowe użycie algorytmu GPR na zbiorze danych dotyczących występowania cukrzycy typu 1

```
1 import random
2 from pathlib import Path
3 import pandas as pd
4 from sklearn.metrics import accuracy_score
5 from sklearn.preprocessing import MinMaxScaler
6 from gpr_algorithm import GPR
7
8 random.seed(0)
9 df = pd.read_csv(
10     Path(__file__).parent.joinpath('data').joinpath('type1diabetes.csv')
11 )
12
13 target_names = ['sick', 'healthy']
14 feature_names = [
15     'age', 'weight', 'height', 'step_count',
16     'sedentary', 'light', 'moderate', 'vigorous'
17 ]
18
19 labels = df['healthy'].values
20 attributes = df[feature_names].values
21 attributes_normalized = MinMaxScaler().fit_transform(attributes)
22
23 gpr = GPR(
24     target_names=target_names,
25     feature_names=feature_names,
26     max_n_of_rules=1,
27     eval_fun=accuracy_score,
28     verbose=False
29 )
30 gpr.fit(attributes_normalized, labels)
31 predicted_labels = gpr.predict(attributes_normalized)
32 for rule in gpr.rules:
33     print(rule)
```

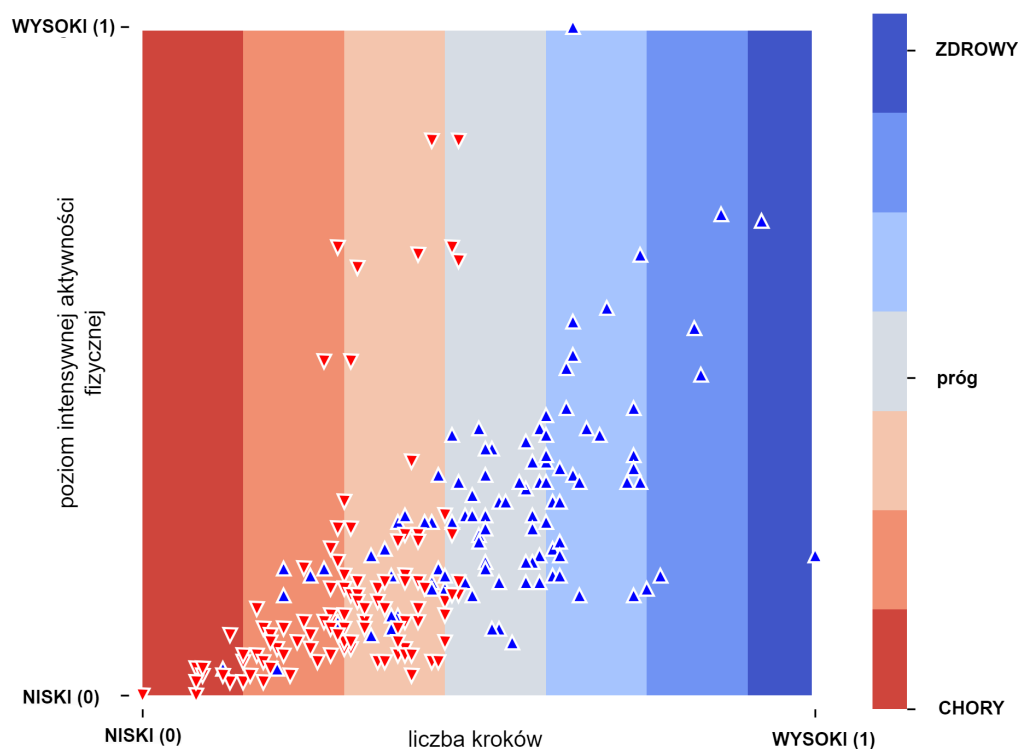
Otrzymano następującą regułę wyjściową:

**IF step\_count is High THEN healthy | Support: 0.5288**

**ELSE sick**

Ponadto, za pomocą drzewa decyzyjnego występowanie choroby zostało prawidłowo sklasyfikowane w 65% przypadków, natomiast algorytm GPR poprawnie sklasyfikował występowanie cukrzycy w 80,87%. Należy jednak zauważyć, że wymaga on

użycia danych znormalizowanych do przedziału  $[0,1]$ .



Rysunek 2.5: Wizualizacja granic systemu decyzyjnego algorytmu GPR wykonana dla danych dotyczących występowania cukrzycy typu 1 (przyjęto próg równy 0,5)

Na rysunku 2.5 zostały przedstawione granice decyzyjne klasyfikatora GPR w przestrzeni cech, składającej się z liczby kroków i poziomu intensywnej aktywności fizycznej, w odniesieniu do zbioru danych dotyczących występowania cukrzycy typu 1 i otrzymanych reguł rozmytych. Wartości liczbowe zmiennych zostały znormalizowane w zakresie  $[0,1]$ .

Wyniki badań opisane w artykule [73] wykazują, że algorytm GPR cechuje się wysoką jakością klasyfikacji zarówno pod względem pola pod krzywą ROC, jak i dokładności. Czyni go to jednym z najlepszych interpretowalnych klasyfikatorów. W związku z tym proponowana implementacja algorytmu w języku Python powinna zainteresować naukowców zajmujących się logiką rozmytą i jej zastosowaniami. GPR generuje łatwe w interpretacji, rozmyte metareguly „jeżeli-to”, które wpisują się w trend intensywnie rozwijanej obecnie wyjaśnialnej sztucznej inteligencji. W odróżnieniu od poprzedniej implementacji wymaga jedynie minimalnej ingerencji ze strony użytkownika – wystarczy tylko znormalizować dane w zakresie  $[0,1]$ .

Wkład własny autorki w powstanie publikacji [A-4], będącej częścią niniejszej rozprawy doktorskiej, polegał na współautorstwie koncepcji artykułu i metodologii, wykonaniu implementacji algorytmu GPR w języku programowania Python, przeprowadzeniu eksperymentów, interpretacji otrzymanych reguł klasyfikatora, walidacji otrzymanych wyników oraz współredakcji pracy.

## **2.5. Porównanie rozmytych klasyfikatorów opartych na regułach i metaregułach**

Artykuł [A-5] stanowiący część rozprawy jest kontynuacją badań dotyczących wyjaśnialnej sztucznej inteligencji i zawiera porównanie 16 wybranych rozmytych algorytmów opartych na regułach, które zostały zastosowane do klasyfikacji danych medycznych, w tym również danych rzeczywistych. Działanie klasyfikatorów oceniono za pomocą metryk wydajnościowych. Przeprowadzono również wiele analiz statystycznych i porównawczych dotyczących złożoności oraz czytelności otrzymanych reguł generowanych przez każdy algorytm, a także ocenę użyteczności każdego algorytmu we wspomaganie decyzji klinicznych.

W dzisiejszych czasach sztuczna inteligencja odgrywa niewątpliwie coraz większą rolę w wielu dziedzinach życia. Aby jednak zaufać decyzjom opartym na sztucznej inteligencji, muszą być one interpretowalne i zrozumiałe dla ludzi [82]. Wyjaśnialna sztuczna inteligencja umożliwia zgłębienie czynników, które wpłynęły na podjęte decyzje i ocenę ich słuszności.

Do opracowywania systemów wspomaganie decyzji medycznych (medical decision support systems, MDSS) często stosuje się rozmyte systemy ekspertowe oparte na regułach. Systemy wspomaganie decyzji medycznych wykorzystują wiedzę z danych opisujących historię choroby i stan pacjenta w celu uzyskania porady klinicznej zakodowanej w postaci zestawu reguł decyzyjnych [83]. Zastosowanie logiki rozmytej pozwala na reprezentowanie danych pacjentów oraz rozumowania klinicznego stosowanego przez lekarzy do oceny stanu ich zdrowia, a także ułatwia zrozumienie przez ekspertów decyzji podjętej przez system w odróżnieniu od algorytmów będących czarnymi skrzynkami (black box) [84].

Istnieje wiele algorytmów regułowych, które można wykorzystać w zastosowaniach medycznych do analizy danych. Należy jednak podkreślić, że problem interpretowalności systemów opartych na regułach rozmytych lub nierozmytych stanowi od

wielu lat kluczowe zagadnienie podejmowane przez wielu autorów. Znane z literatury klasyczne rozmyte systemy regułowe Takagi-Sugeno, Mamdaniego czy Larsena często nie są rozpatrywane z powodu przekleństwa wymiarowości, gdyż wraz ze wzrostem liczby wymiarów w systemie liczba obiektów wymaganych do wiarygodnego oszacowania parametrów rośnie wykładniczo. W porównaniu wzięto zatem pod uwagę także algorytm GPR, oparty na metaregułach, które pozwalają na osiągnięcie kompromisu między interpretowalnością a wydajnością klasyfikacji. Ponadto, wybór odpowiednich miar interpretowalności wciąż pozostaje problemem otwartym [85].

W artykule [A-5] porównano 16 następujących algorytmów opartych na regułach rozmytych: One Rule (1R-C), C4.5 (C4.5-C), C4.5Rules (C45Rules-C), C4.5Rules Simulated Annealing Version (C45RulesSA-C), Hybrid Decision Tree-Genetic Algorithm (DT\_GA-C), Oblique Decision Tree with Evolutionary Learning (DT\_Oblique-C), Exemplar-Aided Constructor of Hyperrectangles (EACH-C), Classifier Based on Fuzzy Logic and Gene Expression Programming (GPR), Hierarchical Decision Rules (Hider-C), New Structural Learning Algorithm in a Vague Environment (NSLV-C), Organizational Co-Evolutionary Algorithm for Classification (OCEC-C) oraz Ordered Incremental Genetic Algorithm (OIGA-C). Implementacje algorytmów, oprócz GPR, pochodzą z oprogramowania KEEL [86]. Implementacja algorytmu GPR została natomiast opisana w artykule [A-4]. Wybrane algorytmy nie tylko mają zdolność do klasyfikacji, ale również generują reguły, które są w pewnym stopniu interpretowalne. Dzięki temu możliwe jest uzyskanie wiedzy na temat danych, w tym zrozumienie motywacji danego klasyfikatora podczas podejmowania konkretnej decyzji.

Wydajność procesu klasyfikacji oceniono pod względem dokładności, precyzji, czułości, specyficzności, pola pod krzywą ROC, współczynnika korelacji Matthews'a (Matthew's correlation coefficient, MCC) oraz zaproponowanej na potrzeby badania metryki ważonej (weighted metric, WM), biorącej pod uwagę wszystkie obliczone wcześniej metryki wydajnościowe. W tym celu użyto 12 zbiorów danych medycznych, które zostały pobrane z repozytorium zbiorów danych KEEL [86] lub pozyskane w trakcie prowadzenia odrębnych badań naukowych [87, 37]. Zbiory zawierały różnorodne dane dotyczące zapalenia wyrostka robaczkowego (appendicitis), raka piersi (breast), 5-letniego przeżycia kobiet po operacji raka piersi (haberman), chorób serca (heart), zapalenia wątroby (hepatitis), przewidywania stopnia ciężkości guza piersi (mammographic), chorób serca u mężczyzn z Afryki Południowej (saheart), obrazów z tomografii

emisyjnej pojedynczego fotonu (spectfheart), diagnostyki raka piersi (wdbc), złośliwości wykrytego guza piersi (wisconsin), a także powikłań okołoperacyjnych po radykalnej histerektomii u pacjentek z rakiem szyjki macicy [87] oraz aktywności fizycznej u dzieci i młodzieży z cukrzycą typu 1 [37].

Tabela 2.2: Wyniki porównania algorytmów opartych na regułach rozmytych z zastosowaniem metryk wydajnościowych

Nr	Algorytm	MCC	ACC	AUC	Spe	Pre	Sen	WM
1	GPR	<b>0.459±0.342</b>	<b>0.807±0.281</b>	0.720±0.171	<b>0.792±0.125</b>	0.772±0.167	<b>0.792±0.125</b>	<b>0.753±0.145</b>
2	OIGA-C	<b>0.457±0.337</b>	<b>0.860±0.253</b>	0.714±0.172	<b>0.793±0.114</b>	<b>0.782±0.152</b>	<b>0.793±0.114</b>	<b>0.755±0.138</b>
3	Ripper-C	<b>0.452±0.319</b>	0.676±0.243	<b>0.730±0.162</b>	0.735±0.158	<b>0.780±0.139</b>	0.735±0.158	0.718±0.164
4	C45RulesSA-C	0.449±0.343	0.752±0.255	<b>0.727±0.172</b>	0.769±0.140	0.776±0.147	0.769±0.140	0.740±0.157
5	OCEC-C	0.447±0.323	0.753±0.221	<b>0.726±0.164</b>	0.753±0.145	0.771±0.145	0.753±0.145	0.730±0.156
6	NSLV-C	0.446±0.338	0.791±0.298	0.716±0.171	<b>0.795±0.122</b>	0.771±0.148	<b>0.795±0.122</b>	<b>0.752±0.141</b>
7	C45Rules-C	0.446±0.340	0.738±0.273	0.724±0.173	0.768±0.142	<b>0.777±0.141</b>	0.768±0.142	0.737±0.159
8	DT GA-C	0.442±0.329	0.799±0.267	0.712±0.163	0.784±0.116	0.775±0.138	0.784±0.116	0.746±0.135
9	SLAVE2-C	0.438±0.338	0.792±0.296	0.712±0.170	0.786±0.123	0.769±0.148	0.786±0.123	0.746±0.144
10	C45-C	0.438±0.343	0.785±0.264	0.710±0.171	0.782±0.128	0.772±0.146	0.782±0.128	0.743±0.147
11	Hider-C	0.414±0.336	0.797±0.274	0.693±0.167	0.767±0.138	0.763±0.144	0.767±0.138	0.728±0.150
12	DT Oblique-C	0.402±0.346	0.741±0.222	0.703±0.173	0.745±0.146	0.754±0.149	0.745±0.146	0.715±0.160
13	SLAVEv0-C	0.394±0.374	0.761±0.315	0.691±0.182	0.772±0.137	0.749±0.161	0.772±0.137	0.727±0.158
14	PGIRLA-C	0.327±0.337	<b>0.819±0.269</b>	0.655±0.165	0.716±0.193	0.668±0.239	0.716±0.193	0.681±0.172
15	EACH-C	0.264±0.340	0.621±0.417	0.626±0.165	0.662±0.185	0.675±0.238	0.662±0.185	0.630±0.180
16	1R-C	0.228±0.331	0.652±0.378	0.610±0.160	0.703±0.162	0.636±0.211	0.703±0.162	0.645±0.160

Wyniki uzyskane dla poszczególnych metryk wydajnościowych zostały obliczone na podstawie 10-krotnej walidacji krzyżowej na każdym ze zbiorów danych i zamieszczone w tabeli 2.2. W przeprowadzonych badaniach algorytm GPR uzyskał najlepszy współczynnik korelacji Matthews (0,459 ± 0,342), natomiast najniższy współczynnik został osiągnięty przez 1R-C (0,228 ± 0,331). Biorąc pod uwagę dokładność, stwierdzono, że najlepsze wyniki uzyskały algorytmy OIGA-C (0,860 ± 0,253), PGIRLA-C (0,819 ± 0,269) oraz GPR (0,807 ± 0,281). Według kryterium pola pod krzywą ROC najlepszy wynik został osiągnięty przez Ripper-C (0,730 ± 0,162), a następnie przez C45RulesSA-C oraz OCEC-C. Najlepszą specyficzność osiągnęły kolejno algorytmy: NSLV-C (0,795 ± 0,122), OIGA-C (0,793 ± 0,114) oraz GPR (0,792 ± 0,125). Najwyższą precyzję uzyskał algorytm OIGA-C (0,782 ± 0,152). Ze względu na czułość najlepsze wyniki uzyskały NSLV-C (0,795 ± 0,122), OIGA-C (0,793 ± 0,114) oraz GPR (0,792 ± 0,125), najgorsze natomiast – EACH-C (0,662 ± 0,185). Jeśli chodzi o metrykę ważoną (WM), to najlepsze wyniki uzyskały OIGA-C (0,755 ± 0,138), GPR (0,753 ± 0,145) oraz NSLV-C (0,752 ± 0,141), natomiast najslabsze – EACH-C (0,630

$\pm 0,180$ ), 1R-C ( $0,645 \pm 0,160$ ) oraz PGIRLA-C ( $0,681 \pm 0,172$ ).

Następnie zbadano rozkłady wartości współczynnika korelacji Matthews, dokładności i pola pod krzywą ROC dla każdego algorytmu we wszystkich zbiorach danych. Klasyfikatory porównano także pod względem średniej długości reguł generowanych w zbiorze danych, średniej liczby reguł w zbiorze, średniej liczby atrybutów na regułę w zbiorze oraz średniej liczby unikatowych atrybutów na regułę w zbiorze. Analiza wyników wykazała, że algorytm 1R-C cechuje się zwięzłymi i prostymi regułami klasyfikacji. Wygenerował on reguły decyzyjne najkrótszej długości (średnio 106,54 znaków), o niewielkiej liczbie reguł oraz niewielkiej liczbie atrybutów w pojedynczej regule. Osiąga on jednak niskie wyniki wskaźników jakości klasyfikacji, takich jak dokładność czy współczynnik korelacji Matthews. Algorytm GPR osiągnął z kolei znacznie lepsze wyniki klasyfikacji, a przy tym wygenerowane przez niego reguły są relatywnie krótkie i zwięzłe (średnia długość reguły wyniosła 156,23 znaków, algorytm generuje średnio 4 reguły, w pojedynczej regule wykorzystuje średnio 6,69 atrybutów, z których średnio 5,31 stanowią atrybuty unikatowe). Reguły te są znacznie krótsze i prostsze niż te wygenerowane np. przez algorytm OIGA-C, który uzyskał bardzo dobre wyniki klasyfikacji, ale generuje długie i trudne w interpretacji reguły (średnia długość reguły wynosi 20 958,08 znaków, średnia liczba reguł – 30,00, średnia liczba atrybutów – 399,23, natomiast średnia liczba unikatowych atrybutów – 15,85). Przeprowadzone badania sugerują, że algorytm DT\_Oblique generuje najbardziej skomplikowane reguły, co odzwierciedla się w średniej długości reguły wynoszącej 32 457,38 znaków.

Tabela 2.3: Przykłady lingwistycznych reguł rozmytych „jeżeli-to” wygenerowanych na zbiorze danych dotyczących cukrzycy typu 1 [37]

Nr	Algorytm	Przykładowe reguły	Liczba reguł	Całkowita liczba zn.
1	1R-C	IF step_count = [13072.0 , 55333.0) THEN 0 IF step_count = [55333.0 , 58288.0) THEN 1 IF step_count = [58288.0 , 60294.0) THEN 0 IF step_count = [60294.0 , 114655.0] THEN 1	4	172
2	GPR	IF step_count is High THEN 1 IF vigorous is High AND sedentary is High THEN 1 ELSE 0	3	87

Tabela 2.3: (cd.) Przykłady lingwistycznych reguł rozmytych „jeżeli-to” wygenerowanych na zbiorze danych dotyczących cukrzycy typu 1 [37]

Nr	Algorytm	Przykładowe reguły	Liczba reguł	Całkowita liczba zn.
3	C45Rules-C	IF height>1.61 AND age>14.0 AND weight<=52.0 THEN 1 IF vigorous>128.75 AND vigorous<=319.5 AND age>8.0 AND moderate>214.916666666667 THEN 1 [...]	8	400
4	C45RulesSA-C	IF height>1.61 AND age>14.0 AND weight<=52.0 THEN 1 IF vigorous>128.75 AND vigorous<=319.5 AND age>8.0 AND moderate>214.916666666667 THEN 1 [...]	8	400
5	EACH-C	IF age in [6.0 , 18.0] AND weight in [19.3 , 98.8] AND height in [1.15 , 1.88] AND step_count in [13072.0 , 60837.0] AND sedentary in [1343.166666666667 , 7813.333333333333] AND [...]	2	603
6	NSLV-C	IF step_count = { VeryLow Low} THEN 0 IF step_count = { High VeryHigh} THEN 1 IF age = { Low High VeryHigh} AND moderate = { Low VeryHigh} THEN 1	3	145
7	Ripper-C	IF step_count<=60837.0 AND height<=1.58 THEN 0 IF step_count<=60837.0 AND mode- rate<=119.0 THEN 0 [...]	9	370
8	C45-C	IF step_count <= 60837.000000 AND vigorous <= 128.750000 AND weight <= 80.500000 THEN 0 [...]	12	1828
9	DT_GA-C	IF step_count <= 60837.0 AND vigorous <= 128.75 AND weight <= 80.5 THEN 0 IF step_count <= 60837.0 AND vigorous <= 128.75 AND weight >80.5 THEN 1 [...]	19	2856
10	SLAVE2-C	IF age = { VeryLow Medium} AND weight = { Medium} AND height = { High VeryHigh} AND step_count = { VeryLow Low} AND sedentary = { Medium} AND light = { Low} AND moderate = { Low} AND vigorous = { VeryLow Medium} THEN 0 [...]	8	2098

Tabela 2.3: (cd.) Przykłady lingwistycznych reguł rozmytych „jeżeli-to” wygenerowanych na zbiorze danych dotyczących cukrzycy typu 1 [37]

Nr	Algorytm	Przykładowe reguły	Liczba reguł	Całkowita liczba zn.
11	SLAVEv0-C	IF step_count = { VeryLow Low} THEN 0 IF age = { VeryLow Low Medium VeryHigh} AND height = { VeryLow Low Medium VeryHigh} AND step_count = { Medium} AND sedentary = { Medium} AND light = { Low Medium High VeryHigh} AND [...]	8	2814
12	PGIRLA-C	IF sedentary = [3801.8675692824663, 5006.615988626676] AND light = [1162.1170959360238, 2362.4439084883884] AND moderate = [414.0390532025578, 474.55751714327096] AND vigorous = [339.7803046746366, 521.320375724421] THEN 0 [...]	19	4340
13	Hider-C	IF age = [7.5, 17.5) AND weight = [29.15, 65.7) AND step_count = [_, 55096.5) AND sedentary = [2270.0833333333335, 4964.9166666666664) AND light = [356.875, 1330.8333333333335) AND moderate = [124.3333333333335, 425.1666666666665) AND vigorous = [_, 497.6666666666665) THEN 0 [...]	14	3595
14	OCEC-C	IF step_count = 3 THEN 1 IF age = 2 AND sedentary = 1 THEN 1 IF sex = 0 AND vigorous = 1 THEN 1 IF sex = 0 AND step_count = 1 AND light = 1 THEN 0 IF height = 2 AND step_count = 1 THEN 0 [...]	62	6763
15	OIGA-C	IF 1.6699878586619132 <sex <1.1982191470913168 AND 9.4429624945491 <age <16.56761035848586 AND 67.72250192298611 <weight <85.23233850170679 AND 1.859257826523217 <height <1.7277770427143613 AND [...]	30	14312
16	DT_Oblique-C	IF -1.0*step_count + 60837.0 >= 0 AND -1.0*vigorous + 128.75 >= 0 AND -1.0*weight + 80.5 >= 0 AND -1.0*height + 1.87 >= 0 AND 168.486174002403*sex + -178.36864022034422*age + -1.0*weight + -36.57868193382831*light + 185.88945474147084*vigorous + 18.795399605016087 >= 0 THEN 1 [...]	30	8625



W tabeli 2.3 przedstawiono przykłady lingwistycznych reguł rozmytych „jeżeli-to” wygenerowanych przez rozmyte klasyfikatory oparte na regułach na zbiorze danych dotyczących cukrzycy typu 1 (opisanym szczegółowo w artykule [A-1]). Reguły dłuższe niż 300 znaków zostały odpowiednio skrócone do limitu znaków bądź jednej reguły. W tabeli 2.3 zawarto także informacje o całkowitej liczbie wygenerowanych reguł i całkowitej liczbie znaków, aby mieć lepszy pogląd na możliwość ich interpretacji.

W celu statystycznego porównania wyników osiągniętych przez GPR z wynikami innych algorytmów opartych na regułach rozmytych przeprowadzono test rang znakowanych Wilcoxon. W badaniach statystycznych uwzględniono dokładność, pole pod krzywą ROC i współczynnik korelacji Matthews. Biorąc pod uwagę uzyskane wyniki, należy stwierdzić, że GPR można z powodzeniem zastosować do generowania reguł z danych medycznych. Trzeba jednak wziąć pod uwagę ograniczenia dokonanych badań, między innymi brak przeprowadzonego testu zużycia zasobów sprzętowych komputera i pomiaru czasu działania, użycie domyślnych wartości hiperparametrów, a także przetestowanie wydajności algorytmów wyłącznie na medycznych zbiorach danych zawierających stosunkowo niewielką liczbą rekordów. Kod potrzebny do obliczeń oraz wyniki wszystkich wykonanych badań są dostępne pod adresem <https://github.com/czmilanna/rules>. Praca [A-5] jest próbą wykazania na wielu przykładach, że wykorzystanie rozmytych metareguł w zastosowaniach medycznych stanowi ważny wkład w rozwiązanie wspomnianego już problemu interpretowalności systemów opartych na regułach rozmytych lub nierozmytych, a także pomoc w doborze odpowiednich miar interpretowalności.

Artykuł [A-5] stanowi autorskie podejście do rozwiązania problemu wyboru optymalnego rozmytego klasyfikatora opartego na regułach bądź metaregułach w zastosowaniach medycznych. Udział własny autorki polegał na zaproponowaniu koncepcji artykułu, opracowaniu metodologii, implementacji metod i przeprowadzeniu eksperymentów obliczeniowych, opracowaniu, analizie i walidacji otrzymanych wyników oraz redakcji pracy.



### 3. Podsumowanie i wnioski

Niniejsza praca koncentruje się na wykorzystaniu metod sztucznej inteligencji w usprawnieniu procesu diagnostyki medycznej. Zaprezentowane badania wykazują skuteczność metod sztucznej inteligencji w różnych aspektach diagnostycznych. Postawiona hipoteza, że *możliwe jest wykorzystanie różnych metod sztucznej inteligencji do analizy danych medycznych i automatyzacji wybranych procesów diagnostycznych, pozwalające na uzyskanie interpretowalnych wyników z dokładnością i efektywnością nie gorszą niż innych istniejących metod znanych z literatury* została uprawdopodobniona przez realizację następujących zadań:

#### **1. Zastosowanie metod sztucznej inteligencji do klasyfikacji cukrzycy typu 1 na podstawie danych uzyskanych za pomocą nieinwazyjnych pomiarów aktywności fizycznej**

Zadanie zostało zrealizowane przez analizę i przygotowanie danych dotyczących wieku, płci, wagi, wzrostu oraz współczynników aktywności fizycznej zmierzonych w sposób nieinwazyjny przy użyciu akcelerometru, a następnie przygotowanie rankingu cech na podstawie kryteriów korelacji i informacji oraz wybór i zastosowanie dziesięciu najpopularniejszych metod sztucznej inteligencji do klasyfikacji cukrzycy typu 1. Wyniki uzyskane przez każdy z wybranych algorytmów zostały zwalidowane za pomocą metryk wydajnościowych, a następnie porównane w celu wyboru optymalnego algorytmu do klasyfikacji rozwiązywanego problemu. Wykonano także wizualizację reguł decyzyjnych w postaci drzewa decyzyjnego, umożliwiającego zrozumienie, w jaki sposób system klasyfikował poszczególne rekordy danych. Dzięki jednoczesnemu zastosowaniu metody klastrowania K-means i klasyfikacji drzewem decyzyjnym odkryto pojedynczą regułę, na podstawie której możliwe jest przewidywanie cukrzycy typu 1 u dzieci.

#### **2. Opracowanie metody pozwalającej na automatyczne, jednoczesne rozpoznawanie i zliczanie czerwonych i białych krwinek oraz płytek krwi na podstawie zdjęć mikroskopowych z wykorzystaniem głębokich sieci neuronowych**

Zadanie zrealizowano przez zgromadzenie rzeczywistego zbioru danych uczących składającego się z 900 obrazów zawierających czerwone i białe krwinki oraz płytki krwi, a także manualne dodanie adnotacji do zdjęć, a następnie wytrenowanie sieci RetinaNet ze szkieletem ResNet50 do rozpoznawania i klasyfikacji komórek krwi przez wykonywa-

nie różnej liczby epok i zapisywanie poszczególnych modeli. Następnie przetestowano modele na zbiorze obrazów leukocytów do segmentacji i klasyfikacji (Leukocyte Images for Segmentation and Classification, LISC) i zbadano wpływ liczby epok na spadki w funkcji straty. Zautomatyzowano proces jednoczesnego zliczania trzech różnych typów komórek na jednym obrazie z wykorzystaniem głębokiej sieci konwolucyjnej RetinaNet przez opracowanie dedykowanego narzędzia. Wykonano testy wyników klasyfikacji czerwonych i białych krwinek oraz płytek krwi na 15 zdjęciach w sposób manualny, obliczając wartości metryki F1 dla modeli uczonych przez 10, 15, 20, 25, 30, 35 i 40 epok dla poszczególnych wartości progowych i wybór optymalnych modeli do dalszych testów (RN10 i RN30). Następnie wykonano manualnie testy modeli RN10 i RN30 dla poszczególnych rodzajów komórek krwi na większej liczbie zdjęć dla poszczególnych wartości progowych za pomocą metryk wydajnościowych, takich jak dokładność, precyzja, czułość i miara F1 oraz porównanie uzyskanych wyników i wybrano optymalne progi do zliczania konkretnych rodzajów krwinek. Ustalono również optymalną wartość progową do zliczania wszystkich typów komórek jednocześnie. Otrzymane wyniki porównano z wynikami innych autorów zajmujących się tematyką liczenia krwinek czerwonych, białych i płytek krwi.

### **3. Opracowanie aplikacji umożliwiającej automatyzację procesu oceny, składania i identyfikacji sekwencji genomowych uzyskanych za pomocą nowych metod sekwencjonowania przy użyciu narzędzi korzystających z metod uczenia maszynowego**

Zadanie zostało zrealizowane dzięki wykonaniu implementacji aplikacji-serwera NanoForms umożliwiającej automatyzację procesu oceny, składania i identyfikacji sekwencji genomowych uzyskanych za pomocą nowych metod sekwencjonowania. Zaproponowano interfejs użytkownika, który wymaga jego minimalnej interakcji, co jest sporym uproszczeniem w stosunku do innych dostępnych na rynku narzędzi. Zaproponowano modułową infrastrukturę systemu składającego się z aplikacji, a także narzędzi służących do przetwarzania danych genomicznych, umożliwiających jednoczesne wykonywanie wielu analiz. Zintegrowano powszechnie stosowane pakiety bioinformatyczne, takie jak Bandage, Fastp, FastQC, Filtlong, Flye, Kraken 2, Kraken Tools, Krona, Medaka, Nanofilt, NanoPlot, Prokka, Rebaler, Unicycler oraz QUAST, a także dodano możliwość konfigurowania różnych parametrów związanych ze składaniem sekwencji genomowych. Serwer został skonfigurowany oraz udostępniony do niekomercyjnego użytku publicz-

nego pod adresem <https://nanofoms.tech/home/>, a kod źródłowy aplikacji-serwera został udostępniony w postaci otwartego oprogramowania. Użytkownikom zapewniono prostą instalację serwera na własnym sprzęcie, dzięki zastosowaniu konteneryzacji aplikacji przy użyciu narzędzi Docker i Docker-compose. Porównano także możliwości oferowane przez NanoForms z innymi serwerami tego typu.

#### **4. Implementacja w języku Python klasyfikatora GPR opartego na logice rozmytej i programowaniu ekspresji genów, służącego do generowania wysoce interpretowalnych reguł rozmytych**

Zadanie zrealizowano przez wykonanie implementacji klasyfikatora GPR w języku Python, który osiąga bardzo dobre wyniki pod względem dokładności i pola pod krzywą ROC. Interfejs algorytmu jest spójny z biblioteką algorytmów uczenia maszynowego Scikit-learn, co umożliwia prostą integrację z jej modułami (np. do obliczania metryk wydajnościowych). W implementacji algorytmu zaproponowano wiele usprawnień pierwotnej wersji algorytmu GPR, obejmujących automatyzację procesu generowania metareguł językowych „jeżeli-to”, dodanie poprzednika metareguł „średni” (*Medium*) oraz określenia intensywności „bardzo” (*very*), obliczenie współczynnika ufności dla każdej reguły. Przygotowano przykłady użycia algorytmu oraz wygenerowanych metareguł wraz z graficzną reprezentacją granic systemu decyzyjnego dla danych medycznych. Porównano także reguły otrzymane przez algorytm GPR zaimplementowany w języku Python z regułami otrzymanymi przez pierwotną implementację algorytmu. Oprogramowanie udostępniono na licencji gwarantującej dostęp do kodu źródłowego. Dokumentacja algorytmu została przygotowana z uwzględnieniem wymagań sprzętowych, zawiera sposób instalacji oprogramowania, opis modułów oraz przykłady użycia.

#### **5. Opracowanie narzędzia pozwalającego na eksperymentalne porównanie wybranych rozmytych algorytmów opartych na regułach do klasyfikacji danych medycznych**

Zadanie zostało zrealizowane przez porównanie wyników osiąganych przez 16 wybranych algorytmów opartych na regułach na 12 zbiorach danych medycznych, opierające się na metrykach wydajnościowych, a następnie zbadanie rozkładów wartości współczynnika korelacji Matthews, dokładności i pola pod krzywą ROC dla każdego algorytmu we wszystkich zbiorach danych. Dokonano porównania klasyfikatorów pod względem średniej długości reguł generowanych w zbiorze danych, średniej liczby reguł w zbiorze, średniej liczby atrybutów na regułę w zbiorze oraz średniej liczby unikato-

wych atrybutów na regułę w zbiorze. Przedstawiono konkretne przykłady lingwistycznych reguł rozmytych „jeżeli-to” wygenerowanych przez rozmyte klasyfikatory oparte na regułach na zbiorze danych dotyczących cukrzycy typu 1, a także wykonano test rang znakowanych Wilcoxon’a w celu statystycznego porównania wyników osiąganych przez GPR z wynikami innych algorytmów opartych na regułach rozmytych. Rezultaty tej pracy mogą się przyczynić do rozwiązania trudnego problemu polegającego na zaprojektowaniu systemu wspomagania decyzji medycznych, który byłby bardziej przyjazny dla jego użytkownika (mała liczba łatwych do zrozumienia reguł) przy zachowaniu dobrych wskaźników klasyfikacji danych (dokładność, czułość itd.).

## **Wkład autorki**

Główny wkład autorki rozprawy w działalność naukową w dyscyplinie informatyka techniczna i telekomunikacja polega na:

- identyfikacji i sformułowaniu problemów badawczych, które są ważne z perspektywy usprawnienia oraz poprawy jakości i skuteczności procesu diagnostyki medycznej za pomocą metod sztucznej inteligencji,
- przeprowadzeniu przeglądu literatury i przedstawieniu aktualnego stanu wiedzy,
- wykorzystaniu nowoczesnych narzędzi i metod informatycznych, w tym języka programowania Python wraz z jego bibliotekami oraz zaawansowanych metod sztucznej inteligencji, takich jak algorytmy płytkie i głębokie,
- znacznym udziale w zastosowaniu metod sztucznej inteligencji do klasyfikacji cukrzycy typu 1,
- znacznym udziale w opracowaniu metody pozwalającej na automatyczne, jednoczesne rozpoznawanie i zliczanie czerwonych i białych krwinek oraz płytek krwi na podstawie zdjęć mikroskopowych z wykorzystaniem głębokich metod uczenia maszynowego,
- znacznym udziale w opracowaniu aplikacji umożliwiającej automatyzację procesu oceny, składania i identyfikacji sekwencji genomowych uzyskanych za pomocą nowych metod sekwencjonowania z wykorzystaniem narzędzi korzystających z metod uczenia maszynowego,

- znacznym udziale w wykonaniu w języku Python implementacji klasyfikatora GPR opartego na logice rozmytej i programowaniu ekspresji genów, służącego do generowania wysoce interpretowalnych reguł rozmytych,
- samodzielnym przygotowaniu oprogramowania pozwalającego na eksperymentalne porównanie wybranych rozmytych algorytmów opartych na regułach do klasyfikacji danych medycznych,
- sformułowaniu wniosków wynikających z przeprowadzonych eksperymentów,
- znacznym udziale w opracowaniu publikacji naukowych dotyczących wymienionych zagadnień.

## **Kierunki dalszych badań**

Obiecujące wyniki otrzymane w ramach przeprowadzonych eksperymentów potwierdzają celowość prowadzenia dalszych badań. Multidyscyplinarny charakter niniejszej pracy otwiera szerokie możliwości do kontynuowania prac nad ulepszeniem i rozwinięciem zaproponowanych rozwiązań. Idee przedstawione w tej rozprawie potencjalnie mogą zostać rozwinięte w następujący sposób:

- Opracowanie aplikacji mobilnej do diagnostyki cukrzycy typu 1, skierowanej do dzieci i młodzieży, która może być skutecznym narzędziem w opiece zdrowotnej oraz może prowadzić do lepszego zrozumienia wpływu aktywności fizycznej na zdrowie i rozwijania bardziej skutecznych strategii przeciwdziałania problemom wynikającym z jej braku.
- Rozwinięcie oprogramowania służącego do identyfikacji i zliczania komórek na obrazach z rozmazów krwi w aplikację, pozwalającą użytkownikowi na dodawanie własnych zdjęć z rozmazu krwi z poziomu interfejsu użytkownika i umożliwiającą automatyczne zliczanie trzech rodzajów krwinek.
- Rozszerzenie możliwości serwera NanoForms przez wykorzystanie sekwencji DNA do budowy drzew filogenetycznych, dodanie możliwości wykrywania biosyntetycznych klastrów genów metabolitów wtórnych w genomach bakterii i grzybów, rozszerzenie możliwości oceny jakości złożenia genomu i kompletności adnotacji.

- Przeprowadzenie optymalizacji hiperparametrów algorytmu GPR polegającej na doborze optymalnej wartości progowej dla funkcji dyskryminacyjnej, mającej na celu zastąpienie przyjętej wartości progowej wynoszącej 0.5.
- Stworzenie interfejsu użytkownika, którego zadaniem będzie generowanie prostych reguł decyzyjnych za pomocą optymalnego dla danego zadania algorytmu i wyświetlaniu ich w ujednolicony sposób, co pozwoli na zwiększenie użyteczności algorytmów opartych na regułach w medycynie i ułatwienie podejmowania decyzji medycznych.
- Przeprowadzenie testów zużycia zasobów sprzętowych komputera i pomiaru czasu działania algorytmów opartych na regułach, dostrojenie wartości hiperparametrów, a także przetestowanie wydajności algorytmów na większej liczbie zbiorów danych.



## Literatura

- [1] A. M. Turing, Computing Machinery and Intelligence, *Mind* LIX (236) (1950) 433–460. doi:10.1093/mind/lix.236.433.  
URL <https://doi.org/10.1093/mind/lix.236.433>
- [2] V. Kaul, S. Enslin, S. A. Gross, History of artificial intelligence in medicine, *Gastrointestinal Endoscopy* 92 (4) (2020) 807–812. doi:10.1016/j.gie.2020.06.040.  
URL <https://doi.org/10.1016/j.gie.2020.06.040>
- [3] M. Warszycki, Wykorzystanie sztucznej inteligencji do predykcji emocji konsumentów, *Studia i Prace Kolegium Zarządzania i Finansów* (173) (2019) 111–121. doi:10.33119/sip.2019.173.7.  
URL <https://doi.org/10.33119/sip.2019.173.7>
- [4] E. H. Shortliffe, Mycin: A knowledge-based computer program applied to infectious diseases, *Proc Annu Symp Comput Appl Med Care* (1977) 66–69.  
URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2464549/pdf/procascamc00015-0074.pdf>
- [5] M. Furmankiewicz, P. Ziuziański, Systemy ekspertowe w e-zdrowiu: studium przypadku diagnostyki grypy, *Zeszyty Naukowe Warszawskiej Wyższej Szkoły Informatyki* 8 (11) (2014) 55–68.
- [6] C. Kulikowski, S. Weiss, Representation of expert knowledge for consultation: The CASNET and EXPERT projects, *Artificial Intelligence in Medicine*. Szolovits, P., editor. Westview Press, Inc. Boulder, Colorado (1982) 21–56.
- [7] Y. Chen, J. E. Argentinis, G. Weber, IBM Watson: How cognitive computing can be applied to big data challenges in life sciences research, *Clinical Therapeutics* 38 (4) (2016) 688–701. doi:10.1016/j.clinthera.2015.12.001.  
URL <https://doi.org/10.1016/j.clinthera.2015.12.001>
- [8] N. Bakkar, T. Kovalik, I. Lorenzini, S. Spangler, A. Lacoste, K. Sponaugle, P. Ferrante, E. Argentinis, R. Sattler, R. Bowser, Artificial intelligence in neurodegenerative disease research: use of IBM Watson to identify additional RNA-binding

proteins altered in amyotrophic lateral sclerosis, *Acta Neuropathologica* 135 (2) (2017) 227–247. doi:10.1007/s00401-017-1785-8.

URL <https://doi.org/10.1007/s00401-017-1785-8>

[9] ChatGPT - OpenAI, <https://chat.openai.com/>, dostęp: 8.06.2023 r.

[10] A. Hosny, C. Parmar, J. Quackenbush, L. H. Schwartz, H. J. W. L. Aerts, Artificial intelligence in radiology, *Nature Reviews Cancer* 18 (8) (2018) 500–510. doi:10.1038/s41568-018-0016-5.

URL <https://doi.org/10.1038/s41568-018-0016-5>

[11] X. Tang, The role of artificial intelligence in medical imaging research, *BJR|Open* 2 (1) (2020) 20190031. doi:10.1259/bjro.20190031.

URL <https://doi.org/10.1259/bjro.20190031>

[12] N. Gautam, P. Saluja, A. Malkawi, M. G. Rabbat, M. H. Al-Mallah, G. Pontone, Y. Zhang, B. C. Lee, S. J. Al'Aref, Current and future applications of artificial intelligence in coronary artery disease, *Healthcare* 10 (2) (2022) 232. doi:10.3390/healthcare10020232.

URL <https://doi.org/10.3390/healthcare10020232>

[13] A. De, A. Sarda, S. Gupta, S. Das, Use of artificial intelligence in dermatology, *Indian Journal of Dermatology* 65 (5) (2020) 352. doi:10.4103/ijd.ijd\_418\_20.

URL [https://doi.org/10.4103/ijd.ijd\\_418\\_20](https://doi.org/10.4103/ijd.ijd_418_20)

[14] J. Elkhader, O. Elemento, Artificial intelligence in oncology: From bench to clinic, *Seminars in Cancer Biology* 84 (2022) 113–128. doi:10.1016/j.semcancer.2021.04.013.

URL <https://doi.org/10.1016/j.semcancer.2021.04.013>

[15] R. Dias, A. Torkamani, Artificial intelligence in clinical and genomic diagnostics, *Genome Medicine* 11 (1) (Nov. 2019). doi:10.1186/s13073-019-0689-8.

URL <https://doi.org/10.1186/s13073-019-0689-8>

[16] U. K. Patel, A. Anwar, S. Saleem, P. Malik, B. Rasul, K. Patel, R. Yao, A. Seshadri, M. Yousufuddin, K. Arumaithurai, Artificial intelligence as an emerging technology in the current care of neurological disorders, *Journal of Neurology*

- 268 (5) (2019) 1623–1642. doi:10.1007/s00415-019-09518-3.  
URL <https://doi.org/10.1007/s00415-019-09518-3>
- [17] A. Nomura, M. Noguchi, M. Kometani, K. Furukawa, T. Yoneda, Artificial intelligence in current diabetes management and prediction, *Current Diabetes Reports* 21 (12) (Dec. 2021). doi:10.1007/s11892-021-01423-2.  
URL <https://doi.org/10.1007/s11892-021-01423-2>
- [18] Y. E. Alaoui, A. Elomri, M. Qaraqe, R. Padmanabhan, R. Y. Taha, H. E. Omri, A. E. Omri, O. Aboumarzouk, A review of artificial intelligence applications in hematology management: Current practices and future prospects, *Journal of Medical Internet Research* 24 (7) (2022) e36490. doi:10.2196/36490.  
URL <https://doi.org/10.2196/36490>
- [19] X. Li, T. Zhang, An exploration on artificial intelligence application: From security, privacy and ethic perspective, in: *2017 IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, 2017, pp. 416–420. doi:10.1109/ICCCBDA.2017.7951949.
- [20] T. Davenport, R. Kalakota, The potential for artificial intelligence in healthcare, *Future Healthcare Journal* 6 (2) (2019) 94–98. doi:10.7861/futurehosp.6-2-94.  
URL <https://doi.org/10.7861/futurehosp.6-2-94>
- [21] S. Prakash, J. N. Balaji, A. Joshi, K. M. Surapaneni, Ethical conundrums in the application of artificial intelligence (AI) in healthcare—a scoping review of reviews, *Journal of Personalized Medicine* 12 (11) (2022) 1914. doi:10.3390/jpm12111914.  
URL <https://doi.org/10.3390/jpm12111914>
- [22] J. E. H. Korteling, G. C. van de Boer-Visschedijk, R. A. M. Blankendaal, R. C. Boonekamp, A. R. Eikelboom, Human- versus artificial intelligence, *Frontiers in Artificial Intelligence* 4 (Mar. 2021). doi:10.3389/frai.2021.622364.  
URL <https://doi.org/10.3389/frai.2021.622364>
- [23] P. Kawalec, M. Kielar, A. Pilc, Costs related to type 1 and 2 diabetes mellitus in poland, *Clinical Diabetology* 7 (5) (2006) 287 – 294.
- [24] M. Monaghan, V. Helgeson, D. Wiebe, Type 1 diabetes in young adulthood, *Current Diabetes Reviews* 11 (4) (2015) 239–250. doi:10.2174/

1573399811666150421114957.

URL <https://doi.org/10.2174/1573399811666150421114957>

- [25] R. Streisand, M. Monaghan, Young children with type 1 diabetes: Challenges, research, and future directions, *Current Diabetes Reports* 14 (9) (Jul. 2014). doi: 10.1007/s11892-014-0520-2.  
URL <https://doi.org/10.1007/s11892-014-0520-2>
- [26] M. A. Zamarlik, K. Piątek, Providing care for children with type 1 diabetes in kindergartens and schools, *Pediatric Endocrinology Diabetes and Metabolism* 26 (4) (2020) 205–210. doi:10.5114/pedm.2020.98998.  
URL <https://doi.org/10.5114/pedm.2020.98998>
- [27] C. Kamrath, J. Rosenbauer, A. J. Eckert, K. Siedler, H. Bartelt, D. Klose, M. Sindichakis, S. Herrlinger, V. Lahn, R. W. Holl, Incidence of type 1 diabetes in children and adolescents during the COVID-19 pandemic in germany: Results from the DPV registry, *Diabetes Care* 45 (8) (2022) 1762–1771. doi:10.2337/dc21-0969.  
URL <https://doi.org/10.2337/dc21-0969>
- [28] 16. diabetes care in the hospital: Standards of medical care in diabetes 2023, *Diabetes Care* 45 (Supplement\_1) (2021) S244–S253. doi:10.2337/dc22-s016.  
URL <https://doi.org/10.2337/dc22-s016>
- [29] 2021 guidelines on the management of patients with diabetes. a position of diabetes poland, *Clinical Diabetology* 10 (1) (2021) 1–113. doi:10.5603/dk.2021.0001.  
URL <https://doi.org/10.5603/dk.2021.0001>
- [30] A. D. Deshpande, M. Harris-Hayes, M. Schootman, Epidemiology of diabetes and diabetes-related complications, *Physical Therapy* 88 (11) (2008) 1254–1264. doi: 10.2522/ptj.20080020.  
URL <https://doi.org/10.2522/ptj.20080020>
- [31] Z.-P. Teng, R. Tian, F.-L. Xing, H. Tang, J.-J. Xu, B.-W. Zhang, J.-W. Qi, An association of type 1 diabetes mellitus with auditory dysfunction: A systematic review and meta-analysis, *The Laryngoscope* 127 (7) (2016) 1689–1697. doi: 10.1002/lary.26346.  
URL <https://doi.org/10.1002/lary.26346>

- [32] J. K. Rustad, D. L. Musselman, C. B. Nemeroff, The relationship of depression and diabetes: Pathophysiological and treatment implications, *Psychoneuroendocrinology* 36 (9) (2011) 1276–1286. doi:10.1016/j.psyneuen.2011.03.005.  
URL <https://doi.org/10.1016/j.psyneuen.2011.03.005>
- [33] C. Tana, S. Ballestri, F. Ricci, A. D. Vincenzo, A. Ticinesi, S. Gallina, M. A. Giamberardino, F. Cipollone, R. Sutton, R. Vettor, A. Fedorowski, T. Meschi, Cardiovascular risk in non-alcoholic fatty liver disease: Mechanisms and therapeutic implications, *International Journal of Environmental Research and Public Health* 16 (17) (2019) 3104. doi:10.3390/ijerph16173104.  
URL <https://doi.org/10.3390/ijerph16173104>
- [34] P. A. Moore, R. J. Weyant, M. B. Mongelluzzo, D. E. Myers, K. Rossie, J. Guggenheimer, H. Hubar, H. M. Block, T. Orchard, Type 1 diabetes mellitus and oral health: Assessment of tooth loss and edentulism, *Journal of Public Health Dentistry* 58 (2) (1998) 135–142. doi:10.1111/j.1752-7325.1998.tb02498.x.  
URL <https://doi.org/10.1111/j.1752-7325.1998.tb02498.x>
- [35] D. B. Sacks, M. Arnold, G. L. Bakris, D. E. Bruns, A. R. Horvath, M. S. Kirkman, A. Lernmark, B. E. Metzger, D. M. Nathan, Guidelines and recommendations for laboratory analysis in the diagnosis and management of diabetes mellitus, *Diabetes Care* 34 (6) (2011) e61–e99. doi:10.2337/dc11-9998.  
URL <https://doi.org/10.2337/dc11-9998>
- [36] K. Tonyushkina, J. H. Nichols, Glucose meters: A review of technical challenges to obtaining accurate results, *Journal of Diabetes Science and Technology* 3 (4) (2009) 971–980. doi:10.1177/193229680900300446.  
URL <https://doi.org/10.1177/193229680900300446>
- [37] E. Czenczek-Lewandowska, Poziom aktywności fizycznej dzieci i młodzieży z cukrzycą typu 1 w zależności od metody stosowanej insulinoterapii, Ph.D. thesis, Uniwersytet Rzeszowski, Wydział Medyczny (2017).
- [38] S. Osowski, Głębokie sieci neuronowe i ich zastosowania w eksploracji danych, *Przegląd Telekomunikacyjny + Wiadomości Telekomunikacyjne* 5 (2018) 112–121. doi:10.15199/59.2018.5.2.

URL <http://yadda.icm.edu.pl/baztech/element/bwmeta1.element.baztech-68050b0c-aef5-432a-b5c6-c5b2174405a3>

- [39] L. Agnello, R. V. Giglio, G. Bivona, C. Scazzone, C. M. Gambino, A. Iacona, A. M. Ciaccio, B. L. Sasso, M. Ciaccio, The value of a complete blood count (CBC) for sepsis diagnosis and prognosis, *Diagnostics* 11 (10) (2021) 1881. doi:10.3390/diagnostics11101881.  
URL <https://doi.org/10.3390/diagnostics11101881>
- [40] L. R. Dixon, The complete blood count: Physiologic basis and clinical usage, *The Journal of Perinatal & Neonatal Nursing* 11 (3) (1997) 1–18. doi:10.1097/00005237-199712000-00003.  
URL <https://doi.org/10.1097/00005237-199712000-00003>
- [41] M. M. Ahmed, S. K. Ghauri, A. Javaeed, N. Rafique, W. Hussain, N. Khan, Trends of utilization of complete blood count parameters for patient management among doctors in azad kashmir, *Pakistan Journal of Medical Sciences* 36 (5) (Jun. 2020). doi:10.12669/pjms.36.5.1885.  
URL <https://doi.org/10.12669/pjms.36.5.1885>
- [42] N. M. Deshpande, S. Gite, R. Aluvalu, A review of microscopic analysis of blood cells for disease detection with AI perspective, *PeerJ Computer Science* 7 (2021) e460. doi:10.7717/peerj-cs.460.  
URL <https://doi.org/10.7717/peerj-cs.460>
- [43] A. KWASIGROCH, M. GROCHOWSKI, Rozpoznawanie obiektów przez głębokie sieci neuronowe, *Zeszyty Naukowe Wydziału Elektrotechniki i Automatyki Politechniki Gdańskiej* 2018 (60) (2018) 63–66. doi:10.32016/1.60.12.  
URL <https://doi.org/10.32016/1.60.12>
- [44] M. Kim, C. Yan, D. Yang, Q. Wang, J. Ma, G. Wu, Deep learning in biomedical image analysis, in: *Biomedical Information Technology*, Elsevier, 2020, pp. 239–263. doi:10.1016/b978-0-12-816034-3.00008-0.  
URL <https://doi.org/10.1016/b978-0-12-816034-3.00008-0>
- [45] M. R. Reena, P. Ameer, Localization and recognition of leukocytes in peripheral blood: A deep learning approach, *Computers in Biology and Medicine* 126 (2020)

104034. doi:10.1016/j.compbimed.2020.104034.

URL <https://doi.org/10.1016/j.compbimed.2020.104034>

- [46] A. Khan, A. Eker, A. Chefranov, H. Demirel, White blood cell type identification using multi-layer convolutional features with an extreme-learning machine, *Biomedical Signal Processing and Control* 69 (2021) 102932. doi:10.1016/j.bspc.2021.102932.  
URL <https://doi.org/10.1016/j.bspc.2021.102932>
- [47] X. Huang, H. Jeon, J. Liu, J. Yao, M. Wei, W. Han, J. Chen, L. Sun, J. Han, Deep-learning based label-free classification of activated and inactivated neutrophils for rapid immune state monitoring, *Sensors* 21 (2) (2021) 512. doi:10.3390/s21020512.  
URL <https://doi.org/10.3390/s21020512>
- [48] M. Loey, M. Naman, H. Zayed, Deep transfer learning in diagnosing leukemia in blood cells, *Computers* 9 (2) (2020) 29. doi:10.3390/computers9020029.  
URL <https://doi.org/10.3390/computers9020029>
- [49] Q. Wang, S. Bi, M. Sun, Y. Wang, D. Wang, S. Yang, Deep learning approach to peripheral leukocyte recognition, *PLOS ONE* 14 (6) (2019) e0218808. doi:10.1371/journal.pone.0218808.  
URL <https://doi.org/10.1371/journal.pone.0218808>
- [50] M. H. Motlagh, M. Jannesari, Z. Rezaei, M. Totonchi, H. Baharvand, Automatic white blood cell classification using pre-trained deep learning models: ResNet and inception, in: J. Zhou, P. Radeva, D. Nikolaev, A. Verikas (Eds.), *Tenth International Conference on Machine Vision (ICMV 2017)*, SPIE, 2018. doi:10.1117/12.2311282.  
URL <https://doi.org/10.1117/12.2311282>
- [51] M. M. Alam, M. T. Islam, Machine learning approach of automatic identification and counting of blood cells, *Healthcare Technology Letters* 6 (4) (2019) 103–108. doi:10.1049/htl.2018.5098.  
URL <https://doi.org/10.1049/htl.2018.5098>

- [52] A. Acevedo, S. Alférez, A. Merino, L. Puigví, J. Rodellar, Recognition of peripheral blood cell images using convolutional neural networks, *Computer Methods and Programs in Biomedicine* 180 (2019) 105020. doi:10.1016/j.cmpb.2019.105020. URL <https://doi.org/10.1016/j.cmpb.2019.105020>
- [53] C. D. Ruberto, A. Loddo, L. Putzu, Detection of red and white blood cells from microscopic blood images using a region proposal approach, *Computers in Biology and Medicine* 116 (2020) 103530. doi:10.1016/j.combiomed.2019.103530. URL <https://doi.org/10.1016/j.combiomed.2019.103530>
- [54] V. D. Dvanesh, P. S. Lakshmi, K. Reddy, A. S. Vasavi, Blood cell count using digital image processing, in: *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, IEEE, 2018. doi:10.1109/icctct.2018.8550999. URL <https://doi.org/10.1109/icctct.2018.8550999>
- [55] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection (2017).
- [56] S. H. Rezaatofghi, H. Soltanian-Zadeh, Automatic recognition of five types of white blood cells in peripheral blood, *Computerized Medical Imaging and Graphics* 35 (4) (2011) 333–343. doi:10.1016/j.compmedimag.2011.01.003. URL <https://doi.org/10.1016/j.compmedimag.2011.01.003>
- [57] A. Żmieńko, A. Satyr, Sekwencjonowanie nanoporowe i jego zastosowanie w biologii, *Postępy Biochemii* (2020) 1–12doi:10.18388/pb.2020\_328. URL [https://doi.org/10.18388/pb.2020\\_328](https://doi.org/10.18388/pb.2020_328)
- [58] A. Behera, Use of artificial intelligence for management and identification of complications in diabetes, *Clinical Diabetology* (Feb. 2021). doi:10.5603/dk.a2021.0007. URL <https://doi.org/10.5603/dk.a2021.0007>
- [59] S. Quainoo, J. P. M. Coolen, S. A. F. T. van Hijum, M. A. Huynen, W. J. G. Melchers, W. van Schaik, H. F. L. Wertheim, Whole-genome sequencing of bacterial pathogens: the future of nosocomial outbreak analysis, *Clinical Microbiology*



- Reviews 30 (4) (2017) 1015–1063. doi:10.1128/cmr.00016-17.  
URL <https://doi.org/10.1128/cmr.00016-17>
- [60] F. R. Fields, S. W. Lee, M. J. McConnell, Using bacterial genomes and essential genes for the development of new antibiotics, *Biochemical Pharmacology* 134 (2017) 74–86. doi:10.1016/j.bcp.2016.12.002.  
URL <https://doi.org/10.1016/j.bcp.2016.12.002>
- [61] D. E. Wood, S. L. Salzberg, Kraken: ultrafast metagenomic sequence classification using exact alignments, *Genome Biology* 15 (3) (Mar. 2014). doi:10.1186/gb-2014-15-3-r46.  
URL <https://doi.org/10.1186/gb-2014-15-3-r46>
- [62] D. E. Wood, J. Lu, B. Langmead, Improved metagenomic analysis with Kraken 2, *Genome Biology* 20 (1) (Nov. 2019). doi:10.1186/s13059-019-1891-0.  
URL <https://doi.org/10.1186/s13059-019-1891-0>
- [63] Oxford Nanopore Technologies Ltd., Medaka, <https://github.com/nanoporetech/medaka>, dostep: 24.02.2023 r. (2008 - 2023).
- [64] M. V. Larsen, K. G. Joensen, E. Zankari, J. Ahrenfeldt, O. Lukjancenko, R. S. Kaas, L. Roer, P. Leekitcharoenphon, D. Saputra, S. Cosentino, M. C. F. Thomsen, J. L. B. Cisneros, V. Jurtz, S. Rasmussen, T. N. Petersen, H. Hasman, T. Sicheritz-Ponten, F. M. Aarestrup, O. Lund, The CGE tool box, in: *Applied Genomics of Foodborne Pathogens*, Springer International Publishing, 2017, pp. 65–90. doi:10.1007/978-3-319-43751-4\_5.  
URL [https://doi.org/10.1007/978-3-319-43751-4\\_5](https://doi.org/10.1007/978-3-319-43751-4_5)
- [65] Z. Zhou, N.-F. Alikhan, K. Mohamed, Y. Fan, M. A. and, The EnteroBase user’s guide, with case studies on *Salmonella* transmissions, *Yersinia pestis* phylogeny, and *Escherichia* core genomic diversity, *Genome Research* 30 (1) (2019) 138–152. doi:10.1101/gr.251678.119.  
URL <https://doi.org/10.1101/gr.251678.119>
- [66] P. J. Cock, B. A. Grüning, K. Paszkiewicz, L. Pritchard, Galaxy tools and workflows for sequence analysis with applications in molecular plant pathology, *PeerJ*

1 (2013) e167. doi:10.7717/peerj.167.

URL <https://doi.org/10.7717/peerj.167>

- [67] W. de Koning, M. Miladi, S. Hiltemann, A. Heikema, J. P. Hays, S. Flemming, M. van den Beek, D. A. Mustafa, R. Backofen, B. Grüning, A. P. Stubbs, NanoGalaxy: Nanopore long-read sequencing data analysis in Galaxy, *GigaScience* 9 (10) (Oct. 2020). doi:10.1093/gigascience/giaa105.  
URL <https://doi.org/10.1093/gigascience/giaa105>
- [68] J. J. Gillespie, A. R. Wattam, S. A. Cammer, J. L. Gabbard, M. P. Shukla, O. Dalay, T. Driscoll, D. Hix, S. P. Mane, C. Mao, E. K. Nordberg, M. Scott, J. R. Schulman, E. E. Snyder, D. E. Sullivan, C. Wang, A. Warren, K. P. Williams, T. Xue, H. S. Yoo, C. Zhang, Y. Zhang, R. Will, R. W. Kenyon, B. W. Sobral, PATRIC: the comprehensive bacterial bioinformatics resource with a focus on human pathogenic species, *Infection and Immunity* 79 (11) (2011) 4286–4298. doi:10.1128/iai.00207-11.  
URL <https://doi.org/10.1128/iai.00207-11>
- [69] Oxford Nanopore Technologies Ltd., EPI2ME, <https://epi2me.nanoporetech.com>, dostęp: 24.02.2023 r. (2008 - 2023).
- [70] V. Shabardina, T. Kischka, F. Manske, N. Grundmann, M. C. Frith, Y. Suzuki, W. Makałowski, NanoPipe—a web server for nanopore MinION sequencing data analysis, *GigaScience* 8 (2) (2019) giy169.
- [71] R. Tadeusiewicz, *Informatyka Medyczna, Uniwersytet Marii Curie-Skłodowskiej w Lublinie, Instytut Informatyki*, 2011.
- [72] P. Kaźmierczyk, M. Kupis, M. Maj, *Biała Księga AI w Praktyce Klinicznej, Koalicja AI w Zdrowiu*, Warszawa, 2022.
- [73] J. Kluska, M. Madera, Extremely simple classifier based on fuzzy logic and gene expression programming, *Information Sciences* 571 (2021) 560–579. doi:10.1016/j.ins.2021.05.041.  
URL <https://doi.org/10.1016/j.ins.2021.05.041>
- [74] J. Kluska, Selected applications of P1-TS fuzzy rule-based systems, in: L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, J. M. Zurada

- (Eds.), *Artificial Intelligence and Soft Computing*, Springer International Publishing, Cham, 2015, pp. 195–206.
- [75] J. Kluska, *Analytical Methods in Fuzzy Modeling and Control*, Studies in Fuzziness and Soft Computing, Springer, Berlin, Heidelberg, 2009. doi:10.1007/978-3-540-89927-3.
- [76] J. Kluska, Transformation lemma on analytical modeling via Takagi-Sugeno fuzzy system and its applications, in: L. Rutkowski, R. Tadeusiewicz, L. A. Zadeh, J. M. Żurada (Eds.), *Artificial Intelligence and Soft Computing – ICAISC 2006*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 230–239.
- [77] C. Ferreira, *Gene expression programming: mathematical modeling by an artificial intelligence*, Springer-Verlag, Berlin, 2006. doi:10.1007/3-540-32849-1.
- [78] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, C. Gagné, DEAP: Evolutionary algorithms made easy, *Journal of Machine Learning Research* 13 (2012) 2171–2175.  
URL <https://www.jmlr.org/papers/volume13/fortin12a/fortin12a.pdf>
- [79] S. Gao, Geppy: a Python framework for gene expression programming (2020). doi:10.5281/zenodo.3946297.
- [80] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T. E. Oliphant, Array programming with NumPy, *Nature* 585 (7825) (2020) 357–362. doi:10.1038/s41586-020-2649-2.
- [81] S. Raschka, V. Mirjalili, *Python Machine Learning*, 2nd Edition, Packt Publishing Ltd., Livery Place 35 Livery Street Birmingham B3 2PB, UK, 2017.
- [82] A. F. Markus, J. A. Kors, P. R. Rijnbeek, The role of explainability in creating trustworthy artificial intelligence for health care: A comprehensive survey of the terminology, design choices, and evaluation strategies, *Journal of Biomedical Informatics* 113 (2021) 103655. doi:10.1016/j.jbi.2020.103655.  
URL <https://doi.org/10.1016/j.jbi.2020.103655>

- [83] A. Niederliński, Regułow-modelowe systemy ekspertowe rmse, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice, 2013.
- [84] M. Mazurek, Architektura systemu wspomaganie decyzji medycznych wykorzystująca technologię przetwarzania danych big data, Roczniki Kolegium Analiz Ekonomicznych (nr 35 Technologie informatyczne w służbie zdrowia) (2014) 257–271.
- [85] M. Gacto, R. Alcalá, F. Herrera, Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures, Information Sciences 181 (20) (2011) 4340–4360. doi:10.1016/j.ins.2011.02.021.  
URL <https://doi.org/10.1016/j.ins.2011.02.021>
- [86] J. Alcala-Fdez, A. Fernández, J. Luengo, J. Derrac, S. Garc'ia, L. Sanchez, F. Herrera, KEEL Data-Mining Software Tool: Data set repository, integration of algorithms and experimental analysis framework, Journal of Multiple-Valued Logic and Soft Computing 17 (2010) 255–287.
- [87] J. Kluska, M. Kusy, B. Obrzut, The classifier for prediction of peri-operative complications in cervical cancer treatment, in: Artificial Intelligence and Soft Computing, Springer International Publishing, 2014, pp. 143–154. doi:10.1007/978-3-319-07176-3\_13.  
URL [https://doi.org/10.1007/978-3-319-07176-3\\_13](https://doi.org/10.1007/978-3-319-07176-3_13)

## Dorobek naukowy autorki

W niniejszej rozprawie doktorskiej opisano artykuły [A-1]-[A-5] wchodzące w skład cyklu publikacji. Dodatkowo, dorobek naukowy autorki obejmuje następujące pozycje:

### Artykuły naukowe

- [D-1] Czmił, S., Kluska, J., & **Czmił, A.** (2022). CACP: Classification Algorithms Comparison Pipeline. *SoftwareX*, 19, 101134. doi:10.1016/j.softx.2022.101134 (IF<sub>2022</sub> = 2,868, 200 punktów).
- [D-2] Bartusik-Aebisher, D., Aebisher, D., **Czmił, A.**, & Mazur, D. (2020). Evaluation of MR relaxation times following trastuzumab treatment of breast cancer cells in a 3D bioreactor. *Acta Poloniae Pharmaceutica - Drug Research*, 77(1), 35–41. doi:10.32383/appdr/115519 (IF<sub>2020</sub> = 0,447, 70 punktów, obecnie IF<sub>2022</sub> = 0,555, 100 punktów).
- [D-3] Bartusik-Aebisher, D., Aebisher, D., **Czmił, A.**, & Mazur, D. (2020). Trastuzumab Efficacy Quantified by Fluorine-19 Magnetic Resonance Imaging. *Acta Poloniae Pharmaceutica - Drug Research*, 77(3), 495–503. doi:10.32383/appdr/120010. (IF<sub>2020</sub> = 0,447, 70 punktów, obecnie IF<sub>2022</sub> = 0,555, 100 punktów).
- [D-4] Bober, Z., Galiniak, S., Leksa, D., Dynarowicz, K., Aebisher, D., Ostańska, E., **Czmił, A.**, Bar, P., Bartusik-Aebisher, D., and Tutka, P. (2020). Cytisine parameters measured in the 1.5 Tesla magnetic field. *European Journal of Clinical and Experimental Medicine*, 18(1), 16–19. doi:10.15584/ejcem.2020.1.3.
- [D-5] **Czmił, A.** (2020). Wielopoziomowe przekształtniki energoelektroniczne –topologie, zasada działania, metody modulacji. *Scientific Journals of Rzeszów University of Technology, Series: Electrotechnics*, 5–18. doi:10.7862/re.2020.1.
- [D-6] Wołoszyn, F., & **Czmił, A.** (2019). Upper limb analysis measured by inertial measurement unit tool: a case report. *European Journal of Clinical and Experimental Medicine*, 17(1), 94–100. doi:10.15584/ejcem.2019.1.16.

### Wystąpienia konferencyjne

- [K-1] **A. Czmił**, S. Czmił. *Platforma do wykrywania anomalii i predykcji zużycia energii elektrycznej w czasie rzeczywistym*. SPETO 2019 : XLII Konferencja

z Podstaw Elektrotechniki i Teorii Obwodów Polskie Towarzystwo Elektrotechniki Teoretycznej i Stosowanej. Oddział Gliwicko-Opolski. Gliwice - Ustroń 18.05.2019 r.

- [K-2] G. Drałus, D. Mazur, **A. Czmił**. *Automatic detection and counting of blood cells in images using convolutional neural networks*. International Conference WZEE 2021 – „Selected Issues of Electrical Engineering and Electronics”, Politechnika Rzeszowska im. Ignacego Łukasiewicza, Rzeszów, 14.09.2021 r.
- [K-3] **A. Czmił**, G. Drałus, A. Banaś – Ząbczyk, D. Bartusik-Aebisher, R. Depta, P. Jakubczyk, M. Kopańska, A. Myszka, J. Podgórska –Bednarz, K. Szmuc. *Classification of Autism Spectrum Disorder Using Selected Machine Learning Algorithms*. Konferencja WZEE 2022 „Innowacyjne metody leczenia i technologie w medycynie”, Uniwersytet Rzeszowski, Rzeszów, 22.09.2022 r.

#### **Udział w projektach badawczych**

- [P-1] Udział w grantie *Technologia Oxford Nanopore: optymalizacja enzymów oraz analizy danych genomicznych pod kątem zastosowań komercyjnych*, Program grantowy na prace B+R jednostek naukowych w ramach projektu *Podkarpackie Centrum Innowacji*. Kierownik projektu: dr hab. inż. Dominik Strzałka, prof. PRz.
- [P-2] Udział w grantie *Oprogramowanie bazujące na SI rozróżniające autyzm przy szerokim wykorzystaniu markerów fizykalnych*, Program grantowy na prace B+R jednostek naukowych w ramach projektu *Podkarpackie Centrum Innowacji*. Kierownik projektu: dr hab. n. med. Jacek Szczygielski, prof. UR.

#### **Inne osiągnięcia naukowe**

- [I-1] Zajęcie II miejsca w Ogólnopolskim Konkursie Młodych Inżynierów - EDYCJA IT. **A. Czmił**, S. Czmił, *Aplikacja umożliwiająca wizualizację ochrony odgrzewanej przy użyciu metody toczącej się kuli*, 2018.

## Wykaz stosowanych oznaczeń

1R-C - One Rule

3GS - third-generation sequencing (sekwencjonowanie trzeciej generacji)

A - adenina

ACC - accuracy (dokładność)

ADA - American Diabetes Association (Amerykańskie Stowarzyszenie Diabetyków)

AI - sztuczna inteligencja (artificial intelligence)

AUC - area under the ROC curve (pole pod krzywą ROC)

C - cytozyna

C4.5-C - C4.5

C45Rules-C - C4.5Rules

C45RulesSA-C - C4.5Rules Simulated Annealing Version

CNN - convolutional neural networks (konwolucyjne sieci neuronowe)

DNA - deoxyribonucleic acid (kwas deoksyrybonukleinowy)

DT - decision tree (drzewo decyzyjne)

DT\_GA-C - Hybrid Decision Tree-Genetic Algorithm

DT\_Oblique-C - Oblique Decision Tree with Evolutionary Learning

EACH-C - Exemplar-Aided Constructor of Hyperrectangles

EEG - elektroencefalografia

ENA - European Nucleotide Archive (Europejskie Archiwum Nukleotydów)

FAQ - frequently-asked questions (najczęściej zadawane pytania)

FPN - Feature Pyramid Network

G - guanina

G-index - goodness index (wskaźnik dobroci)

GEP - gene expression programming

GMDH - group method of data handling

GPLv3 - GNU General Public License (licencja wolnego i otwartego oprogramowania)

GPR - classifier based on fuzzy logic and gene expression programming

H - high (wysoki)

HbA1c - hemoglobina glikowana

Hider-C - Hierarchical Decision Rules

IoLT - Internet of Living Things

L - low (niski)  
LISC - Leukocyte Images for Segmentation and Classification  
M - medium (średni)  
MCC - Matthew's correlation coefficient (współczynnik korelacji Matthews'a)  
MDSS - medical decision support systems (systemy wspomagania decyzji medycznych)  
MLP - multilayer perceptron  
NGS - next-generation sequencing (sekwencjonowanie następnej generacji)  
NSLV-C - New Structural Learning Algorithm in a Vague Environment  
OCEC-C - Organizational Co-Evolutionary Algorithm for Classification  
OIGA-C - Ordered Incremental Genetic Algorithm  
ONT - Oxford Nanopore Technology (technologia Oxford Nanopore)  
PNN - probabilistic neural network  
Pre - precision (precyzja)  
RBF - radial basis function network  
RF - random forest, las losowy  
ROC - receiver operating characteristic (krzywa charakterystyki działania odbiornika)  
Sen - sensitivity (czułość)  
Spe - specificity (specyficzność)  
SVM - support vector machine (maszyna wektorów wspierających)  
T - tymina  
WM - weighted metric (metryka ważona)  
WDL - Workflow Description Language  
WHO - Światowa Organizacja Zdrowia  
XAI - explainable artificial intelligence



## Artykuły naukowe wchodzące w skład cyklu (opublikowane w latach 2019-2023)




W niniejszym rozdziale zamieszczono pełną treść opublikowanych prac wchodzących w skład cyklu publikacji:

- [A-1] **A. Czmił**, S. Czmił, D. Mazur. *A method to detect type 1 diabetes based on physical activity measurements using a mobile device*. Applied Sciences 9 (12) (2019) 2555. <https://doi.org/10.3390/app9122555> (str. 63).
- [A-2] G. Drałus, D. Mazur, **A. Czmił**. *Automatic detection and counting of blood cells in smear images using RetinaNet*. Entropy 23 (11) (2021) 1522. doi:10.3390/e23111522 (str. 79).
- [A-3] **A. Czmił**, M. Wroński, S. Czmił, M. Sochacka-Piętal, M. Ćmił, J. Gawor, T. Wołkowicz, D. Plewczyński, D. Strzałka, M. Piętal. *NanoForms: an integrated server for processing, analysis and assembly of raw sequencing data of microbial genomes, from Oxford Nanopore technology*. PeerJ, 2022, 10:e13056. doi:10.7717/peerj.13056 (str. 101).
- [A-4] **A. Czmił**, J. Kluska, S. Czmił. *GPR: A Python implementation of an extremely simple classifier based on fuzzy logic and gene expression programming*, SoftwareX, 2023; 22:101362. <https://doi.org/10.1016/j.softx.2023.101362> (str. 115).
- [A-5] **A. Czmił**. *Comparative Study of Rule-based Fuzzy Logic Classifiers for Medical Applications*. Sensors, 2023; 23(2):992. <https://doi.org/10.3390/s23020992> (str. 123).



Article

# A Method to Detect Type 1 Diabetes Based on Physical Activity Measurements Using a Mobile Device

Anna Czmił \*, Sylwester Czmił and Damian Mazur

Faculty of Electrical and Computer Engineering, Rzeszow University of Technology, 35-959 Rzeszow, Poland; sylwesterczmil@gmail.com (S.C.); mazur@prz.edu.pl (D.M.)

\* Correspondence: czmilanna@gmail.com

Received: 28 May 2019; Accepted: 19 June 2019; Published: 22 June 2019



**Featured Application:** Non-invasive method of type 1 diabetes detection based on physical activity measurement.

**Abstract:** Type 1 diabetes is a chronic disease marked by high blood glucose levels, called hyperglycemia. Diagnosis of diabetes typically requires one or more blood tests. The aim of this paper is to discuss a non-invasive method of type 1 diabetes detection, based on physical activity measurement. We solved a binary classification problem using a variety of computational intelligence methods, including non-linear classification algorithms, which were applied and comparatively assessed. Prediction of disease presence among children and adolescents was evaluated using performance measures, such as accuracy, sensitivity, specificity, precision, the goodness index, and AUC. The most satisfying results were obtained when using the random forest method. The primary parameters in disease detection were weekly step count and the weekly number of vigorous activity minutes. The dependence between the weekly number of steps and the type 1 diabetes presence was established after an insightful analysis of data using classification and clustering algorithms. The findings have shown promising results that type 1 diabetes can be diagnosed using physical activity measurement. This is essential regarding the non-invasiveness and flexibility of the detection method, which can be tested at any time anywhere. The proposed technique can be implemented on a mobile device.

**Keywords:** type 1 diabetes; classification; physical activity; artificial intelligence

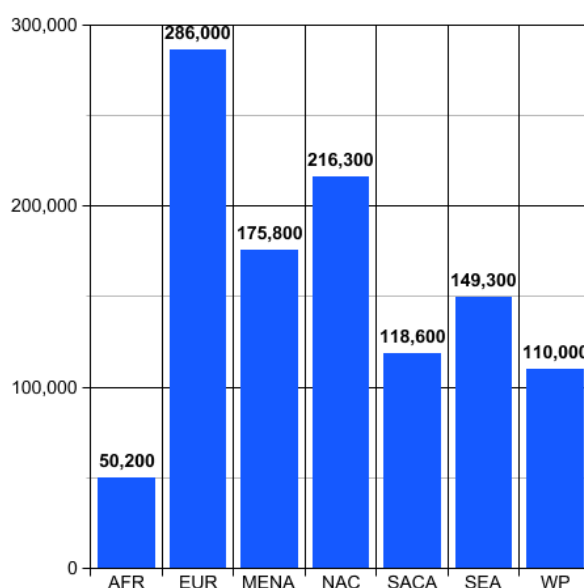
## 1. Introduction

Diabetes mellitus is a group of metabolic diseases that is characterized by hyperglycemia and results from defects in insulin action, insulin secretion, or both [1]. Elevated blood glucose connected with this disease can cause dysfunction and failure of various organs, which are the effects of long-term diabetes. Currently, according to the WHO and American Diabetes Association classification (ADA), there are four types of diabetes: type 1, type 2, other specific types of diabetes, and gestational diabetes [2,3].

Type 1 diabetes causes the patient's blood glucose to become too high. This happens when his or her body cannot produce enough insulin, which controls blood glucose. Patients need daily injections of insulin to keep blood glucose levels under control. It is one of the leading health problems in Poland and Europe, for people of all ages. It causes constant damage to health and contributes to premature death [4,5]. According to the International Diabetes Federation estimation, the incidence of type 1 diabetes among children and adolescents under the age of 15 years is increasing in many countries,

and the overall annual increase is estimated to be around 3%, with strong indications of geographic differences. More than 96,000 children and adolescents under 15 are estimated to be diagnosed with type 1 diabetes annually. The number is estimated to be more than 132,600 when the age range is extended to 20 years. In total, more than one million children globally and adolescents below 20 are estimated to have type 1 diabetes [6].

There are large regional differences in the number of children and adolescents with type 1 diabetes. Last year, in Europe, there were 28.4% of children and adolescents with type 1 diabetes and 21.5% in North America and the Caribbean. The United States, India, and Brazil have the largest incidence and prevalence of children with type 1 diabetes under both age groups below 15 and 20 years old (Figure 1) [6].



**Figure 1.** Estimated number of children and adolescents <20 years with type 1 diabetes by IDF region, 2017 [6].

Type 1 diabetes is described as the most prevalent metabolic disease and the third most common and irreversible chronic disease in childhood, especially below 15 years of age [7]. Despite great progress in medicine, diabetes is an incurable disease, and it is an extraordinary burden on patients and their families. Due to its chronic, progressive, and incurable nature, it greatly affects adolescents, in particular basically their self-esteem, educational opportunities, and lifestyle. Children and adolescents with type 1 diabetes must face many problems related to treatment restrictions.

Measurement of blood sugar is the basic test most often ordered by doctors to detect carbohydrate tolerance disorders and also to diagnose and monitor the treatment of diabetes. Blood is drawn for testing on an empty stomach, followed by a meal or after administration of glucose solution. Serious barriers in the treatment of diabetes among children are problems with painful injections or blood tests, shame about diabetes, arguing with parents about the plan for diabetes control, and compliance. Particularly troublesome are activities related to measuring the level of glucose in the blood, making injections of insulin, exercising, controlling the content of carbohydrate dietary exchanges in the diet, wearing a diabetic or information bracelet, carrying sweets for hypoglycemia, and eating snacks [8].

An additional problem is the fact that the symptoms of diabetes are often ambiguous. They may be confused or attributed to other diseases. Diabetes can only be unequivocally diagnosed when a glucose load test is performed. Too late of a diagnosis of diabetes in childhood can lead to serious

changes, such as destruction of blood vessels, visual disturbances, and problems with the nervous system and kidneys. Very serious diabetes, having been unrecognized for a long time, may endanger children's lives; therefore, extraordinary vigilance should be maintained while observing children, in order to react in time to the first signals of the disease [9,10].

While analyzing the information above, the question arises whether it is possible to diagnose diabetes without performing blood tests. The present work aims to diagnose type 1 diabetes among children based on their physical activity. Selected classification algorithms are compared to obtain the most satisfying results. The promising results encourage developing an application using computational intelligence methods.

## 2. Background

### 2.1. Available Methods of Assessing Physical Activity

Physical activity results in an increase in energy expenditure above resting levels. The rate of energy expenditure is directly linked to the intensity of the activity [11]. Physical activity can be classified according to the Borg scale, ranging from sedentary, light, moderate, to vigorous activities [12].

Currently, there are many methods that allow determining the parameters of physical activity with high accuracy. These include all monitors like pedometers and accelerometers that have motion sensors and are worn on the body of the subject to perform various motion measurements, e.g., step count, the duration of physical activity, and its intensity [13].

### 2.2. Pedometers and Accelerometers in Physical Activity Measuring

The simplest and most popular devices allowing activity measurements are pedometers, which record the number of steps. Thanks to the ability to display the result on a regular basis, they are considered as a motivating tool to perform more physical activity in everyday life. However, measurements by pedometers in scientific research have many limitations. Devices provide information on the frequency of movement, but they do not determine the intensity of physical exercise. Pedometer step counts are also more inaccurate at slow speeds (<60 m/min); therefore, they may be inappropriate for older adults, and the result may not be reliable. Pedometer readings can also vary according to where the pedometer is mounted. In addition, its weakness is also the possibility of falsifying and increasing results by intentionally shaking the device or by shocks caused by driving a car, which do not prove that the subject was more active [14].

Currently, the most accurate motion sensors used to assess physical activity are accelerometers. The devices detect the acceleration of body movement, giving the opportunity to measure reliably the intensity and duration of physical activity, as well as the number of steps taken, and sedentary analysis [15]. Those parameters of the motion are read by the piezoelectric sensor, which converts the analog signal into the digital one in the range (0.1–3.6 Hz). Thanks to this, very accurate monitoring of physical activity is possible. An example of a commonly-used accelerometer is ActiGraph.

### 2.3. ActiGraph Activity Monitor

ActiGraph has been used in large-scale field studies and has become the de facto standard device for objective physical activity monitoring [16]. It is particularly recommended for examination of children and adolescents because it allows for detection of acceleration in three planes of motion, which provides more accurate analysis of the movement relative to pedometers. This is especially important in the case of children's examination because the device records all forms of physical activity, such as doing push-ups or climbing. Many publications describe the advantages of using accelerometers in scientific research, such as objectivity, non-invasiveness, and accuracy, while maintaining the comfort of the user [15].

Published findings related to the application of ActiGraph concerned with exploring differences in daily physical activity profiles among individuals with mild Alzheimer's disease were compared to a control group [17]. Features that can be derived from the accelerometer have been also used to recognize the presence and severity of motor fluctuations in patients with Parkinson's disease [18]. It has been also used with measurements of physical activity to evaluate the effectiveness of surgical and therapy-based interventions in children with cerebral palsy or to derive diurnal rest-activity patterns from actigraphy in adolescents and to analyze associations with adiposity measures and cardiometabolic risk factors [19,20].

However, ActiGraph activity monitors have limited memory and battery capacity to store raw signal data and are additionally quite expensive. One of the current models, ActiGraph wGT3X-BT, currently sells for 225 USD [21]. The costs of devices may vary if bought separately, as compared to bulk orders.

Due to memory limitations, information about movement is read by the accelerometer in the form of the number of pulses (named counts), which are added up in the designated time unit [22]. A count is a unit aimed to be proportional to the average overall acceleration of the human body in a specified period of time. The sum of the received counts is converted into the intensity of physical activity, categorized as sedentary behaviors, light physical activity (LPA), moderate PA (MPA), and vigorous PA (VPA).

There are commonly-used regression equations named as cut points for the ActiGraph accelerometers in predicting energy expenditure (EE) in children and adolescents [23]. The cut points are derived as a part of published research aimed at quantifying activity levels using ActiGraph products. All cut point sets are scaled to 60-s epochs.

In this study, the parameters of physical activity are calculated according to the Freedson Children (2005) model. Definitions of the cut point levels for this model are given in Table 1.

**Table 1.** Freedson Children ActiGraph cut points.

Activity Label	Cut Point	
	From	To
Sedentary	0	149
Light	150	499
Moderate	500	3999
Vigorous	4000	7599
Very Vigorous	7600	$\infty$

#### 2.4. Methods to Compare New and Traditional Accelerometer Data

There are many publications describing how to convert a raw accelerometer signal into the output data of the ActiGraph [16,24,25]. Such data can be obtained using a common smartphone, which is equipped with an accelerometer and a pedometer. Mapping the conversion of counts will allow performing tests in an inexpensive and easy way, which will be comparable to those obtained using the ActiGraph activity monitor.

The research literature describes that counts are calculated as the area under the filtered and rectified (non-negative) curve. The ratio between raw acceleration signal and counts is likely to be brand specific [16]. The experiment described in the literature showed that a third-order Butterworth filter resulted in the highest correlation between ActiGraph counts and unscaled raw accelerometer counts ( $r = 0.975$ ,  $p < 0.01$ ) [24].

The complete method of the conversion of raw accelerometer data to the output the ActiGraph signal is presented below as steps. First, it is necessary to gather 60 s of analog accelerometer reads and calculate the Euclidean distance on analog data in order to create one signal from three axes. Second, this signal should be processed using a third-order Butterworth filter. Next, the area under the filtered

and rectified signal should be calculated. Then, the result should be labeled by type of activity (i.e., sedentary, vigorous, etc.) using predefined cut points and a count of the selected incremented label. All steps should be repeated until enough data are collected.

This method allows for consistency with traditional physical activity measurements so that it is possible to make a historical analysis and comparisons.

### 3. Materials and Methods

#### 3.1. Data Source

The dataset was collected from a group of schoolchildren between the ages of 6 and 18 being under the care of the diabetic clinic for children at Rzeszow State Hospital in Poland in 2016 by E.Czeczek-Lewandowska as a part of her Ph.D. thesis research [8]. The dataset was divided into two groups based on the results of HbA1c glycated hemoglobin tests for diabetes that were read from the patient's medical records provided by the diabetic clinic with parental consent. The analysis included the last two results from the maximum period of one year prior to the study, on the basis of which the arithmetic mean was calculated.

Of the 451 children that took part in the research, the inclusion and exclusion criteria were extracted and analyzed. The eligibility criteria that were applied were: ages between 6 and 18, type 1 diabetes diagnosed a minimum of one year prior to the examination, HbA1c values determined at least twice in the year prior to the start of the study, informing parents about the study and child consent, required physical activity record length (excluding night hours and activities performed in contact with water), and training the parent and child in terms of using the accelerometer. Children who did not meet the inclusion criteria, were diagnosed with type 2 diabetes or other metabolic disorders, had current complications in the course of diabetes, and became sick during the study period were excluded from the study. Additional excluding criteria were exceptionally bad weather conditions, a period of holidays, and holiday break during the study period. Finally, the study group consisted of 215 children with type 1 diabetes and 115 healthy children from a control group. Nine parameters for each child were collected and are listed below.

1. General and BMI parameters:
  - Age
  - Sex
  - Weight
  - Height
2. Physical activity parameters (per week):
  - Step count
  - Sedentary activity minutes
  - Light activity minutes
  - Moderate activity minutes
  - Vigorous activity minutes
3. Type 1 diabetes presence (binary parameter)

The weight and height of the body were obtained using a Radwag WPT 60/150 OW electronic scale during a three-stage measurement. The level of physical activity was assessed with a hip-worn ActiGraph wGT3X-BT activity monitor used by the children 12 h a day for a week, excluding night time and activities performed in contact with water, i.e., bath, swimming pool. The parameters of physical activity were calculated according to the Freedson Children (2005) method.

### 3.2. Classification Methods

Classification systems have an important role in decision-making tasks by categorizing the available information based on some criteria [26]. The purpose of this research was to assess the relative efficacy of some well-known classification methods. We have considered classification techniques that are based on statistical and AI techniques. A brief review of the relevant classification methods is presented in this section.

#### 3.2.1. Support Vector Machine

Support vector machine (SVM) is a classification algorithm used for finding an optimal hyperplane that maximizes the margin between classes. That hyperplane is orientated in such a way that it is as far as possible from the closest data points from each of the classes. These closest points are called support vectors [27]. The key element of the SVM algorithm is the kernel function. It transforms a non-linear feature space into a linear one before the hyperplane search [28].

#### 3.2.2. Probabilistic Neural Network

The probabilistic neural network (PNN) is a feedforward neural network model. It consists of input, pattern, summation, and output layers. The input layer is represented by the features of the input vector. The pattern layer is composed of as many neurons as learning samples. The summation layer consists of  $n$  neurons where each of them computes the signal only for patterns that belong to the  $n^{\text{th}}$  class. The output layer is used to yield the decision; its result with the largest probability value is 1, and the rest of the outputs are 0 [29].

#### 3.2.3. Multilayer Perceptron

Multilayer perceptron (MLP) is a feedforward artificial neural network that uses the backpropagation technique for training. It is composed of one or more layers of neurons. Data are transferred to the input layer; there may be one or more hidden layers; and predictions are made on the output layer [30].

#### 3.2.4. Group Method of Data Handling

The group method of data handling (GMDH) is a family of inductive algorithms of multi-parametric datasets. It features the fully-automatic parametric and structural optimization of models. GMDH is used for constructing a high-order regression-type polynomial [31].

#### 3.2.5. Gene Expression Programming

Gene expression programming (GEP) is an evolutionary algorithm that creates models, equations, or computer programs. GEP programs are encoded in the so-called chromosomes, which are mutated by computing the expression of each chromosome. Next, the predefined genetic operators are applied, and the fitness is calculated. Finally, the best chromosomes are selected to reproduce [32].

#### 3.2.6. Linear Regression

Linear regression is one of the simplest and best known algorithms in statistics and machine learning used for finding a linear relationship between the target and one or more predictors. The core idea of linear regression is to obtain a line that best fits the data [33].

#### 3.2.7. Radial Basis Function Network

The radial basis function network (RBF) is an artificial neural network that uses radial basis functions as activation functions. The output of the RBF network is composed of neuron parameters and radial basis functions of the inputs [34].



### 3.2.8. Logistic Regression

Logistic regression is a statistical method for analyzing a dataset with one or more independent variables that determine an outcome. The goal of logistic regression is to find the best fitting model to describe the relationship between the binary dependent variable and a set of independent variables [35].

### 3.2.9. Decision Tree

The decision tree (DT) is a type of model used for both classification and regression. Trees answer sequential questions, which are sent down a certain route of the tree given the answer. They are intuitive and provide one of the simplest portrayals for classification purposes. Tree depth represents how many questions are asked before reaching the predicted classification [36].

### 3.2.10. Random Forests

Random forests (RF) are a classification algorithm that is a combination of decision tree predictors so that each of them depends on the values of a randomly -elected independent vector with the same distribution for all trees in the forest [37]. After training, predictions for unseen samples can be made by taking the majority vote [36].

## 3.3. Validation Methods

Commonly-used evaluation measures are precision, sensitivity, and accuracy. These measures can be defined with the help of four cardinalities of the confusion matrix, namely the truth positive (TP), true negative (TN), false positive (FP), and false negative (FN) [38].

### 3.3.1. Accuracy

The accuracy metric measures the total number of correct classifications (true positives and true negatives) [38].

$$ACC_i = \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}, TP_i + TN_i + FP_i + FN_i > 0 \quad (1)$$

### 3.3.2. Sensitivity

The sensitivity (recall) measures the proportion of actual positives that are correctly identified as such (e.g., the percentage of children with type 1 diabetes who are correctly identified as having the condition): [38].

$$SE_i = \frac{TP_i}{TP_i + FN_i}, TP_i + FN_i > 0 \quad (2)$$

### 3.3.3. Specificity

The specificity measures the proportion of actual negatives that are correctly identified as such (e.g., the percentage of healthy children who are correctly identified as not having the condition): [38].

$$SP_i = \frac{TN_i}{TN_i + FP_i}, TN_i + FP_i > 0 \quad (3)$$

### 3.3.4. Precision

The precision metrics determine the quality of positive predictions (true positives and false positives): [38].

$$PPV_i = \frac{TP_i}{TP_i + FP_i}, TP_i + FP_i > 0 \quad (4)$$

### 3.3.5. AUC

For a binary classification problem, the evaluation of the performance is typically illustrated with the receiver operating characteristic (ROC) curve, which plots the true positives versus the false positive rate at various threshold settings. It is convenient to reduce it to a single scalar value representing expected performance. A common method is to calculate the area under the ROC curve (AUC). An ideal classifier achieves an AUC equal to 1, while the classifier that makes a random decision achieves an AUC equal to 0.5 [38,39].

### 3.3.6. Goodness Index

The goodness index (G) represents the Euclidean distance between the evaluated point in the receiver operating characteristic space and the point (0,1), which represents the perfect classifier that classifies all positive cases and negative cases correctly.

$$G_i = \sqrt{\left(1 - \left(\frac{TP_i}{TP_i + FN_i}\right)^2\right) + \left(1 - \left(\frac{TN_i}{FP_i + TN_i}\right)^2\right)} \quad (5)$$

G can assume values between 0 and  $\sqrt{2}$ , and a classifier can be considered as:

- optimum, when  $G \leq 0.25$ ,
- good, when  $0.25 < G < 0.70$ ,
- random, if  $G = 0.70$ ,
- bad, if  $G > 0.70$  [40].

The G value result analysis allows evaluating the best-performing classifier [28].

## 3.4. Other Data Analysis Methods

### 3.4.1. Clustering Method

The  $k$ -means clustering algorithm is one of the most popular clustering algorithms, which is used to find groups that are not explicitly labeled in the data. It uses iterative refinement to produce a final result. The inputs of the algorithm are the dataset and the number of clusters  $k$ . A cluster is a collection of data points that have been aggregated together because of certain similarities, and the dataset is a collection of features for each data point. The algorithms start with initial estimates for the  $k$  centroids, which can either be randomly initialized or randomly selected from the dataset. Then, the algorithm iterates between two steps:

- Data assignment: each data point is assigned to its nearest centroid, based on the squared Euclidean distance. If  $c_i$  is the collection of centroids in set  $C$ , then each data point  $x$  is assigned to a cluster based on:

$$\arg \min_{c_i \in C} \text{dist}(c_i, x)^2 \quad (6)$$

where  $\text{dist}(\cdot)$  is the standard Euclidean distance.  $S_i$  is the set of data point assignments for each  $i^{\text{th}}$  cluster centroid.

- Centroid update: centroids are recomputed by taking the mean of all data points assigned to that centroid's cluster.

$$c_i = \frac{1}{|S_i|} \sum_{x_i \in S_i} x_i \quad (7)$$

The algorithm iterates between those two steps until convergence. Convergence is reached when the computed centroids do not change or the centroids and the assigned points oscillate back and forth from one iteration to the next one. The result may be a local optimum, so assessing more than one run of the algorithm with randomized starting centroids may give a better outcome [41].

### 3.4.2. Feature Selection Methods

Feature selection is the first and fundamental step in data analysis. This is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. Feature selection methods aid in creating an accurate predictive model by choosing only features that are relevant. Irrelevant features in the dataset can decrease the performance of the models; redundant data can allow a greater opportunity to make decisions based on noise and increase algorithm complexity, while algorithms are trained more slowly.

There are three general classes of feature selection algorithms: filter methods, wrapper methods, and embedded methods. Filter feature selection methods apply a statistical measure to assign a score to each feature. The features are ranked by the score and are either selected to be kept or removed from the dataset. The methods are often univariate and consider the feature independently, or with regard to the dependent variable. Examples of some filter methods include correlation coefficient scores and information gain. These methods are used to create the feature ranking [42].

Pearson's correlation coefficient is one of the methods of measuring the association between variables of interest, and it is based on the covariance method. It gives information about the magnitude of the association, or correlation, as well as the direction of the relationship [43].

Entropy measures the amount of uncertainty in the dataset. The information gain is based on the decrease in entropy after splitting a dataset on an attribute [44]. It is used to generate a decision tree from a dataset. Constructing a decision tree comes down to finding an attribute that returns the highest information gain.

The information gain  $IG$  is the change in information entropy  $H$  from a prior state to a state that takes some information as given:

$$IG(d | a) = H(d) - H(d | a) \quad (8)$$

where  $H(d | a)$  is the conditional entropy of decision  $d$  given attribute  $a$  and  $H(d)$  is the entropy of decision  $d$ , which is equal to:

$$H(d) = - \sum_{i=1}^k p(d_i) \cdot \ln p(d_i) \quad (9)$$

Information gain can be calculated for each remaining attribute. The attribute with the largest information gain is used to split the dataset on this iteration [45]:

$$IG(d | a) = H(d) - \sum_{j=1}^l p(a_j) \cdot H(d | a_j) \quad (10)$$

## 4. Results

### 4.1. Data Analysis Results

The aim of this research was to answer whether type 1 diabetes among children and adolescents can be diagnosed based on physical activity. We defined the prediction problem of type 1 diabetes presence among children as a binary classification problem. The results were obtained using DTREG, Weka, and Python Scikit-Learn software packages [46–48].

The assessment of physical activity impact on the prevalence of type 1 diabetes among children and adolescents was based on parameters closely related to the intensity of physical activity. These parameters were calculated according to the Freedson Children (2005) model (Table 1). We considered the dataset consisting of parameter values of 215 sick and 115 healthy children. The selected classification parameters set was composed of the total number of steps and sedentary, light, moderate, and vigorous activity minutes per week.

Subsequently, we decided to create a feature ranking (FR) automatically. FR specifies the significance of features for a problem by ranking features according to their importance in the model using ranking algorithms [42]. The FR based on correlation coefficient scores was performed, and the results are presented in Table 2.

**Table 2.** Correlation coefficient feature ranking.

	<b>Feature Name</b>	<b>Score</b>
1	step count	0.2362
2	vigorous activity minutes	0.0505
3	moderate activity minutes	0.0469
4	sedentary activity minutes	0.0408
5	light activity minutes	0.0127

Due to the fact that evaluating the entropy is a key step in the decision tree algorithm, it was used to calculate the homogeneity of a sample, and we decided to create FR based on information gain, which is based on the entropy. The results are presented in Table 3.

**Table 3.** Information gain feature ranking.

	<b>Feature Name</b>	<b>Score</b>
1	vigorous activity minutes	0.1435
2	moderate activity minutes	0.1375
3	step count	0.084
...	...	0

The data presented in Table 2 showed that the most significant parameter was the step count (per week). Data presented in Table 3 resulted in three important parameters, i.e., vigorous activity minutes, moderate activity minutes, and step count.

The presented results of the FR are purely illustrative, because threshold values were set to exclude unimportant parameters. We decided to use the classification of all physical activity parameters.

#### 4.2. Classification Result

Firstly, we built a decision tree with an overall goal to extract general information from a dataset and transform that information into a structure that can be understood by an ordinary user. A decision tree was built from physical activity parameters, i.e., the total number of steps and the groups of sedentary, light, moderate and vigorous activities, using the implementation of the c4.5 algorithm, called J48, from the Weka software package. The algorithm was started with default values, such as the confidence threshold for pruning the set to 0.25 and a minimum number of instances per leaf equal to two.

At each node of the tree, the algorithm chose the attribute of the data that most effectively split the set of samples into subsets enriched in one class or the other. The splitting criterion was the normalized information gain. The information gain feature ranking results described in Section 4.1 had the vigorous activity minutes parameter in the first place. Hence, it could be concluded that the root of the decision tree would be the same parameter. In Figure 2, as can be observed, the results of the decision tree classification are not completely consistent with logical thinking, and in some cases, they seem contradictory.

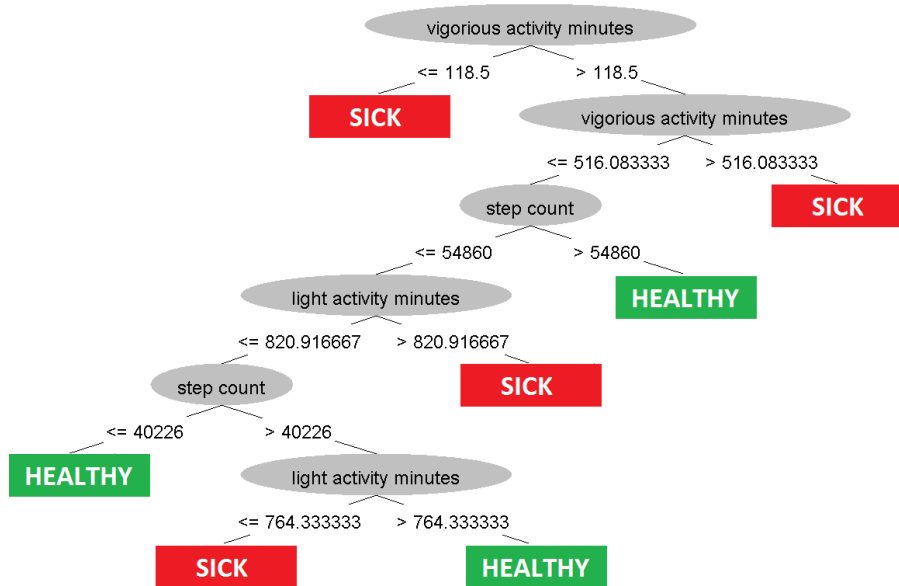


Figure 2. J48 tree design.

Accurate and reliable information is vital for effective decision making. Thus, we employed an undersampling technique to obtain reliable estimates. It is a technique used to adjust the class distribution of a dataset. For this purpose, 115 of the 215 sick children were chosen by the random selection process to obtain two equivalent ratios of sick and healthy patient classes. After undersampling, the remaining 230 patients were considered eligible and were enrolled in the study.

The best results in the prediction of type 1 diabetes presence among children and adolescents were obtained with decision tree forests. This model enabled the prediction with the highest accuracy (86.09%), specificity (84.35%), and precision (84.87%). The PNN also showed high accuracy (84.35%) and the highest sensitivity (89.57%), but markedly lower specificity (79.13%) and precision (81.10%). The AUC for PNN (0.926578) also exceeded the values of this parameter for the remaining classifiers (Figure 1). The averaged accuracy, sensitivity, specificity, precision, goodness index, and AUC value obtained for all applied computational intelligence methods and a linear regression model are presented in Table 4.

**Table 4.** The accuracy, sensitivity, specificity, precision, goodness index, and the area under the receiver operating characteristic curve obtained for the set of physical activity variables.

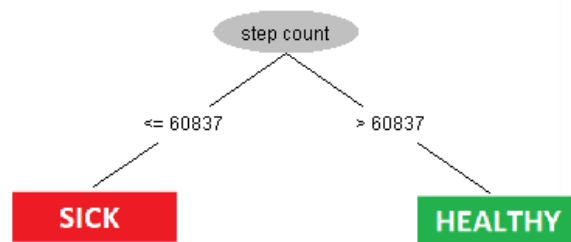
Algorithm Name	Acc(%)	Sen(%)	Spe(%)	Prec(%)	G	AUC
<b>Decision Tree Forest</b>	86.09	87.83	84.35	84.87	0.1983	-
<b>PNN</b>	84.35	89.57	79.13	81.10	0.2333	0.926578
<b>SVM</b>	84.35	86.96	81.74	82.64	0.2244	0.909716
<b>Single tree</b>	83.48	86.09	80.87	81.82	0.2365	-
<b>GEP</b>	83.04	83.48	82.61	82.76	0.2399	0.830435
<b>Logistic regression</b>	82.61	84.35	80.87	81.51	0.2472	0.883478
<b>GMDH</b>	82.61	82.61	82.61	82.61	0.2460	0.905482
<b>RBF network</b>	82.17	85.22	79.13	80.33	0.2557	0.905331
<b>MLP</b>	81.30	86.09	76.52	78.57	0.2729	0.897921
<b>Linear regression</b>	80.87	85.22	76.52	78.40	0.2774	0.884008

The values were obtained using a 10-fold cross-validation procedure [49]. The given dataset consisting of 230 samples was split into 10 folds, where each fold was used as a testing set at some point. In the first iteration, the first fold was used to test the model, and the rest were used to train the model. In the second iteration, the second fold was used as the testing set, and the rest served as the training set. This process was repeated until each fold of the 10 folds had been used as the testing set. Based on the obtained scores in every iteration, the mean value was calculated in order to assess the performance of the model.

#### 4.3. Clustering Result

In the last step, we wish to explain the correlation between physical activity parameter values and type 1 diabetes presence. This relied on finding the equation that played a major role in the correct diabetes classification among children and adolescents. For this purpose, we assumed that we did not have a classification into sick and healthy patients and used the k-means clustering algorithm.

After clustering, we compared the obtained clusters with their corresponding classes from the dataset. It turned out that 215 of 330 records had identical classes. Then, we built a decision tree for the remaining 215 records using the c4.5 algorithm with the same setup as described in the Classification Result section. The result of the decision tree is presented in Figure 3.



**Figure 3.** Classification results after clustering.

The obtained results confirmed the assumption that the correlation between physical activity and type 1 diabetes presence can be evaluated based on measuring step count. It is possible to predict the prevalence of the disease correctly at least in 65% of the cases. As a result, a child was determined to be sick when performing fewer than 60,837 steps per week.

## 5. Discussion

The purpose of this research was to find a relationship between the intensity of physical activity and the presence of type 1 diabetes among children and develop a non-invasive method of type 1 diabetes detection. Assessment of the physical activity was based on ActiGraph activity monitor measurements. The ActiGraph measurements for health-related research were also carried out in published findings [17–20].

Decision tree forests, as well as other computational intelligence methods were applied for the detection of different diseases, e.g., breast cancer and heart disease [50,51]. Application of decision tree forests, which included five parameters connected with the intensity of physical activity, enabled the prediction of type 1 diabetes presence among children and adolescents between the ages of six and 18 with a high accuracy of 86.09% and specificity of 84.35%. The PNN also showed a high accuracy of 84.35%. Our results were comparable to similar articles, in which neural networks were used for outcome prediction of diabetes presence. For example, an SVM algorithm using the RBF kernel, the same as was used in this paper, was able to predict the presence of elevated blood glucose level via electrochemical measurement of saliva with approximately 85% accuracy [52].

As the final result of the study, it was concluded that if the number of steps is lower than around 61,000 a week, it is likely that the child is suffering from type 1 diabetes. After dividing this by the seven days of a week, we obtained the average number of steps per day, which was around 9000, but it should be noted that gender and age were not included in the calculation of this result. The updated international literature indicates that we can expect, among children, boys to average 12,000–16,000 steps/day, girls to average 10,000–13,000 steps per day, and adolescents to reach approximately 8000–9000 steps/day [53]. Thus, the obtained result was consistent with the normative international literature.

Decreased physical activity of ill children compared to healthy peers was the result of the disease. Many children also complained that it was difficult for them to go through a pitch of more than 100 m, that it was difficult for them to run, play sports, exercise, lift something heavy, take a bath, or shower by themselves, and that they felt pain and were tired.

The results of the research are promising and encourage developing a mobile application for type 1 diabetes diagnosis dedicated to children and adolescents. Although the popularity of using mobile phones applications in various health disorders has reached about 30%, it should be taken into account that young people are more likely to use and more effective at using new mobile phone applications, and the popularity and potential acceptance of mobile health solutions have an increasing tendency [54].

**Author Contributions:** Conceptualization, A.C., S.C., and D.M.; data curation, S.C.; formal analysis, S.C.; funding acquisition, D.M.; investigation, S.C.; methodology, A.C. and S.C.; project administration, A.C.; software, S.C.; supervision, D.M.; validation, D.M.; visualization, S.C.; writing, original draft, A.C.; writing, review and editing, A.C., S.C., and D.M.

**Funding:** This project is financed by the Minister of Science and Higher Education of the Republic of Poland within the “Regional Initiative of Excellence” program for years 2019–2022; Project Number 027/RID/2018/19; the amount granted: 11,999,900 PLN.

**Acknowledgments:** There are no acknowledgments related to this study.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

ADA	American Diabetes Association
AUC	Area under the receiver operating characteristic curve
BMI	Body mass index
DT	Decision tree
EE	energy expenditure
FN	False negative
FP	False positive
FR	Feature ranking
G	Goodness index
GEP	Gene expression programming
GMDH	Group method of data handling
LPA	light physical activity
MLP	Multilayer perceptron
MPA	moderate physical activity
PNN	Probabilistic neural network
RBF	Radial basis function
RF	Random forest
SVM	Support vector machine
TN	True negative
TP	Truth positive
WHO	World Health Organization
VPA	vigorous physical activity

## References

1. American Diabetes Association. Diagnosis and classification of diabetes mellitus. *Diabetes Care* **2009**, *33* (Suppl. 1), 62–67.
2. Tatoń, J.; Czech, A.; Bernas, M. Edukacja terapeutyczna, samokontrola glikemii i psychologia cukrzycy. Terapeutyczny styl życia. In *Diabetologia Kliniczna*; Tatoń, J., Czech, A., Bernas, M., Eds.; PZWL: Warsaw, Poland, 2008; pp. 339–429.
3. World Health Organization. *Global Strategy on Diet, Physical Activity and Health*; WHO Library Cataloguing-in-Publication Data; World Health Organization: Geneva, Switzerland, 2004; pp. 1–18.
4. Iannotti, R.J.; Kalman, M.; Inchley, J.; Tynjälä, J.; Bucksch, J.; The HBSC Physical Activity Focus Group. Social determinants of health and well-being among young people. In *Health Behaviour in School-Aged Children (HBSC) Study: International Report from the 2009/2010 Survey*; Currie, C., Ed.; WHO Regional Office for Europe: Copenhagen, Denmark, 2012; pp. 129–132.
5. Faigenbaum, A. Physical Activity in Children and Adolescents. *ACSM Bull.* **2015**. Available online: <https://www.acsm.org/> (accessed on 16 October 2018).
6. International Diabetes Federation. *IDF Diabetes Atlas. Eighth Edition 2017*; International Diabetes Federation: Brussels, Belgium, 2017; Volume 8, pp. 1–150.
7. Pettitt, D.J.; Talton, J.; Dabelea, D.; Divers, J.; Imperatore, G.; Lawrence, J.M.; Liese, A.D.; Linder, B.; Mayer-Davis, E.J.; Pihoker, C.; et al. Prevalence of Diabetes in U.S. Youth in 2009: The SEARCH for Diabetes in Youth Study. *Diabetes Care* **2014**, *37*, 402–408. [[CrossRef](#)] [[PubMed](#)]
8. Czenczek-Lewandowska, E. Level of Physical Activity in Children and Adolescents with type 1 Diabetes, Relative to the Insulin Therapy Applied. Ph.D. Thesis, University of Rzeszów, Rzeszów, Poland, 2017; pp. 1–165.
9. Czenczek-Lewandowska, E.; Grzegorzczak, J.; Mazur, A. Physical activity in children and adolescents with type 1 diabetes and contemporary methods of its assessment. *Pediatr. Endocrinol. Diabetes Metab.* **2018**, *24*, 179–184. [[CrossRef](#)] [[PubMed](#)]
10. Allen, N.; Gupta, A. Current Diabetes Technology: Striving for the Artificial Pancreas. *Diagnostics* **2019**, *9*, 31. [[CrossRef](#)] [[PubMed](#)]
11. Strath, S.J.; Kaminsky, L.A.; Ainsworth, B.E.; Ekelund, U.; Freedson, P.S.; Gary, R.A.; Richardson, C.R.; Smith, D.T.; Swartz, A.M. Guide to the Assessment of Physical Activity: Clinical and Research Applications A Scientific Statement From the American Heart Association. *Circulation* **2013**, *128*, 2259–2279. [[CrossRef](#)]



12. Katch, V.L.; McArdle, W.D.; Katch, F.I. Energy expenditure during rest and physical activity. In *Essentials of Exercise Physiology*, 4th ed.; McArdle, W.D., Katch, F.I., Katch, V.L., Eds.; Lippincott Williams & Wilkins: Baltimore, MD, USA, 2011; pp. 237–262.
13. Sylvia, L.G.; Bernstein, E.E.; Hubbard, J.L. Practical Guide to Measuring Physical Activity. *J. Acad. Nutr. Diet.* **2014**, *114*, 199–208. [[CrossRef](#)]
14. Hills, A.P.; Mokhtar, N.; Byrne, N.M. Assessment of Physical Activity and Energy Expenditure: An Overview of Objective Measures. *Front. Nutr.* **2014**, *1*, 1–14. [[CrossRef](#)]
15. Tanaka, C.; Hikiyama, Y.; Ando, T.; Oshima, Y.; Usui, C.; Ohgi, Y.; Kaneda, K.; Tanaka, S. Prediction of Physical Activity Intensity with Accelerometry in Young Children. *Int. J. Environ. Res. Public Health* **2019**, *16*, 931. [[CrossRef](#)]
16. van Hees, V.T.; Pias, M.; Taherian, S.; Ekelund, U.; Brage, S. A method to compare new and traditional accelerometry data in physical activity monitoring. In Proceedings of the 2010 IEEE International Symposium on “A World of Wireless, Mobile and Multimedia Networks”, Montreal, QC, Canada, 14–17 June 2010; pp. 1–6.
17. Vijay, R.; Watts, A.; Watts, V. Daily Physical Activity Patterns During the Early Stage of Alzheimer’s Disease. *J. Alzheimer’s Dis.* **2016**, *55*, 659–667.
18. Bonato, P.; Sherrill, D.M.; Standaert, D.G.; Salles, S.S.; Akay, M. Data mining techniques to detect motor fluctuations in Parkinson’s disease. *Conf. Proc. IEEE Eng. Med. Biol. Soc.* **2004**, *7*, 4766–4769. [[PubMed](#)]
19. Ahmadi, M.; O’Neil, M.; Fragala-Pinkham, M.; Lennon, N.; Trost, S. Machine learning algorithms for activity recognition in ambulant children and adolescents with cerebral palsy. *J. Neuroeng. Rehabil.* **2018**, *15*, 105. [[CrossRef](#)] [[PubMed](#)]
20. Quante, M.; Cespedes Feliciano, E.M.; Rifas-Shiman, S.L.; Mariani, S.; Kaplan, E.R.; Rueschman, M.; Oken, E.; Taveras, E.M.; Redline, S. Association of Daily Rest-Activity Patterns With Adiposity and Cardiometabolic Risk Measures in Teens. *J. Adolesc. Health* **2019**. [[CrossRef](#)] [[PubMed](#)]
21. Kanna, K.R.; Sugumaran, V.; Vijayaram, T.R.; Karthikeyan, C.P. Activities of Daily Life (ADL) Recognition using Wrist-worn Accelerometer. *Int. J. Eng. Technol. (IJET)* **2016**, *8*, 1406–1413.
22. Welk, G.J. Use of accelerometry-based activity monitors to assess physical activity. In *Physical Activity Assessments for Health-Related Research*; Welk, G.J., Ed.; Human Kinetics Publishers: Champaign, IL, USA, 2002; pp. 125–142.
23. Crouter, S.E.; Horton, M.; Bassett, D.R. Validity of ActiGraph Child-Specific Equations during Various Physical Activities. *Med. Sci. Sports Exerc.* **2013**, *45*, 1403–1409. [[CrossRef](#)] [[PubMed](#)]
24. Hekler, E.B.; Buman, M.P.; Grieco, L.; Rosenberger, M.; Winter, S.J.; Haskell, W.; King, A.C. Validation of Physical Activity Tracking via Android Smartphones Compared to ActiGraph Accelerometer: Laboratory-Based and Free-Living Validation Studies. *JMIR MHealth UHealth* **2015**, *3*, e36. [[CrossRef](#)] [[PubMed](#)]
25. Migueles, J.H.; Cadenas-Sanchez, C.; Ekelund, U.; Delisle Nyström, C.; Mora-Gonzalez, J.; Löf, M.; Labayen, I.; Ruiz, J.R.; Ortega, F.B. Accelerometer Data Collection and Processing Criteria to Assess Physical Activity and Other Outcomes: A Systematic Review and Practical Considerations. *Sports Med.* **2017**, *47*, 1821–1845. [[CrossRef](#)] [[PubMed](#)]
26. Jacob, E. Classification and Categorization: A Difference that Makes a Difference. *Libr. Trends* **2004**, *52*, 515–540.
27. Huang, S.; Cai, N.; Pacheco, P.P.; Narrandes, S.; Wang, Y.; Xu, W. Applications of Support Vector Machine(SVM) Learning in Cancer Genomics. *Cancer Genom. Proteom.* **2018**, *15*, 41–51.
28. Taborri, J.; Palermo, E.; Rossi, S. Automatic Detection of Faults in Race Walking: A Comparative Analysis of Machine-Learning Algorithms Fed with Inertial Sensor Data. *Sensors* **2019**, *19*, 1461. [[CrossRef](#)]
29. Sun, Q.; Lin, F.; Yan, W.; Wang, F.; Chen, S.; Zhong, L. Estimation of the Hydrophobicity of a Composite Insulator Based on an Improved Probabilistic Neural Network. *Energies* **2018**, *11*, 2459. [[CrossRef](#)]
30. Nazzal, J.M.; El-Emary, I.M.; Najim, S.A. Multilayer Perceptron Neural Network (MLPs) For Analyzing the Properties of Jordan Oil Shale. *World Appl. Sci. J.* **2008**, *5*, 546–552.
31. Li, R.Y.M.; Fong, S.; Chong, W.S. Forecasting the REITs and stock indices: Group Method of Data Handling Neural Network approach. *Pac. Rim Prop. Res. J.* **2017**, *23*, 1–38. [[CrossRef](#)]
32. Ferreira, C. The Basic Gene Expression Algorithm. In *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence*; Springer: Berlin, Germany, 2006; pp. 55–120.

33. Godfrey, K. Simple Linear Regression in Medical Research. *N. Engl. J. Med.* **1985**, *313*, 1629–1636. [[CrossRef](#)] [[PubMed](#)]
34. Acosta, F.M.A. Radial basis function and related models: An overview. *Signal Process.* **1995**, *45*, 37–58. [[CrossRef](#)]
35. Peng, C.Y.J.; Lee, K.L.; Ingersoll, G.M. An Introduction to Logistic Regression Analysis and Reporting. *J. Educ. Res.* **2002**, *96*, 3–14. [[CrossRef](#)]
36. James, G.; Witten, D.; Hastie, T.; Tibshirani, R. Tree-Based Methods. In *An Introduction to Statistical Learning with Applications in R*; Springer: New York, NY, USA, 2017; pp. 303–336.
37. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
38. Fawcett, T. An introduction to ROC analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874. [[CrossRef](#)]
39. Brisimi, T.; Xu, T.; Wang, T.; Dai, W.; Adams, W.; Paschalidis, I. Predicting Chronic Disease Hospitalizations from Electronic Health Records: An Interpretable Classification Approach. *Proc. IEEE* **2018**, *106*, 690–707. [[CrossRef](#)]
40. Taborri, J.; Scalona, E.; Palermo, E.; Rossi, S.; Cappa, P. Validation of Inter-Subject Training for Hidden Markov Models Applied to Gait Phase Detection in Children with Cerebral Palsy. *Sensors* **2015**, *15*, 24514–24529. [[CrossRef](#)]
41. Wilkin, G.A.; Huang, X. K-Means Clustering Algorithms: Implementation and Comparison. In *Second International Multi-Symposiums on Computer and Computational Sciences (IMSCCS 2007)*; IEEE: Iowa City, IA, USA, 2007.
42. Cilia, N.; De Stefano, C.; Fontanella, F.; Raimondo, S.; di Freca, A.S. An Experimental Comparison of Feature-Selection and Classification Methods for Microarray Datasets. *Information* **2019**, *10*, 109. [[CrossRef](#)]
43. Rodgers, J.L.; Nicewander, W.A. Thirteen Ways to Look at the Correlation Coefficient. *Am. Stat.* **1988**, *42*, 59–66. [[CrossRef](#)]
44. Robert, C. An entropy concentration theorem: Applications in artificial intelligence and descriptive statistics. *J. Appl. Probab.* **1990**, *27*, 303–313. [[CrossRef](#)]
45. Quinlan, J.R. Induction of Decision Trees. *Mach. Learn.* **1986**, *1*, 81–106. [[CrossRef](#)]
46. Sherrod, P. DTREG Predictive Modeling Software. 2003. Available online: [www.dtrek.com](http://www.dtrek.com) (accessed on 12 February 2019).
47. Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I.H. The WEKA data mining software. *ACM SIGKDD Explor. Newsl.* **2009**, *11*, 10–18. [[CrossRef](#)]
48. Michel, V.; Gramfort, A.; Varoquaux, G.; Eger, E.; Keribin, C.; Thirion, B. A supervised clustering approach for fMRI-based inference of brain states. *Pattern Recognit.* **2012**, *45*, 2041–2049. [[CrossRef](#)]
49. Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proc. Int. Jt. Conf. Artif. Intell.* **1995**, *14*, 1137–1143.
50. Übeyli, E.D. Implementing automated diagnostic systems for breast cancer detection. *Expert Syst. Appl.* **2007**, *33*, 1054–1062. [[CrossRef](#)]
51. Nahar, J.; Imam, T.; Tickle, K.S.; Chen, Y.P. Computational intelligence for heart disease diagnosis: A medical knowledge driven approach. *Expert Syst. Appl.* **2013**, *40*, 96–104. [[CrossRef](#)]
52. Malik, S.; Khadgawat, R.; Anand, S.; Gupta, S. Non-invasive detection of fasting blood glucose level via electrochemical measurement of saliva. *SpringerPlus* **2016**, *5*, 701. [[CrossRef](#)]
53. Tudor-Locke, C.; Craig, C.L.; Beets M.W.; Belton, S.; Cardon, G.M.; Duncan, S.; Hatano, Y.; Lubans, D.R.; Olds, T.S.; Raustorp, A.; et al. How many steps/day are enough? for children and adolescents. *Int. J. Behav. Nutr. Phys. Act.* **2011**, *8*, 78. [[CrossRef](#)] [[PubMed](#)]
54. Cerna, L.; Maresova, P. Patients' attitudes to the use of modern technologies in the treatment of diabetes. *Patient Prefer Adherence* **2016**, *10*, 1869–1879. [[PubMed](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

# Automatic Detection and Counting of Blood Cells in Smear Images Using RetinaNet

Grzegorz Drałus <sup>†</sup>, Damian Mazur <sup>†</sup> and Anna Czmił <sup>\*</sup>

Department of Electrical and Computer Engineering Fundamentals, Rzeszow University of Technology, 35-959 Rzeszow, Poland; gregor@prz.edu.pl (G.D.); mazur@prz.edu.pl (D.M.)

\* Correspondence: czmilanna@gmail.com

† These authors contributed equally to this work.

**Abstract:** A complete blood count is one of the significant clinical tests that evaluates overall human health and provides relevant information for disease diagnosis. The conventional strategies of blood cell counting include manual counting as well as counting using the hemocytometer and are tedious and time-consuming tasks. This research-based paper proposes an automatic software-based alternative method to count blood cells accurately using the RetinaNet deep learning network, which is used to recognize and classify objects in microscopic images. After training, the network automatically recognizes and counts red blood cells, white blood cells, and platelets. We tested a model trained on smear images and found that the trained model has generalized capabilities. We assessed the quality of detection and cell counting using performance measures, such as accuracy, sensitivity, precision, and F1-score. Moreover, we studied the dependence of the confidence thresholds and the number of learning epochs on the obtained results of recognition and counting. We compared the performance of the proposed approach with those obtained by other authors who dealt with the subject of cell counting and show that object detection and labeling can be an additional advantage in the task of counting objects.



**Citation:** Drałus, G.; Mazur, D.; Czmił, A. Automatic Detection and Counting of Blood Cells in Smear Images Using RetinaNet. *Entropy* **2021**, *23*, 1522. <https://doi.org/10.3390/e23111522>

Academic Editor: Anton Civit

Received: 10 September 2021

Accepted: 11 November 2021

Published: 16 November 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** confidence threshold; convolution neural networks; platelet; RBC; WBC

## 1. Introduction

A complete blood count (CBC) is a typical clinical test that provides relevant information for disease diagnosis. The main three types of blood cells are: Red Blood Cells (RBCs), also called erythrocytes, White Blood Cells (WBCs), also called leukocytes, and platelets, also called thrombocytes. CBC provides information about the production of all blood cells, identifies the patient's ability to carry oxygen by evaluating RBC counts, and allows for immune system evaluation by assessing WBC counts with differential. This test helps diagnose anemia, certain cancers, infections, and many many others, as well as monitor the side effects of certain medications [1]. For this reason, medical laboratories are flooded with a large number of blood and tissue samples that need to be analyzed as accurately as possible and in the shortest possible time. The ability to accurately quantitate specific populations of cells is important for precision diagnostics in laboratory medicine. Thus, medical staff work under heavy loads and time pressure. Medical workers often have to work overtime to analyze all samples on time, causing even greater fatigue of the staff, which may result in mistakes and lower work efficiency [2]. These errors may lead to severe and even fatal consequences in the treatment of patients.

An alternative to traditional manual counting of various cells by specialists are semi-automatic and automatic methods. Automatic detection and counting of cells in images is a difficult and complex task, especially in reality the resolution of input medical images could be very high, at the same time the target cells could easily be extremely dense. Moreover, there are a large number of them in the image, the cells are often overlapped and

there are problems with distinguishing cells. This is the principal motivation of automatic cell counting.

There are generally two main approaches in the automated counting of blood cells. We can distinguish traditional methods, which involve several steps such as preprocessing, segmentation, feature extraction, and classification, while other methods are based on deep neural networks (DNN). The selected traditional automatic RBCs counting methods are presented in [3,4]. Various methods of automatic WBC counting are presented in [5–10]. Despite the numerous advantages of the automated methods, they also have disadvantages, such as the accuracy of counting and the preparation of cell images. Reliable and accurate cell detection is usually a difficult problem due to a great variability of cells and the complexity of data. Detection can determine the presence of a specific cell in a microscopic image, e.g., lymphocytes. Moreover, detection can be also combined with their counting and quantitative analysis of cells [11]. Automatic cell counting involves obtaining the number of cells in a medical image [12].

In recent years, due to the rapid development of deep learning networks, they have become a key component of many computer vision applications such as object detection, classification or segmentation. The efficiency and efficacy of deep learning in the medical imaging field is unquestionable, as evidenced by a large number of independent studies in different modalities and applications, including those suggested for automatic cell counting [13]. For example, deep learning models that classify various types of erythrocytes were proposed in [14,15]. Vogado et al. [16] proposed LeukNet, which is based on a convolutional neural network (CNN). Acevedo et al. proposed recognition of peripheral blood cell images using CNNs [17]. Automatic white blood cell classification using deep learning models was also presented in [18–23]. Automatic identification and counting of all three types of blood cells simultaneously using DNN was proposed in [24].

A literature review indicates that there are only a few articles on the detection and counting of RBCs, WBCs, and platelets simultaneously using deep learning methods [24]. However, it is not clear how to determine the optimal number of epochs and the optimal threshold to achieve the highest performance. We also noted that the obtained results are usually compared only based on accuracy, which in no doubt is an important metric to consider, but it does not always give the full picture. Obtained results should also be discussed in the light of important quality metrics in medical testing: recall, precision, and F1-score. Many works concern recognizing cells in small images that contain just a few cells in the image, while microscopic images can include hundreds of crowded and overlapped cells. Motivated by the lack of a thorough examination of the above issues, we decided to propose our own solution.

This paper aims at developing a precise and automatic method for counting various types of cells in one image using the developed deep learning methods. It will allow for a significant acceleration of cell counting work in laboratories and a reduction of the burden on staff. Doing this work by using a computer will also reduce human error and increase the accuracy and reduce the likelihood of mistakes. To achieve this goal, our work was related to the development of methods that can automatically count blood cells. We proposed an approach that employs RetinaNet based on CNN architecture to detect all three types of blood cells, i.e., RBCs, WBCs, and platelets simultaneously.

The main contribution of this work includes several points. We prepared our own training dataset and manually marked RBCs, WBCs and platelets in the images. Then, we adapted and trained RetinaNet to recognize three types of cells simultaneously by presenting a wide collection of microscopic medical images. Next, we prepared an application that counted cells recognized by the RetinaNet network. Then, we evaluated the impact of learning epochs and confidence thresholds on the performance and effectiveness of cell detection and counting for each class on several images by comparing the number of cells counted by the application with the manually counted number of correctly classified cells, incorrectly classified cells, and unclassified cells. Based on those preliminary results, we selected and tested two of the trained models to evaluate how accurately they mark RBCs,

WBCs, and platelets with a bigger test set for subsequent confidence thresholds. Finally, we calculated the accuracy, precision, recall, and F1-score of automatic counting for each type of cells, determined the optimal confidence thresholds for each type of cells, and compared them with the state-of-the-art.

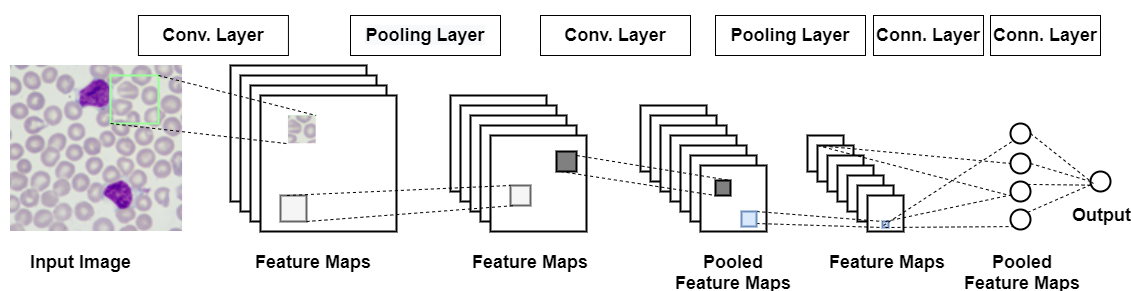
## 2. Materials and Methods

### 2.1. General Concept of CNN Construction

Deep learning is a method that simulates the human brain structure. This method consists of a series of algorithms for finding a hierarchical representation of the input data based on the way that the human brain senses an important part of a sensory data set. It is a part of machine learning, which revolves around the algorithms responsible for modeling high-level abstraction, using many layers composed of nonlinear transformations. Due to their high efficiency, DNNs are nowadays the most popular group of deep learning algorithms.

In recent years, the unrestrainable increase of the data amount has raised new challenges in machine learning in the area of scalability. It was particularly evident in the subject of object recognition and image processing. During the analysis of a small black and white image, each neuron of the hidden layer would still have to have thousands of weights. This fact causes problems of both computational and purely practical nature. Such problems are dealt with by the architecture of CNNs [25].

A CNN is a class of DNNs, most commonly applied to analyzing images and object recognition. Figure 1 shows the sequence of transformations involved in a typical convolutional network [26] that has been adopted in our research to recognize blood cells.



**Figure 1.** The sequence of transformations involved in the convolutional network for recognizing blood cells.

At first, the input image is scanned for feature selection. The checked rectangle is the filter that passes over the image. Activation maps are stacked atop another one for each of the employed filters. Secondly, the next rectangle is downsampled and the activation maps are downsampled. Next, a new set of activation maps is created by passing filters over the first downsampled stack. Then, the second set of activation maps is condensed by the second downsampling. Finally, the fully connected layer classifies the output with one label per node.

It is a solution taken from the human system of vision. Neurons are activated only when something is in the human field of vision, utilizing the fact that the features that represent only this small part of an image can relate to the entire surface of the image. Based on this knowledge, groups of neurons are created with common weights but located in different parts of the image. Several types of layers make CNN:

- Convolutional layers—they create feature maps based on systematically learned filters on input images and summarize the presence of these functions in the input. A map of the activity of a particular feature across the entire image area can be interpreted as a set of output signals from neurons of the same weight shall. The filter is a feature represented by one shared set of weights. The convolutional layer is operating in

three dimensions, where instead of multiplying vectors, as in the classical approach, the convolution operation is applied and it gives better results when detecting a pattern [25,26];

- Pooling layers—they are used to streamline the computation. Combining the outputs of neuron clusters at one layer into a single neuron in the next layer by pooling layers reduces the dimensions of the data. Local pooling combines small clusters, and global pooling acts on all neurons of the convolutional layer. Pooling may calculate a maximum or an average. Max pooling uses the maximum value, and average pooling uses the average value from each of a cluster of neurons at the prior layer [25,26].
- Fully connected layer—uses the convolution results to classify the image into a label. The convolution output is flattened into a single vector of values representing the probability of belonging of a feature to that label. Each neuron receives weights that assign priority to the most appropriate label. Finally, neurons vote for each label, and the winner of this vote is the classification decision [26].

## 2.2. RetinaNet

RetinaNet is a one-stage detector that uses focal loss, whereby the lower loss is contributed by negative samples. The loss is concentrated in problematic samples, which improves the accuracy of prediction. With ResNet and Feature Pyramid Network (FPN) as the backbone for extraction of features and two task-specific subnetworks used for classification and bounding box regression, the formed RetinaNet achieves excellent performance and outperforms Faster R-CNN—the well-known two stage detector [27,28].

The architecture of RetinaNet shown in Figure 2 can be divided into three main groups [29]:

- a backbone FPN is used on the top of the ResNet model for constructing a rich multiscale feature pyramid from a single input image;
- a subnet used for classifying objects based on FPN outputs;
- a subnet that makes regression of the bounding box using the output data of the backbone network.

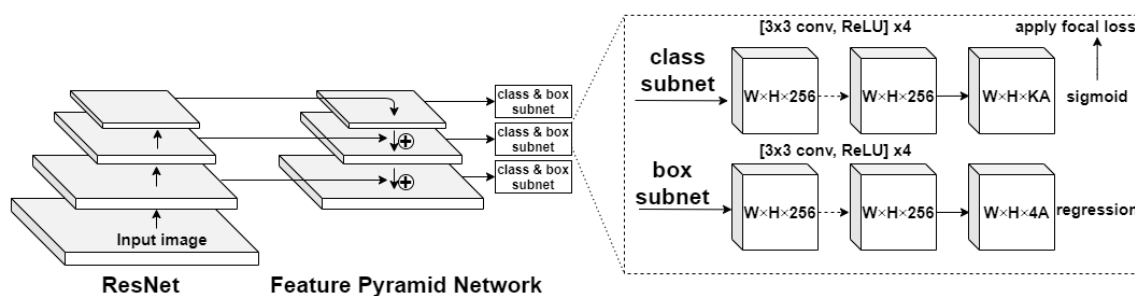


Figure 2. The architecture of RetinaNet Detector.

Feature pyramids are a basic component in recognition systems used for detecting objects at multiple scales. RetinaNet is based on the FPN presented in [30]. In a network containing residual blocks (ResNet), each layer feeds directly into the next layer and two to three jumped layers. In comparison, in traditional neural networks each layer feeds into the next layer. The training of a few layers can be skipped by using shortcut connections. It has been proven that training this type of network is easier than training in simple DNNs, and it particularly deals with the problem of accuracy degradation.

The fully convolutional nature of the network enables downloading an image of any scale and output proportional feature maps on multiple levels in the feature pyramid [31].

FPN consists of a bottom-up and top-down pathway. The bottom-up pathway is a convolutional network used for feature extraction, and the top-down pathway restores resolution to semantic information.



The classification and regression subnets are attached to each feature map obtained using FPN. The classification subnet predicts the object presence probability for each of the  $A$  anchors and  $K$  object classes at each spatial position. It applies four  $3 \times 3$  convolutional layers, each with 256 filters and each followed by the Rectified Linear Unit (ReLU) activation, followed by a  $3 \times 3$  convolutional layer with  $K \times A$  filters. The regression subnet is identical to the classification subnet, except that  $4A$  linear outputs are terminated per spatial location.

We used Keras implementation of RetinaNet object detection [32]. RetinaNet makes use of a ResNet-based backbone, from which a FPN is constructed. We used ResNet50 as the backbone. We took advantage of the possibility of using transfer learning. We set the weight option to the pretrained model when training and used the freeze backbone argument to freeze the backbone layers. We set the input batch size at 5 due to limitations in GPU memory. We trained the RetinaNet model with 36,382,957 parameters, which is equal to the number of trainable and non-trainable parameters.

### 2.3. Focal Loss

The imbalance between the background not containing objects and the foreground that holds interesting objects is the main issue for object detection model training. Focal loss is designed to assign greater weights to difficult, easily misclassified objects and downweight trivial ones. The goal is to minimize the expected value of the loss from the model and in the case of the cross-entropy loss, the expected loss is approximated as:

$$\text{CE}(p_i, y) = -\log(p_i) \approx \frac{1}{n} \sum_{i=1}^n -\log p_i = \frac{1}{n} \sum_{i=1}^n L_i \quad (1)$$

where  $L_i$  is the loss for one training example and the total loss  $L$  is approximated as the mean overall examples,  $p_i \in [0, 1]$  is the model's estimated probability for the class  $y = 1$ , and  $y \in \{\pm 1\}$  specifies the ground-truth class [30].

The loss is calculated depending on the loss function definition. One of the most common loss functions is cross-entropy loss. This loss function is beneficial for image classification tasks, but different tasks need different loss functions. For example, in the detection problem in which bounding boxes are estimated around objects, a regression loss function can be used to get a measure of how well the bounding box is placed in the image.

The cross-entropy loss is used when the model contains the Softmax classifier. The Softmax classifier gives a probability score for each object class. The loss function is calculated as:

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right) \quad (2)$$

where  $L_i$  are all the training examples together,  $f_j$  is the  $j$ -th element of the vector of class scores  $f$ ,  $y_i$  is the output for the correct class.

The Mean Square Error (MSE) is the most commonly used regression loss function. It can be computed as the squared norm of the difference between the true value and the predicted value:

$$L_i = \|g - y_i\|_2^2 \quad (3)$$

where  $g$  are the predicted values and  $y_i$  are the true ones. This loss function can be used when the goal is to find the coordinates of a bounding box when performing object detection.

### 2.4. Metrics

To quantitatively evaluate the results of cell counting, the following measures are defined.

The accuracy is defined by the following formula:

$$\text{Accuracy} = \frac{N_{\text{expert}} \cap N_{\text{count}}}{\max\{N_{\text{expert}}, N_{\text{count}}\}} \cdot 100\% \quad (4)$$

where:  $N_{\text{expert}}$ —number of cells counted by an expert,  $N_{\text{count}}$ —cells counted by the application.

The classifier efficiency is evaluated based on its ability to correctly identify the number of cells belonging to one of the three classes. For each class, the quantitative measurement is performed based on True Positive ( $TP$ ), False Positive ( $FP$ ), True Negative ( $TN$ ), and False Negative ( $FN$ ) parameters.

Precision is the fraction of correctly identified samples of a given class to all correctly recognized samples. This value is given by the formula:

$$\text{Precision} = \frac{TP}{TP + FP} \cdot 100\% \quad (5)$$

Recall (sensitivity) is the number of correctly identified samples belonging to a given class to all samples belonging to that class. It is expressed by the formula:

$$\text{Recall} = \frac{TP}{TP + FN} \cdot 100\% \quad (6)$$

F1-score is the harmonic average of recall and precision, which can be expressed by the formula:

$$\text{F1-score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \cdot 100\% \quad (7)$$

### 3. Implementation of Cell Counting Algorithm

Our goal is to use an object detection and classification algorithm to detect and count three types of blood cells directly from a smear image. For this purpose, we have needed to train the RetinaNet network with selected settings and configurations based on training images with blood cell annotation. In this way, we created an application for recognizing and counting blood cells.

#### 3.1. Datasets

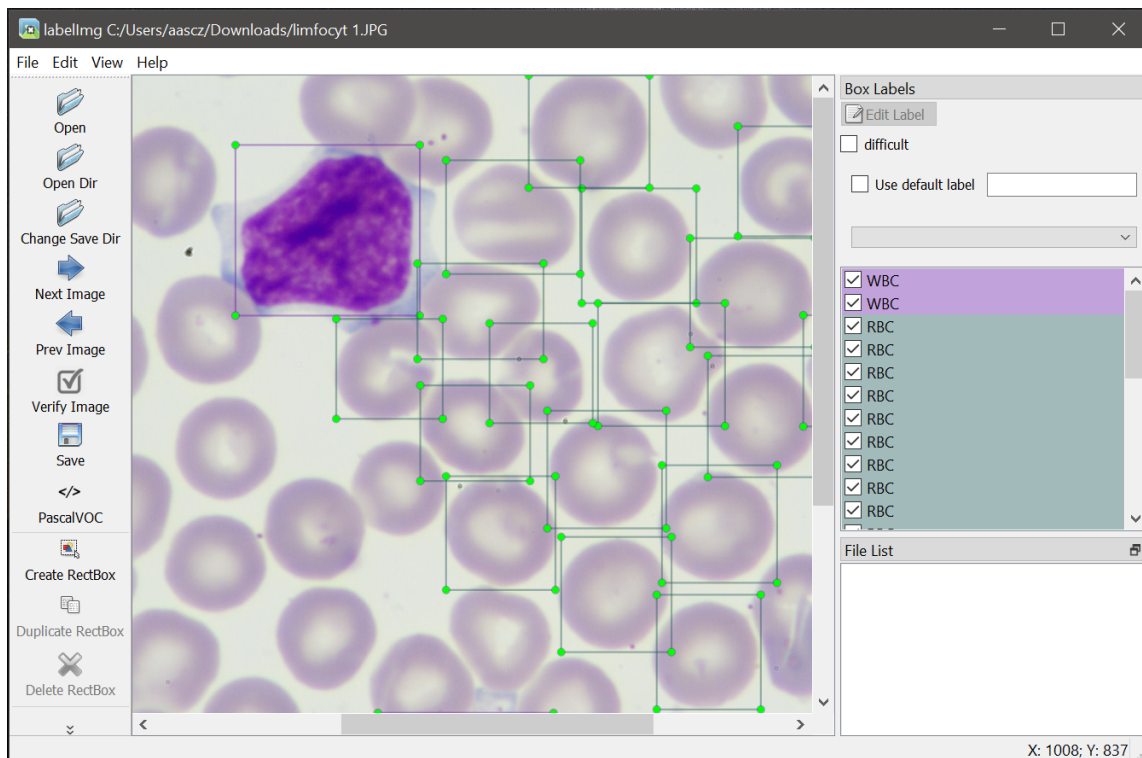
For the learning and validation application, we used our own dataset consisting of 900 images containing WBCs, RBCs, and platelets. In the case of the validation dataset, we randomly selected 15 training images with annotations.

For application tests, we used images from the LISC dataset [33]. The dataset includes 251 images of resolution  $720 \times 576$  acquired by a light microscope (Axioskope 40) with a magnification of  $100\times$ , recorded by a digital camera (Sony Model No. SSCDC50AP). From the test dataset, we randomly selected 131 images for counting WBCs, 64 images for counting platelets, and 15 images for counting RBCs. The different number of images selected for testing is due to the different number of individual cells in one image. Therefore, a small number of images for testing RBCs was selected, because of the large number of RBCs in individual images (average 121 RBCs per image). The situation is similar for the platelet count.

#### 3.2. Image Labelling

Before starting the network training process, we marked manually three types of cells in microscopic images using the Labelling application, which is a graphical image annotation tool [34]. This process is shown in Figure 3. The objects in the images are divided into three categories: WBCs, RBCs, and platelets are marked accordingly. In this way, the annotations of blood cells were acquired for DNN training.





**Figure 3.** The process of marking the training dataset.

### 3.3. Training the RetinaNet of Object Recognition

We used the RetinaNet network with ResNet50 as the backbone with the input batch size set at 5, and the number of epochs set at 40 epochs, each for 500 steps for training. We used 900 images to train the network. The training process outputs a JSON file containing the network trained on these images, based on the set parameters.

As a result of the training of each of the models, we obtained 40 files for each epoch. Then, we selected the results with 10, 15, 20, 25, 30, 35, and 40 epochs to investigate the impact of decreasing loss function on the detection accuracy. The workflow of the network learning process is presented in Figure 4.

Figure 5 shows the learning curve of the RetinaNet algorithm to detect blood cells relative to the regression and classification loss function, as well as according to the sum of losses.

### 3.4. Selection of the Optimal Model

The criteria for selecting the best model variant were based on observation of the loss function, which decreased during learning from epoch to epoch. Additionally, we manually validated the results obtained after 10, 15, 20, 25, 30, 35, and 40 learning epochs using a validation set consisting of 15 images not used for training. We assessed the efficiency of blood cell counting by calculating the mean F1-score for each of the considered thresholds for each epoch. The results of the preliminary analysis are presented in Tables 1–3.

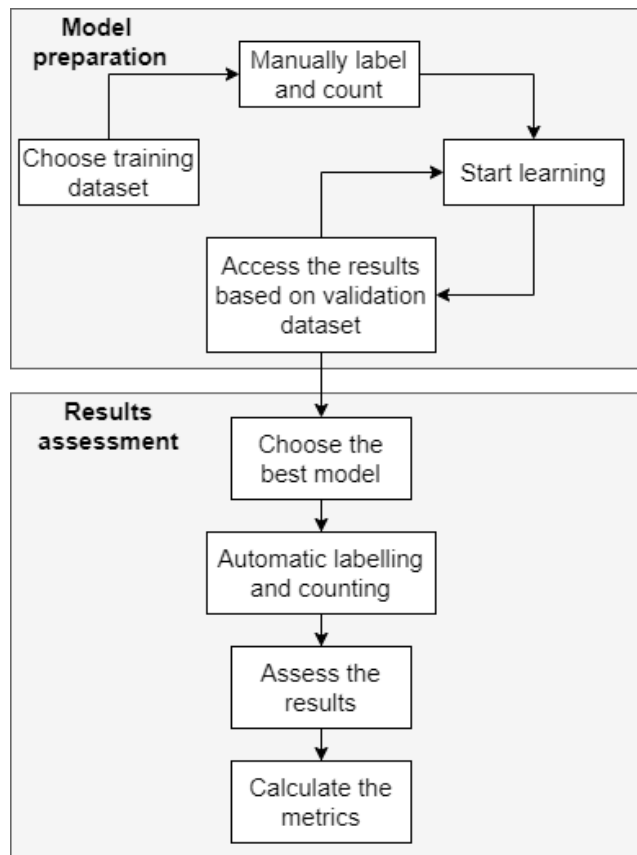


Figure 4. The cell counting workflow.

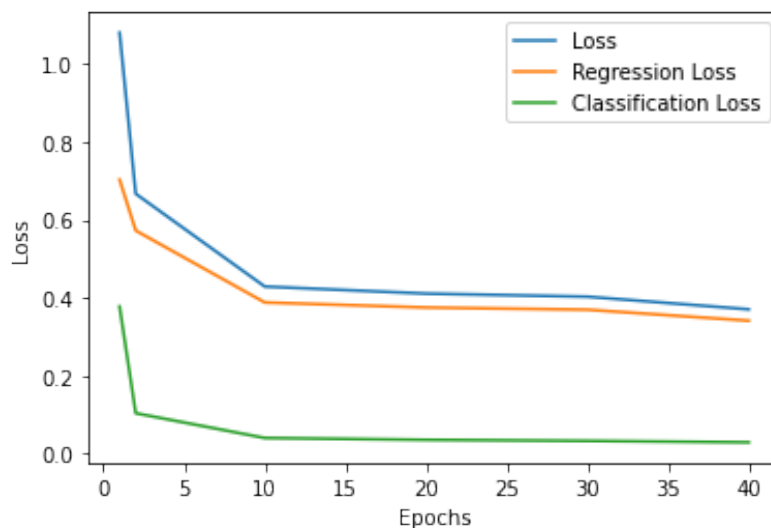


Figure 5. Learning curve of the RetinaNet blood cells identification (500 steps per epoch).

The additional aim of this validation was to compare the quality of cell counting after passing a certain number of epochs and to find the optimal model for further testing. The learning process was quite long. For a detailed analysis, we selected models trained with 10

and 30 epochs. The model trained with 10 epochs achieved very high F1-score results, and the loss function was stabilized for it. The model obtained after learning with 30 epochs achieved the highest F1-score values. We conducted research on a larger testing dataset for these two selected models and calculated metrics, such as F1-score, accuracy, precision and recall, allowing for an in-depth and comprehensive assessment of the quality of the RetinaNet model.

**Table 1.** Preliminary F1-scores for the recognition of RBCs, obtained from the analysis of 15 images, used to select the optimal model.

Threshold	RBCs F1-Score [%]						
	RN10	RN15	RN20	RN25	RN30	RN35	RN40
0.20	87.11	87.36	86.81	87.47	85.65	87.97	87.99
0.25	87.51	87.59	87.97	88.18	87.05	87.68	88.40
0.30	87.57	87.85	87.97	88.39	88.51	87.56	87.99
0.35	87.94	86.99	88.12	88.47	87.22	87.09	86.29
0.40	86.47	84.61	86.85	87.32	86.35	85.91	84.22
0.45	84.04	82.74	84.96	85.37	84.82	83.92	82.86
0.50	81.14	80.64	83.25	83.39	82.49	82.08	81.26
0.55	78.00	78.26	80.69	80.95	80.36	80.00	78.51
0.60	73.95	73.83	78.64	78.92	78.13	77.80	75.97
0.65	69.39	70.16	75.28	76.09	76.10	75.55	73.91
0.70	64.20	66.31	72.14	73.52	72.71	72.45	71.11
0.75	56.67	61.09	68.52	69.73	69.46	68.82	67.76
0.80	48.42	52.19	62.34	65.03	64.92	65.03	64.57
0.85	36.80	43.32	55.41	58.65	60.44	60.83	59.70

**Table 2.** Preliminary F1-score values for the recognition of WBCs, obtained from the analysis of 15 images, used to select the optimal model.

Threshold	WBCs F1-Score [%]						
	RN10	RN15	RN20	RN25	RN30	RN35	RN40
0.20	21.18	28.57	26.28	24.32	19.46	19.88	14.46
0.25	36.73	43.90	39.13	33.96	26.67	27.20	18.28
0.30	51.43	55.74	48.57	42.50	34.29	32.69	23.45
0.35	61.54	64.00	65.38	57.63	44.44	44.16	29.06
0.40	73.17	75.00	65.31	62.50	53.12	58.62	35.79
0.45	85.71	83.33	75.00	69.77	61.54	64.00	43.59
0.50	88.24	88.24	78.95	71.43	63.83	68.18	47.06
0.55	90.91	88.24	85.71	75.00	68.18	71.43	51.72
0.60	90.91	90.91	90.91	83.33	73.17	78.95	57.69
0.65	87.50	90.91	90.91	90.91	83.33	85.71	73.17
0.70	83.87	87.50	90.91	90.91	88.24	88.24	76.92
0.75	80.00	87.50	87.50	87.50	90.91	90.91	85.71
0.80	61.54	75.86	87.50	87.50	87.50	87.50	84.85
0.85	56.00	66.67	80.00	80.00	80.00	83.87	87.50

Analyzing the results of the F1-score presented in Tables 1–3, obtained on the basis of counting 3 types of cells in 15 images for each model and each threshold, it turned out that the best results of RBCs, WBCs, and platelet counting was achieved by RetinaNet trained during 30 epochs. That model returns the highest values of recognized RBCs, platelets, as well as WBCs. The same maximum values of F1-score values for WBCs also occur in other models, except the RN40. However, taking into account the maximum values of F1-score counting of all three types of cells from Tables 1–3, it can be indicated that the best is the RN30 model.

**Table 3.** Preliminary F1-score values for the recognition of platelets, obtained from the analysis of 15 images, used to select the optimal model.

Threshold	Platelets F1-Score [%]						
	RN10	RN15	RN20	RN25	RN30	RN35	RN40
0.20	73.95	71.07	75.62	75.23	73.17	73.49	66.95
0.25	80.65	81.72	82.58	81.82	81.23	80.75	73.35
0.30	83.99	85.46	83.03	83.57	86.53	80.94	76.88
0.35	82.05	85.36	83.91	85.45	85.11	82.39	80.60
0.40	80.13	81.70	82.39	83.60	83.97	81.97	80.51
0.45	77.82	80.00	80.41	80.95	81.19	80.41	79.21
0.50	73.84	78.08	77.03	78.32	78.50	77.35	77.51
0.55	68.42	73.76	75.27	74.82	75.00	73.45	74.47
0.60	58.30	68.66	64.59	68.18	70.85	67.18	68.66
0.65	47.83	58.06	57.61	60.80	64.59	60.24	63.53
0.70	41.28	52.14	52.77	52.77	55.00	53.16	53.78
0.75	35.24	42.73	38.32	44.84	49.57	47.58	49.35
0.80	27.86	34.45	30.39	35.24	41.28	39.07	39.81
0.85	13.98	24.37	24.37	24.37	27.86	26.13	26.13

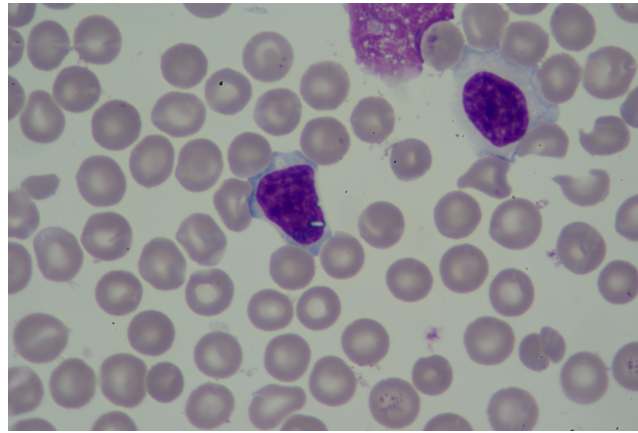
#### 4. Experiments and Results

After the network training, we performed tests using a specially developed application, which allows for the import of trained models, cell detection, and presentation of results in a graphical and numerical form.

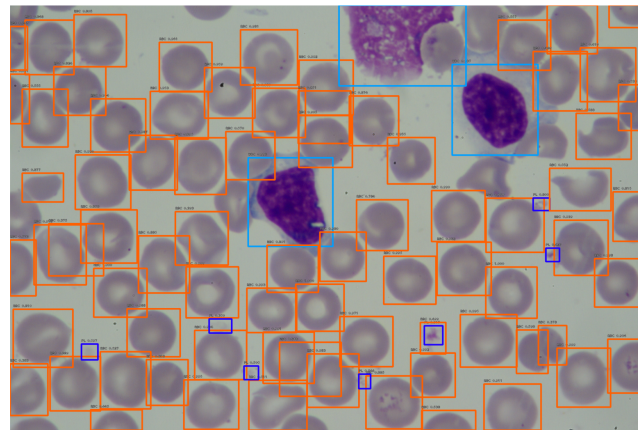
The tests of the developed models were performed for 15 images with RBCs, 151 images with WBCs, and 64 images with platelets. The output of the deep learning model is an image with an appropriate marking of the recognized samples. To verify the correctness of the obtained results, we counted all marked cells applied to their type in the dedicated application. Thus, our application counts different cells in the selected testing dataset with a different confidence threshold for selected models (RetinaNet model trained with 10 epochs (RN10) and the RetinaNet trained with 30 epochs (RN30)). We compared the results obtained for both types of models in order to check the impact of loss function values on the performance of object recognition.

It should be noted that the confidence threshold plays an instrumental role. Accuracy of identification and counting significantly depends on the appropriate confidence threshold setting. The values of different measures to estimate the accuracy of the recognition and counting of blood cells for testing data were presented in Tables 4–10.

To visualize the operation of the proposed labeling and blood cell counting method, we presented one of the images from the validation set. Figure 6 shows an original blood smear image, and Figure 7 shows the same image with automatically drawn bounding boxes, labels, and probabilities of each marked blood cell. It was returned by our application using the RN30 model with the confidence threshold of 0.35. Recognized cells were automatically marked on the bounding boxes according to their type. Orange bounding boxes mark RBCs, light blue mark WBCs, and dark blue mark platelets. It is seen in Figure 7 that WBCs and platelets are detected without error. Almost all RBCs are correctly labeled. However, three erroneous orange frames are also noticed, which includes a part of two neighboring RBC cells.



**Figure 6.** An example of blood smear image from the validation dataset.



**Figure 7.** An example of blood smear image with recognized RBCs, WBC and platelets by the RN30 model.

#### 4.1. Results for the RN10

Table 4 contains the determined values of accepted quality measures for the RetinaNet model after 10 learning epochs (RN10) for 15 test images. Table 5 includes an assessment of WBCs counting quality in 131 images, and Table 6 contains the results of counting platelets in 64 images. Table 7 contains the ground truth of cells and cell numbers counted by our application for the confidence thresholds considered.

**Table 4.** The accuracy, precision, recall, and F1-score of automatic counting of RBCs using RetinaNet model for 10 epochs (15 images).

Threshold	Accuracy [%]	Precision [%]	Recall [%]	F1-Score [%]
0.20	86.68	81.30	93.80	87.10
<b>0.25</b>	<b>96.91</b>	<b>87.23</b>	<b>90.01</b>	<b>88.60</b>
0.30	93.47	90.66	84.74	87.60
0.35	85.24	93.69	79.86	86.22
0.40	78.05	95.64	74.64	83.85
0.45	71.68	97.17	69.65	81.14
0.50	64.76	98.39	63.72	77.35
0.55	59.22	99.35	58.84	73.91
0.60	53.51	99.79	53.40	69.57
0.65	47.64	99.88	45.94	62.93
0.70	41.60	100.0	41.60	58.76
0.75	34.80	100.0	34.80	51.63
0.80	28.38	100.0	28.38	44.21
0.85	19.92	100.0	19.92	33.23

**Table 5.** The accuracy, precision, recall, and F1-score of automatic counting of WBCs using RetinaNet model for 10 epochs (131 images).

Threshold	Accuracy [%]	Precision [%]	Recall [%]	F1-Score [%]
0.20	18.51	18.25	98.61	30.80
0.25	31.17	30.74	98.61	46.86
0.30	45.71	45.08	98.61	61.87
0.35	66.67	64.81	97.22	77.78
0.40	79.56	77.35	97.22	86.15
0.45	92.90	89.68	97.20	93.29
0.50	97.30	93.24	96.50	94.85
0.55	97.22	96.43	93.75	95.07
<b>0.60</b>	<b>94.44</b>	<b>98.53</b>	<b>93.06</b>	<b>95.71</b>
0.65	90.28	98.46	90.14	94.12
0.70	81.25	99.15	81.69	89.58
0.75	70.83	99.02	72.14	83.47
0.80	52.78	97.37	52.86	68.52
0.85	36.11	100.0	37.14	54.17

**Table 6.** The accuracy, precision, recall, and F1-score of automatic counting of platelets using RetinaNet model for 10 epochs (64 images).

Threshold	Accuracy [%]	Precision [%]	Recall [%]	F1-Score [%]
0.20	69.18	68.56	98.97	81.01
0.25	86.36	82.93	95.90	88.94
<b>0.30</b>	<b>98.72</b>	<b>91.68</b>	<b>90.85</b>	<b>91.26</b>
0.35	88.45	96.08	86.09	90.81
0.40	81.39	97.48	79.84	87.78
0.45	72.79	99.47	73.06	84.24
0.50	63.80	99.80	64.50	78.36
0.55	53.79	100.0	54.49	70.54
0.60	45.96	100.0	46.86	63.81
0.65	38.25	100.0	39.11	56.23
0.70	30.94	100.0	31.54	47.96
0.75	22.21	100.0	22.21	36.34
0.80	14.25	100.0	14.25	24.94
0.85	8.09	100.0	8.09	14.96

**Table 7.** Ground truth and the estimated number of blood cells at different confidence thresholds for the RN10.

Threshold	RBC		WBC		Platelets	
	Ground truth	Estimated	Ground Truth	Estimated	Ground Truth	Estimated
0.20	1822	2102	144	778	779	1126
0.25	<b>1822</b>	<b>1880</b>	144	462	779	902
0.30	1822	1703	144	315	<b>779</b>	<b>769</b>
0.35	1822	1553	144	216	779	689
0.40	1822	1422	144	181	779	634
0.45	1822	1306	144	155	779	567
0.50	1822	1180	144	148	779	497
0.55	1822	1079	144	140	779	419
0.60	1822	975	<b>144</b>	<b>136</b>	779	358
0.65	1822	868	144	130	779	298
0.70	1822	758	144	117	779	241
0.75	1822	634	144	102	779	173
0.80	1822	517	144	76	779	111
0.85	1822	363	144	52	779	63

#### 4.2. Results for the RN30

Table 8 contains the calculated values of the adopted quality measures for the RetinaNet model after 30 learning epochs (RN30) for 15 test images. Table 9 includes an assessment of the WBCs counting quality in 131 images, and Table 10 contains the results of counting platelets in 64 images. Total estimated numbers of cells of different types for different confidence threshold values are presented in Table 11.

**Table 8.** The accuracy, precision, recall, and F1-score of automatic counting of RBCs using RetinaNet model for 30 epochs (15 images).

Threshold	Accuracy [%]	Precision [%]	Recall [%]	F1-Score [%]
0.20	91.24	81.18	88.41	84.64
<b>0.25</b>	<b>99.67</b>	<b>86.44</b>	<b>86.54</b>	<b>86.49</b>
0.30	91.99	90.51	83.35	86.78
0.35	86.39	93.20	80.60	86.45
0.40	80.90	94.91	76.87	84.94
0.45	75.30	96.43	72.69	82.89
0.50	70.25	97.11	68.30	80.19
0.55	65.81	97.75	64.40	77.64
0.60	62.18	98.23	61.15	75.38
0.65	58.29	98.59	57.53	72.66
0.70	53.35	99.18	52.97	69.05
0.75	48.35	99.66	48.08	64.86
0.80	43.14	99.87	43.02	60.14
0.85	37.87	100.0	37.91	54.98

**Table 9.** The accuracy, precision, recall, and F1-score of automatic counting of WBCs using RetinaNet model for 30 epochs (131 images).

Threshold	Accuracy [%]	Precision [%]	Recall [%]	F1-Score [%]
0.20	13.51	13.52	100.0	23.82
0.25	21.02	21.02	100.0	34.74
0.30	30.38	30.38	100.0	46.60
0.35	41.38	41.38	100.0	58.54
0.40	53.33	53.33	100.0	69.57
0.45	65.75	64.84	98.61	78.24
0.50	75.39	73.82	97.92	84.18
0.55	86.75	84.34	97.22	90.32
0.60	96.00	92.67	96.53	94.56
<b>0.65</b>	<b>98.61</b>	<b>97.89</b>	<b>96.53</b>	<b>97.20</b>
0.70	97.22	99.29	96.53	97.89
0.75	95.14	100.0	95.14	97.51
0.80	90.97	100.0	90.97	95.27
0.85	79.86	100.0	79.86	88.80

**Table 10.** The accuracy, precision, recall, and F1-score of automatic counting of platelets using RetinaNet model for 30 epochs (64 images).

Threshold	Accuracy [%]	Precision [%]	Recall [%]	F1-Score [%]
0.20	67.33	66.55	98.97	79.59
0.25	82.43	80.00	97.05	87.70
0.30	93.29	87.90	94.22	90.95
<b>0.35</b>	<b>97.82</b>	<b>92.78</b>	<b>90.76</b>	<b>91.76</b>
0.40	89.73	95.71	85.88	90.53
0.45	83.95	97.40	81.77	88.90
0.50	77.79	98.84	76.89	86.50
0.55	69.96	99.27	69.45	81.72
0.60	62.77	99.59	62.52	76.81
0.65	55.84	100.0	55.84	71.66
0.70	47.37	100.0	47.37	64.29
0.75	40.56	100.0	40.56	57.72
0.80	30.94	100.0	30.94	47.25
0.85	21.44	100.0	21.44	35.31

**Table 11.** Ground truth and the estimated number of blood cells at different confidence thresholds for the RN30.

Threshold	RBC		WBC		Platelets	
	Ground Truth	Estimated	Ground Truth	Estimated	Ground Truth	Estimated
0.20	1822	1997	144	1066	779	1157
0.25	<b>1822</b>	<b>1828</b>	144	685	779	945
0.30	1822	1676	144	474	779	835
0.35	1822	1574	144	348	<b>779</b>	<b>762</b>
0.40	1822	1474	144	270	779	699
0.45	1822	1372	144	219	779	654
0.50	1822	1280	144	191	779	606
0.55	1822	1199	144	166	779	545
0.60	1822	1133	144	150	779	489
0.65	1822	1062	<b>144</b>	<b>142</b>	779	435
0.70	1822	972	144	140	779	369
0.75	1822	881	144	137	779	316
0.80	1822	786	144	131	779	241
0.85	1822	690	144	115	779	167

As it is apparent from Table 8, in order to count RBCs, it is best to use the optimal threshold of 0.25. However, to count WBCs and platelets, the threshold is much higher



(0.65 and 0.35 sequentially for Tables 9 and 10). Thus, appropriate thresholds for each type of cells are selected as follows:

- RBCs—Confidence Threshold: 0.25,
- WBCs—Confidence Threshold: 0.65,
- Platelets—Confidence Threshold: 0.35.

For the RN10 model, the optimal confidence thresholds determined based on Tables 4–6 are as follows: 0.25 for RBCs, 0.60 for WBCs, and 0.30 for platelets. Thus, the thresholds in the counting of WBCs and platelets in the RN30 model are higher and increased as a result of learning the RN10 for another 20 epochs. Higher confidence thresholds give greater certainty of correct recognition of individual blood cells.

The growth of the confidence threshold, which occurs in the RetinaNet model due to the network learning process, can be seen by analyzing and comparing Tables 4–11, as well as analyzing Figures 8–10, which shows the impact of the value of the confidence threshold on the number of counted cells concerning ground truths. From this figure, you can also effortlessly determine the optimal confidence thresholds for the three blood cell classes considered.

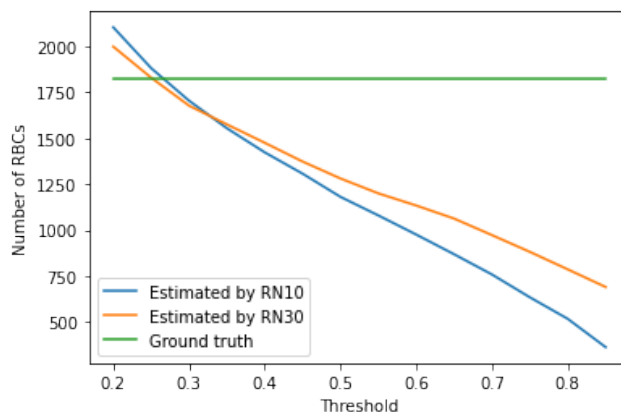


Figure 8. Number of detected RBCs vs. threshold value.

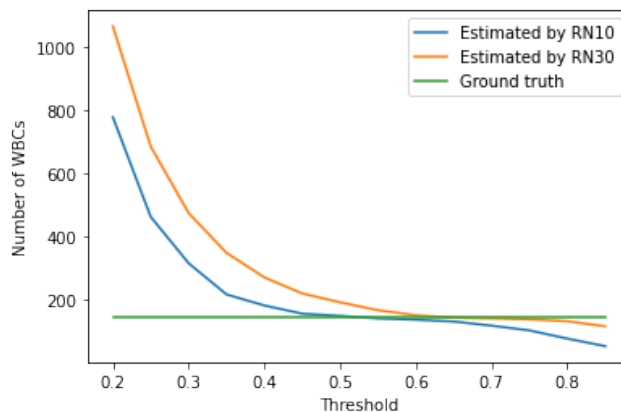
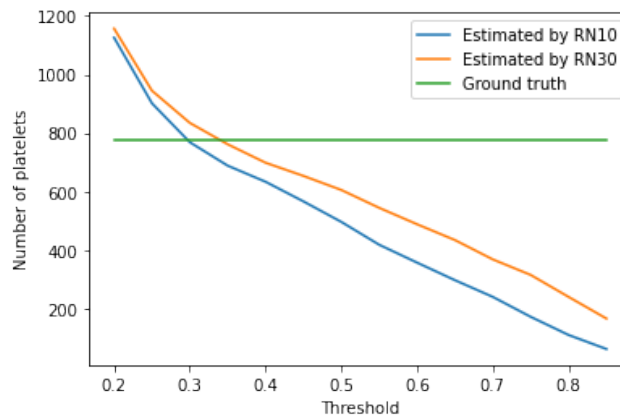
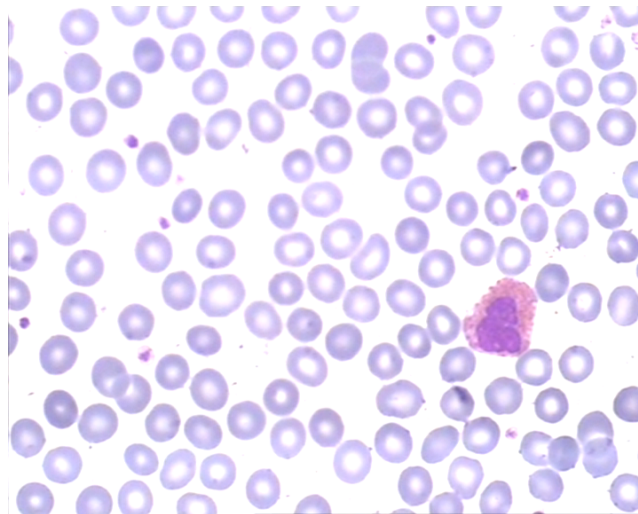


Figure 9. An example of blood smear image with recognized RBCs, WBC and platelets by the RN30 model.

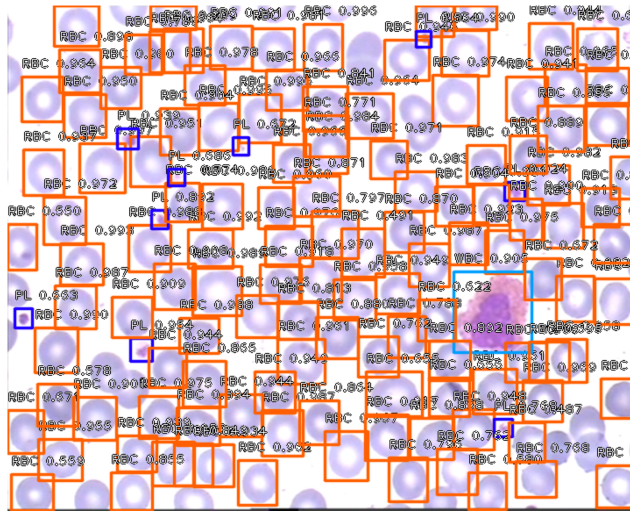


**Figure 10.** An example of blood smear image with recognized RBCs, WBC and platelets by the RN30 model.

Figure 11 and shows the original image from the testing dataset. It was processed by our application, which automatically drew the bounding boxes, labels, and probabilities of each marked blood cell. An image processed using the RN30 model with the established confidence threshold of 0.45 is in Figure 12. Labels have determined colors, relevant names, and a probability value for each blood cell. At an established confidence threshold of 0.45, the WBC was correctly recognized and all platelets have been correctly recognized, labeled, and counted. The vast majority of RBCs are recognized and labeled correctly. At this confidence threshold, the RN30 model counts RBCs with an accuracy of about 75%. It is seen in Figure 12 that there are only a few unchecked RBCs (typical for this threshold value), especially the RBCs that are overlapped or trimmed near the edge of the image. The application correctly recognized and counted one WBC. It also correctly recognized and counted 9 platelets and 123 RBCs when the ground truth is 135. The application also returns the probability values of each marked cell and the average probability of all recognized cells depending on their type. In this case, the average probability for WBCs was 0.905, for RBCs it was 0.875, and for platelets it was 0.784.



**Figure 11.** An example of blood smear image from the testing dataset.



**Figure 12.** An example of blood smear image with recognized RBCs, WBC, and platelets by the RN30 model.

#### 4.3. Comparison with the State-of-the-Art

To evaluate the performance of the proposed approach, we used the accuracy, precision, recall, and F1-score metrics, which are used most often for counting purposes. We compared the performance of the proposed approach with those obtained by other authors who dealt with the subject of cell counting for RBC, WBC, or platelet counting. It must be noted that only a few methods aimed to count both RBCs, WBCs, and platelets at the same time [24]. The selected methods work on the basis of deep learning as well as traditional image processing. Alam et al. [24] proposed an approach that employs YOLO to detect all three types of blood cells simultaneously. Their method does not require any greyscale conversion or binary segmentation and the whole process is fully automated. It is very similar to our approach because it uses deep neural networks to detect and count three types of cells. Dvanesh et al. [35] presented a method to digitally analyze the image of blood cells and find the RBC and WBC count values from the blood smear microscopic images using Digital Image Processing. Acevedo et al. [17] proposed a system for the automatic classification of peripheral blood cells (WBCs and platelets) by means of a transfer learning approach using convolutional neural networks. Di Ruberto et al. [36] proposed a system for detecting and quantifying red and white blood cells, which is based on the Edge Boxes method. That method is an approach for generating object bounding box proposals directly from edges.

A comparison of the RBC, WBC and platelet counting results with the results obtained by the other authors are reported in Tables 12–14. As can be observed, the proposed approach improves the counting performances; in particular, it significantly enhances accuracy. To highlight the performances obtained with the proposed method, in Table 12, we also report the number of images or ground truths used by the authors to test their approaches. The method proposed by [36] performed a higher precision, recall, and F1-score than our method.

**Table 12.** RBCs counting performance compared with the state-of-the-art.

	Alam [24]	Dvanesh [35]	Acevedo [17]	Ruberto [36]	Our Approach
Model	Tiny YOLO	ABCCS	-	Region proposal approach	RetinaNet50
No. images	60	63	-	108	15
Ground truths	792	-	-	-	1822
Accuracy	96.09	91.0	-	95.6	99.67
Precision	-	-	-	98.4	86.44
Recall	-	-	-	95.0	86.54
F1-score	-	-	-	96.6	86.49

The WBC counting results are reported in Table 13, which again have been directly compared with the results obtained by the other authors. The numerical results shown in Table 13 confirm the good performance of our approach, as it is able to detect WBCs with higher accuracy and precision than other methods. Only one method [36] performed higher recall and F1-score while losing accuracy and precision.

**Table 13.** WBC counting performance compared with the state-of-the-art.

	Alam [24]	Dvanesh [35]	Acevedo [17]	Ruberto [36]	Our Approach
Model	Tiny YOLO	ABCCS	Vgg-16	Region proposal approach	RetinaNet50
No. images	60	63	1919	108	131
Ground truths	61	-	-	-	144
Accuracy	86.89	85.0	96.20	97.0	98.61
Precision	-	-	-	97.6	97.89
Recall	-	-	-	98.7	96.53
F1-score	-	-	-	98.0	97.20

The platelet counting results are reported in Table 14, which have been compared with the results obtained by the same authors. The proposed approach obtained better accuracy than presented by Alam et al. [24] and is slightly worse than presented by Acevedo et al. [17].

**Table 14.** Platelet counting performance compared with the state-of-the-art.

	Alam [24]	Dvanesh [35]	Acevedo [17]	Ruberto [36]	Our Approach
Model	Tiny YOLO	-	Vgg-16	-	RetinaNet50
No. images	60	-	1919	-	64
Ground truths	55	-	-	-	144
Accuracy	96.36	-	99.61	-	97.82
Precision	-	-	-	-	92.78
Recall	-	-	-	-	90.76
F1-score	-	-	-	-	91.76

The obtained results are very satisfactory if we take into account that we are dealing with the recognition and counting of three types of cells simultaneously. However, it should be noted that similar to our approach was present only in one of the selected works [24] and in comparison to it, we obtained better performance in recognizing all three types of cells. Other works concerned the simultaneous recognition of two types of cells—WBCs and RBCs [35,36] or WBCs and platelets [17]. However, it should be noted that the compared results were obtained when tested on different datasets. For an accurate comparison of the results obtained, the approaches should be tested under the same conditions using the same datasets. Furthermore, the images used to test our approach contained average resolution images with a large number of cells (typically 100–150 cells per image) and the cells often overlapped each other, which impeded their correct recognition and counting.

## 5. Discussion

The results listed in Tables 4–11 indicate that each cell type has its optimal confidence threshold. For optimal thresholds, the highest accuracy of recognizing and counting individual cells was obtained. Using one common confidence threshold generally cannot provide accurate results, because when choosing an indirect common threshold, for example, 0.55, all counting indicators will be worse and the application will not count exactly individual cells, as in the case of optimal confidence thresholds. Thus, each type of cells should be counted separately with the individually selected confidence threshold to obtain the most accurate results.

For the Retina model, after 30 epochs (RN30), at the confidence threshold of 0.25, the accuracy of RBCs counting by application is 99.7%, precision is 86.4%, and recall is 86.5%. Accuracy of WBCs counting by application at a confidence threshold of 0.65 is 98.6%, the counting precision is 97.9%, and recall is 96.5%. In the case of counting platelets, for the optimal threshold of 0.35, the accuracy is 97.8, precision is 92.8%, and recall is 90.8%. Almost all quality indicators for the RN10 model are slightly lower than for the RN30. Only in an assured range of confidence thresholds, the accuracy and precision of counting cells in the RN10 model is better than in the RN30. However, comparing F1-score values, the maximum value of this metric was obtained for the RN30 model.

In the light of the results presented above, general conclusions can be made. The model RN10 after the relatively short learning process (10 epochs) may quite accurately count the blood cells. Furthermore, learning up to 30 epochs improves almost all counting performance metrics, and it also grows the confidence threshold for the best results. The model trained by 40 epochs shows signs of overtraining visible on preliminary results for validation data. The presented results, however, partially present the complexity of the problem of counting blood cells. Regarding the selection of the optimal model for blood cell counting applications, we came to the conclusion that it is a difficult, complex, and time-consuming process because the accuracy of counting depends on the confidence threshold, the time of learning, the number of epochs, selection of performance evaluation metrics and perhaps many other factors which we did not include in this work. With such a wide study, the optimal confidence thresholds have been established, for which the application counted cells very accurately with high precision. We can dispose of redundant and incorrect estimates of the number of cells by selecting an appropriate confidence threshold for each cell type instead of a general threshold for all blood cells. The results obtained are very satisfactory for the recognition and counting of three cell types simultaneously compared to other works on cell counting, and we achieved better quality measure values for assessing the effectiveness of our approach for most of them.

Finally, we have to mention that a very big advantage of the application, in addition to the precise counting, is the appropriate marking of all recognized cells with labels and probabilities. It allows for easy verification of the obtained results. Marking recognized cells so far is still a rare functionality used in counting methods. Our method works in images of high resolution and dimensions. Different methods must divide a large image into a smaller one with a few number of cells in the individual image, which gives our method an additional advantage.

## 6. Conclusions

This article presents a machine learning approach to automatic identification and counting of blood cells from a smear image based on CNN RetinaNet. The proposed method is evaluated on the basis of publicly available datasets. The developed methods have been tested on different types of cells with different cell density in the images and they show promising results. The developed application returns the results in numerical and graphical form, which enables their simple verification. Additionally, the graphical results, i.e., labeled cells, ensure the probability of correct recognition of the right cell. We observed that in the case of the testing dataset, our method accurately recognizes and

counts RBCs, WBCs, and platelets. However, the counting accuracy depends on the proper selection of the confidence threshold for individual cell classes.

An essential advantage is that the medical images do not require preliminary preparation, and all results are obtained after a single presentation of an image. All calculated metrics allow for in-depth and comprehensive evaluation of the quality of RetinaNet models. Due to the accuracy and performance of the detection, the proposed method has the potential to replace the manual identification of blood cells and the counting process. The developed application would allow for speeding up cell counting and increasing its accuracy.

**Author Contributions:** Conceptualization, G.D. and D.M.; methodology, G.D. and D.M.; software, G.D. and A.C.; validation, D.M. and G.D.; formal analysis, G.D.; investigation, G.D. and A.C.; resources, G.D.; data curation, G.D.; writing—original draft preparation, D.M., G.D. and A.C.; writing—review and editing, D.M., G.D. and A.C.; visualization, A.C.; supervision, G.D.; project administration, D.M.; funding acquisition, D.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This project is financed by the statutory funds (UPB) of the Department of Electrical and Computer Engineering Fundamentals, Rzeszow University of Technology.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

### Abbreviations

The following abbreviations are used in this manuscript:

CBC	Complete blood count
CNN	Convolutional neural network
DNN	Deep Neural Network
FN	False Negative
FP	False Positive
FPN	Feature Pyramid Network
MSE	Mean Square Error
RBC	Red Blood Cell
ReLU	Rectified Linear Unit
TN	True Negative
TP	True Positive
WBC	White Blood Cell

### References

- George-Gay, B.; Parker, K. Understanding the complete blood count with differential. *J. Perianesthesia Nurs.* **2003**, *18*, 96–114. [[CrossRef](#)]
- Lockley, S.W.; Barger, L.K.; Ayas, N.T.; Rothschild, J.M.; Czeisler, C.A.; Landrigan, C.P. Effects of Health Care Provider Work Hours and Sleep Deprivation on Safety and Performance. *Jt. Comm. J. Qual. Patient Saf.* **2007**, *33*, 7–18. [[CrossRef](#)]
- Maitra, M.; Gupta, R.K.; Mukherjee, M. Detection and counting of red blood cells in blood cell images using hough transform. *Int. J. Comput. Appl.* **2012**, *53*, 13–17. [[CrossRef](#)]
- Tomari, R.; Zakaria, W.N.W.; Ngadengon, R. An Empirical Framework For Automatic Red Blood Cell Morphology Identification and Counting. *ARPJ J. Eng. Appl. Sci.* **2015**, *10*, 8894–8901.
- Nemane, J.B.; Chakkarwar, V.A.; Lahoti, P.B. White blood cell segmentation and counting using global threshold. *Int. J. Emerg. Technol. Adv. Eng.* **2013**, *3*, 639–643.
- Putzu, L.; Di Ruberto, C. White Blood Cells Identification and Counting from Microscopic Blood Image. *Eng. Technol. Int. J. Med Health Sci.* **2013**, *7*, 20–27.
- Arslan, S.; Ozyurek, E.; Gunduz-Demir, C. A color and shape based algorithm for segmentation of white blood cells in peripheral blood and bone marrow images. *Cytom. Part A* **2014**, *85*, 480–490. [[CrossRef](#)] [[PubMed](#)]



8. Nazlibilek, S.; Karacor, D.; Ercan, T.; Sazli, M.H.; Kalender, O.; Ege, Y. Automatic segmentation, counting, size determination and classification of white blood cells. *Measurement* **2014**, *55*, 58–65. [[CrossRef](#)]
9. Safuan, S.; Tomari R.; Zakaria, W.N.W. White blood cell (WBC) counting analysis in blood smear images using various color segmentation methods. *Measurement* **2018**, *116*, 543–555. [[CrossRef](#)]
10. Zhang, C.; Xiao, X.; Li, X.; Chen, Y.-J.; Zhen, W.; Chang, J.; Zheng, C.; Liu, Z. White Blood Cell Segmentation by Color-Space-Based KMean Clustering. *Sensors* **2014**, *14*, 16128–16147. [[CrossRef](#)] [[PubMed](#)]
11. Xing, F.; Yang, L. Robust nucleus/cell detection and segmentation in digital pathology and microscopy images: A Comprehensive Review. *IEEE Rev. Biomed. Eng.* **2016**, *9*, 234–263. [[CrossRef](#)] [[PubMed](#)]
12. Xie, Y.; Xing, F.; Kong, X.; Su, H.; Yang, L. Beyond Classification: Structured Regression for Robust Cell Detection Using Convolutional Neural Network. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*; Navab N., Hornegger J., Wells W., Frangi, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2015; Volume 9351, pp. 358–365. 43. [[CrossRef](#)]
13. Kim, M.; Yan, C.; Yang, D.; Wang, Q.; Ma, J.; Wu, G. Chapter Eight—Deep learning in biomedical image analysis. In *Biomedical Engineering, Biomedical Information Technology*, 2nd ed.; Feng, D.D., Ed.; Academic Press: Cambridge, MA, USA, 2020; pp. 239–263. [[CrossRef](#)]
14. Alzubaidi, L.; Fadhel, M.A.; Al-Shamma, O.; Zhang, J.; Duan, Y. Deep Learning Models for Classification of Red Blood Cells in Microscopy Images to Aid in Sickle Cell Anemia Diagnosis. *Electronics* **2020**, *9*, 427. [[CrossRef](#)]
15. Parab, M.A.; Mehendale, N.D. Red Blood Cell Classification Using Image Processing and CNN. *SN Comput. Sci.* **2021**, *2*, 1–10. [[CrossRef](#)]
16. Vogado, L.; Veras, R.; Aires, K.; Araújo, F.; Silva, R.; Ponti, M.; Tavares, J.M.R.S. Diagnosis of Leukaemia in Blood Slides Based on a Fine-Tuned and Highly Generalisable Deep Learning Model. *Sensors* **2021**, *21*, 2989. [[CrossRef](#)]
17. Acevedo, A.; Alférez, S.; Merino, A.; Puigvi, L.; Rodellar, J. Recognition of peripheral blood cell images using convolutional neural networks. *Comput. Methods Programs Biomed.* **2019**, *180*, 105020. [[CrossRef](#)]
18. Habibzadeh, M.; Jannesari, M.; Rezaei, Z.; Baharvand, H.; Totonchi, M. Automatic white blood cell classification using pre-trained deep learning models: ResNet and Inception. In *Proceedings of the Tenth International Conference on Machine Vision (ICMV 2017)*, Vienna, Austria, 13–15 November 2018; Zhou, J., Radeva, P., Nikolaev, D., Verikas, A., Eds.; SPIE: Vienna, Austria, 2018; Volume 10696, p. 1069612. [[CrossRef](#)]
19. Wang, Q.; Bi, S.; Sun, M.; Wang, Y.; Wang, D.; Yang, S. Deep learning approach to peripheral leukocyte recognition. *PLoS ONE* **2019**, *14*, e0218808. [[CrossRef](#)]
20. Loey, M.; Naman, M.; Zayed, H. Deep Transfer Learning in Diagnosing Leukemia in Blood Cells. *Computers* **2020**, *9*, 29. [[CrossRef](#)]
21. Huang, X.; Liu, J.; Yao, J.; Wei, M.; Han, W.; Chen, J.; Sun, L. Deep-Learning Based Label-Free Classification of Activated and Inactivated Neutrophils for Rapid Immune State Monitoring. *Sensors* **2021**, *21*, 512. [[CrossRef](#)]
22. Khan, A.; Eker, A.; Chefranov, A.; Demirel, H. White blood cell type identification using multi-layer convolutional features with an extreme-learning machine. *Biomed. Signal Process. Control.* **2021**, *69*, 102932. [[CrossRef](#)]
23. Reena, M.R.; Ameer, P.M. Localization and recognition of leukocytes in peripheral blood: A deep learning approach. *Comput. Biol. Med.* **2020**, *126*, 104034. [[CrossRef](#)]
24. Alam, M.M.; Islam, M.T. Machine learning approach of automatic identification and counting of blood cells. *Heal. Technol. Lett.* **2019**, *17*, 103–108. [[CrossRef](#)] [[PubMed](#)]
25. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*, 1st ed.; MIT Press: Cambridge, MA, USA, 2016.
26. Xie, W.; Noble, J.A.; Zisserman, A. Microscopy cell counting and detection with fully convolutional regression networks. *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.* **2016**, *6*, 283292. [[CrossRef](#)]
27. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [[CrossRef](#)]
28. Zhao, Z.Q.; Zheng, P.; Xu, S.-T.; Wu, X. Object Detection With Deep Learning: A Review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [[CrossRef](#)] [[PubMed](#)]
29. Lin, T.Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21–26 July 2017; IEEE Computer Society: Washington, DC, USA, 2009; pp. 936–944.
30. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. In *Proceedings of the 2017 IEEE International Conference on Computer Vision*, Venice, Italy, 22–29 October 2017; IEEE Computer Society: Washington, DC, USA, 2017; pp. 2999–3007.
31. Peteiro-Barral, D.; Guijarro-Berdiñas, B. A Study on the Scalability of Artificial Neural Networks Training Algorithms Using Multiple-Criteria Decision-Making Methods. In *Lecture Notes in Computer Science, Proceeding of the Artificial Intelligence and Soft Computing (ICAISC 2013), Zakopane, Poland, 16–20 June 2019*; Springer: Heidelberg, Germany, 2019; pp. 162–173.
32. Gaiser, H.; de Vries, M.; Lacatusu, V. Keras RetinaNet. 2019. Available online: <https://github.com/fizyr/keras-retinanet/tree/0.5.1> (accessed on 27 May 2021).
33. Rezatofighi, S.H.; Soltanian-Zadeh, H. Automatic recognition of five types of white blood cells in peripheral blood. *Comput. Med. Imaging Graph.* **2011**, *35*, 333–343. [[CrossRef](#)]
34. Tzutalin. LabelImg. Git Code. 2015. Available online: <https://github.com/tzutalin/labelImg> (accessed on 20 July 2021).

35. Dvanesh, V.D.; Lakshmi, P.S.; Reddy, K.; Vasavi, A.S. Blood Cell Count using Digital Image Processing. In Proceedings of the 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), Tamil Nadu, India, 1–3 March 2018; pp. 1–7. [[CrossRef](#)]
36. Ruberto, C.; Loddo, A.; Putzu, L. Detection of red and white blood cells from microscopic blood images using a region proposal approach. *Comput. Biol. Med.* **2020**, *116*, 103530. [[CrossRef](#)] [[PubMed](#)]





# NanoForms: an integrated server for processing, analysis and assembly of raw sequencing data of microbial genomes, from Oxford Nanopore technology

Anna Czmil<sup>1,\*</sup>, Michal Wronski<sup>1,\*</sup>, Sylwester Czmil<sup>1</sup>, Marta Sochacka-Pietal<sup>2</sup>, Michal Cmil<sup>1</sup>, Jan Gawor<sup>3</sup>, Tomasz Wołkowicz<sup>4</sup>, Dariusz Plewczynski<sup>5,6</sup>, Dominik Strzalka<sup>1</sup> and Michal Pietal<sup>1</sup>

<sup>1</sup> Department of Complex Systems, Rzeszow University of Technology, Rzeszow, Subcarpathian, Poland

<sup>2</sup> Department of Biotechnology and Bioinformatics, Rzeszow University of Technology, Rzeszow, Subcarpathian, Poland

<sup>3</sup> DNA Sequencing and Oligonucleotide Synthesis Laboratory, Institute of Biochemistry and Biophysics, Polish Academy of Sciences, Warsaw, Masovian, Poland

<sup>4</sup> Department of Bacteriology and Biocontamination Control, National Institute of Public Health-National Institute of Hygiene, Warsaw, Masovian, Poland

<sup>5</sup> Laboratory of Functional and Structural Genomics, Centre of New Technologies, University of Warsaw, Warsaw, Masovian, Poland

<sup>6</sup> Laboratory of Bioinformatics and Computational Genomics, Warsaw University of Technology, Warsaw, Masovian, Poland

\* These authors contributed equally to this work.

## ABSTRACT

**Background.** Next Generation Sequencing (NGS) techniques dominate today's landscape of genetics and genomics research. Though Illumina still dominates worldwide sequencing, Oxford Nanopore is one of the leading technologies currently being used by biologists, medics and geneticists across various applications. Oxford Nanopore is automated and relatively simple for conducting experiments, but generates gigabytes of raw data, to be processed by often ambiguous set of alternative bioinformatics command-line tools, and genomics frameworks which require a knowledge of bioinformatics to run.

**Results.** We established an inter-collegiate collaboration across experimentalists and bioinformaticians in order to provide a novel bioinformatics tool, free for academics. This tool allows people without extensive bioinformatics knowledge to simply process their raw genome sequencing data. Currently, due to ICT resources' maintenance reasons, our server is only capable of handling small genomes (up to 15 Mb). In this paper, we introduce our tool, NanoForms: an intuitive and integrated web server for the processing and analysis of raw prokaryotic genome data, coming from Oxford Nanopore. NanoForms is freely available for academics at the following locations: <http://nanofoms.tech> (webserver) and <https://github.com/czmilanna/nanofoms> (GitHub source repository).

**Subjects** Bioinformatics, Genomics, Microbiology, Computational Science

**Keywords** NGS, Bioinformatics, Oxford Nanopore, Genomics, Webserver, DNA sequencing, DNA assembly, Microbial genomes

Submitted 23 August 2021  
Accepted 13 February 2022  
Published 29 March 2022

Corresponding author  
Michal Pietal, [m.pietal@prz.edu.pl](mailto:m.pietal@prz.edu.pl)

Academic editor  
Adam Witney

Additional Information and  
Declarations can be found on  
page 9

DOI [10.7717/peerj.13056](https://doi.org/10.7717/peerj.13056)

© Copyright  
2022 Czmil et al.

Distributed under  
Creative Commons CC-BY 4.0

**OPEN ACCESS**

**How to cite this article** Czmil A, Wronski M, Czmil S, Sochacka-Pietal M, Cmil M, Gawor J, Wołkowicz T, Plewczynski D, Strzalka D, Pietal M. 2022. NanoForms: an integrated server for processing, analysis and assembly of raw sequencing data of microbial genomes, from Oxford Nanopore technology. *PeerJ* 10:e13056 <http://doi.org/10.7717/peerj.13056>

## INTRODUCTION

Next Generation Sequencing technologies, such as Illumina ([Gloor et al., 2010](#)), Pacific BioSciences ([Rhoads & Au, 2015](#)) and Oxford Nanopore ([Jain et al., 2016](#)), dominate today's landscape of genetics and genomics research. Each technology has its advantages, disadvantages and market share in specific applications and niches. The applications of Oxford Nanopore technology include eDNA extraction and sequencing (*i.e.*, [Garlapati et al., 2019](#)), rapid viral sequencing (including the current challenge of SARS-CoV2 ([Wang et al., 2020](#))), human genome comparative sequencing ([De Coster et al., 2019](#)) and many others. Short-read sequencing technologies such as Illumina have made bacterial genome sequencing relatively cheap and accessible. However, the procedure of closing microbial genomes is often costly and laborious. Assembly of short reads from genomes that are repetitive and/or have extreme %GC content remains challenging. These difficulties can be mostly overcome by using single-molecule, long-read sequencing technologies such as the Oxford Nanopore. Nanopore helps with closing bacterial genomes ([Risse et al., 2015](#); [Kawalek et al., 2020](#)), can deliver two strategies for bacterial genome assembly ([Goldstein et al., 2019](#)), even helps to obtain complete bacterial chromosomes from microbiomes ([Moss, Maghini & Bhatt, 2020](#)) or is used in routine microbial genome sequencing ([Wick et al., 2017a](#)). Two main strategies are used to assemble bacterial genomes using long read sequencing. In the first, nanopore reads are used for long read only genome assembly followed by polishing with Illumina reads. Alternatively, long reads are used to enhance genome assemblies that are generated from short-read Illumina data. In such case nanopore reads can scaffold contigs generated by short read sequencing. It is also now possible to extract 3D structures of the genome using Oxford Nanopore ([Ulahannan et al., 2019](#)).

Oxford Nanopore is relatively user-friendly, easily operated, inexpensive, and can be simply adjusted to allow for rapid sample processing (including outdoor usage). The downside is, after the experiment is done, researchers are left with a huge amount of raw data. There is a wide variety of software tools available to perform taxonomic classification of the raw data. There are also many comparative studies that evaluate the top performing bioinformatics tools, provide recommendations for use cases, and show how to run these tools ([McIntyre et al., 2017](#); [Escobar-Zepeda et al., 2018](#); [Simon et al., 2019](#)). An inexperienced user, however, could easily become overwhelmed by the complexity of the data, the fast pace of tool development, and the version updates, command changes, installation problems, etc.

## MATERIALS & METHODS

We used the following technologies to create the NanoForms server: Python language, Linux/UNIX/BSD operating system, Django application server, Workflow Description Language and Cromwell, Crontab, Docker and BioContainers ([Da Veiga Leprevost et al., 2017](#)), and a custom set of bioinformatics tools. The NanoForms server is freely available for academic use and a commercial release of the server (for non-academics, businesses, etc.) is planned. We also provide its source code (under GPLv3 license) for non-commercial uses. The server is fully virtualized, with about 30 processor cores and 120 GB RAM available

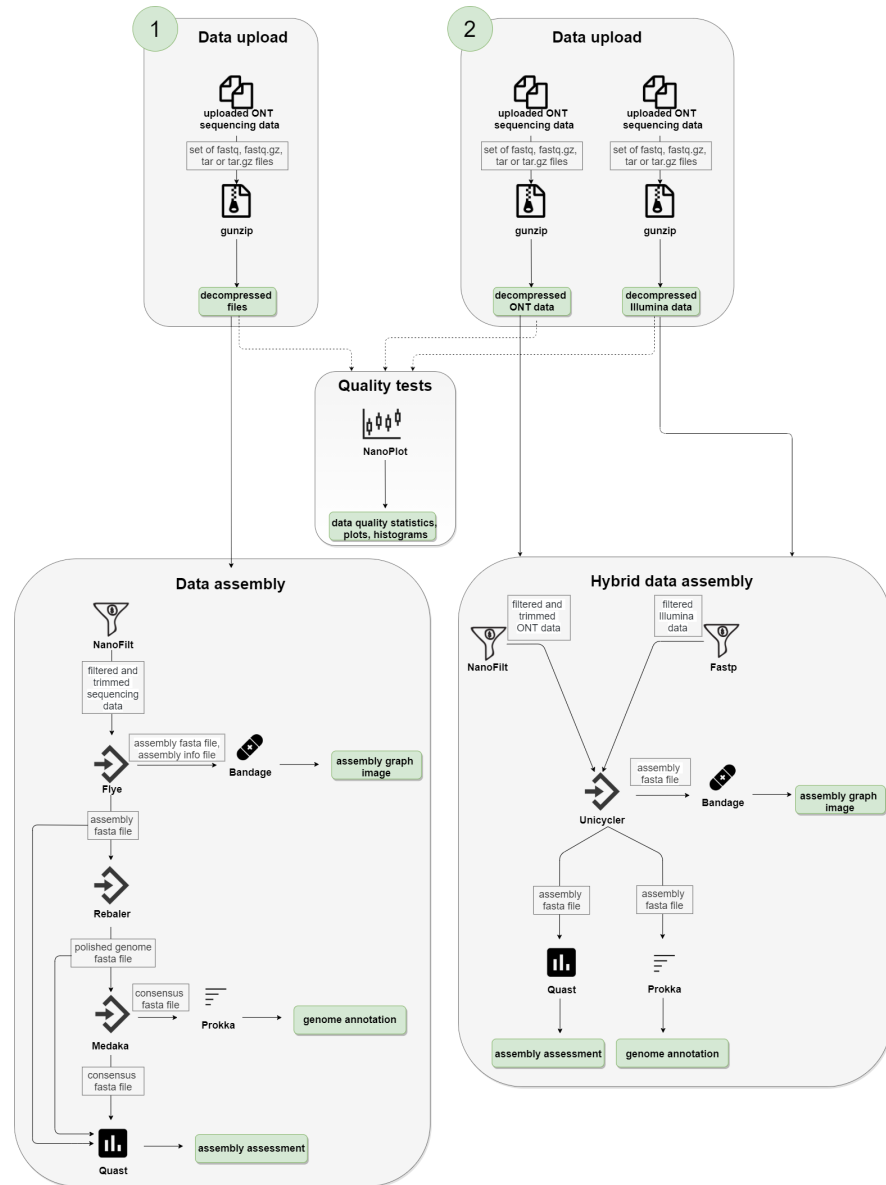
on average. The infrastructure is also hosted in a virtual environment. It can handle about 5-10 parallel jobs (taking into account dataset size limits of 15 GB). In general, single run processing time depends on configuration, sample size and the assembly type (short reads, long reads or both). It takes nearly two hours to process a 220 MB sample of ONT data, yielding the results. Computation time grows in a near linear manner, with a 1 GB dataset taking about five hours. Hybrid assemblies, which include a Unicycler polishing stage, can take up to 10 h for a 1 GB dataset combined (300 MB of ONT and 600 MB of Illumina data). However, actual run times will depend on server usage, so for complete control over timings one can install a local version of nanoforms. The detailed diagram of the server workflow is shown in Fig. 1.

The current version of the server includes the use of the following bioinformatics applications: Nanoplot 1.32.0 NanoFilt 2.7.1 (*De Coster et al., 2018*), FastQC 0.11.9 (<https://www.bioinformatics.babraham.ac.uk/projects/fastqc/>), Flye 2.8.1 (*Lin et al., 2016*), Bandage 0.8.1 (*Wick et al., 2015*), Rebaler 0.2.0 (<https://github.com/rrwick/Rebaler>), Medaka 1.0.1 (<https://github.com/nanoporetech/medaka>), Quast 3.2 (*Mikheenko et al., 2018*), Fitlong 0.1.0 (<https://github.com/rrwick/Fitlong>), Prokka 1.14.6 (*Seemann, 2014*), Kraken Tools 0.1 (*Davis et al., 2013*), Kraken 2.1.0 (*Wood, Lu & Langmead, 2019*) and Krona 2.7.1 (*Ondov, Bergman & Phillippy, 2011*). For hybrid genome assembly, we use Unicycler 0.3.0b (*Wick et al., 2017b*) and Fastp 0.18.0 (*Chen et al., 2018*) for filtering data from Illumina. A hybrid assembly strategy has been developed to overcome the limitations of both Illumina and Oxford Nanopore sequencing and to unlock their full potential for genome assembly. Oxford Nanopore long reads can scaffold contigs generated by Illumina short reads to disambiguate regions of the assembly graph that cannot be resolved by Illumina short reads alone, as implemented in the Unicycler assembler (*Chen, Erickson & Meng, 2020*).

The human genome comprises approximately 3 Gb of nucleotides while a typical raw data set from Oxford Nanopore sequencing (before base-calling) exceeds 1 TB. This makes it difficult to upload the data with even high-speed bandwidth (a 100 Mbps transfer would take over 24 h to transfer the data alone). In addition, such large amounts of data would require substantial funding for computing resources as the required calculations would be considered Big Data. Because of these technological issues, we narrowed the analyses to genomes of prokaryotic sizes (up to 15 Mb in length, up to 15 GB in file size) but our server also can handle small eukaryotic genomes (such as *S. cerevisiae*). Unicycler may not be the best tool for yeast genomes but Flye and nanopore assembly pipeline from NanoForms, might be easily used for small eukaryotic genomes (see: *Martín-Hernández et al., 2021*). After deploying this server and gathering remarks from the users, we are considering designing another milestone which might address this problem. We also plan to launch the commercial version of the server for institutional clients.

## RESULTS

We introduce NanoForms: an intuitive and integrated web server for the processing and analysis of raw data from small genomes, yielding from Oxford Nanopore technology. The



**Figure 1** NanoForms server workflow. Subsequent computation steps are run but in between, the user is given the partial results can take action, *i.e.*, give more specific parameters for the next programmes, based on the data available (*i.e.*, quality, quantity *etc.*). In the end, the report is generated and sent in a form of a PDF file.

Full-size DOI: [10.7717/peerj.13056/fig-1](https://doi.org/10.7717/peerj.13056/fig-1)

user uploads an archived, single sequence file (FASTQ) or a list of archived, sequence files, then the data is preprocessed. The user then chooses several options on the go and, after subsequent steps, the user obtains the DNA/RNA sequence in a form of FASTA file as well as the HTML summary with reports, images or statistics of the calculations performed.

The initial output from the NanoPlot program (read length vs. the read quality) gives the user a quick outlook (as shown in Fig. 2) that helps him or her decide whether to continue the analysis or to go back to the lab to fix the sample. The final output of the NanoForms service is an assembled genome in fasta format, prokka annotation files and Bandage diagram, allowing for easy graphical assessment of assembly completeness. An example, derived for the *Bacillus subtilis*, is depicted in Fig. 3. Since the MinION is marketed as needing only minor laboratory skills to operate, NanoForms needs practically no bioinformatics skills to produce the sequence (however more skilled users can benefit from extra, but optional, commands and options that they can provide during the course of the analysis). Therefore, we claim that our NanoForms server, in combination with Oxford Nanopore technology, has ultimately made NGS available for all, including both biologists and bioinformaticians, as specialized skills are no longer needed to perform certain NGS tasks and analyses. On the server website, after logging in, there are several toy datasets already provided to the user. Users can use these datasets for quality tests or data assembly using the respective forms provided in NanoForms, such as *Bacillus subtilis* SRX6978160.

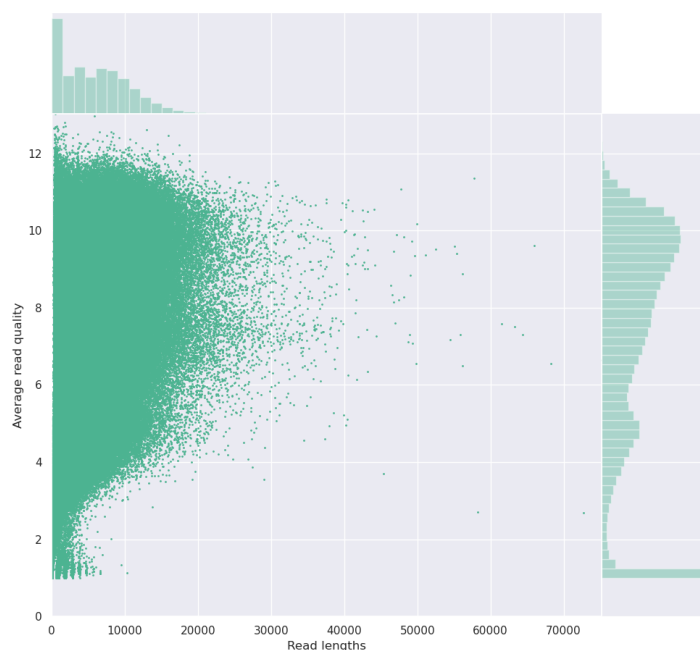
## DISCUSSION

We performed a detailed analysis and comparison of similar services available for researchers. The first service to be tested was the CGE (Larsen et al., 2017) server. We checked and tested this service at the very beginning of this project and at that time, it was an up-to-date and convenient service, and easy to use for biologists without technical knowledge. Shortly before drafting this manuscript, the part of the server dedicated to genome assembly went offline, so the only input is the contigs in FASTA (Pearson, 1990) format. The rest of the tools tested are free to use, but as standalone programs, they are not interconnected with the comprehensive pipeline. In addition, no figures are generated, which makes the qualitative analyses more difficult.

The Enterobase server (Zhou et al., 2020) is aimed at wgMLST analyses. This server is mainly designed for genotyping isolates. Users can screen the database against specific STRs etc., and the service can also generate phylogenetic trees. Enterobase is dedicated to analyses of gut bacteria and supports Illumina or PacBio reads. The user only needs to provide the FASTQ (Cock et al., 2010) files, which need to be compressed by the gzip tool and also, manually curated, before running the service. The figures can be generated or plotted, however this requires additional manual user intervention. The Enterobase server does not accept nanopore data.

Another interesting tool, though with significantly diminished accessibility, is the Galaxy Tools service (Cock et al., 2013). Our experience testing this service suggests that the stand-alone version of the server needs to be installed and run locally for optimal use, but for smaller analysis it can be also run on the public server. The server provides

## Read lengths vs Average read quality plot

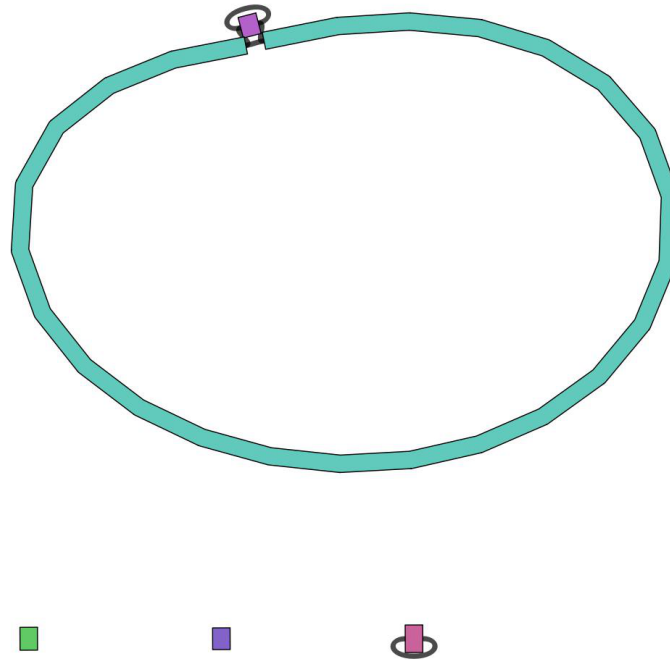


**Figure 2** An example histogram for read length vs. the read quality, as the output from the NanoPlot tool. Based on this information on the server website, the user can decide if the quality of the data is good enough to continue the analysis and thus, save time of the project or decide to which extent to crop the data to exclude low-quality short reads.

Full-size  DOI: [10.7717/peerj.13056/fig-2](https://doi.org/10.7717/peerj.13056/fig-2)

workflows (though only an empty set, for the novel user) which the user can adjust upon request. The tool supports both Illumina and long read (nanopore, PacBio) data input. It is worth noting that both the CGE and Galaxy Tools offer additional applications for practical genome analysis such as: resistance, serotype or virulence, and maintains the updated databases of these genetic targets.

While preparing and programming our tool, a software report was published about the new toolkit, NanoGalaxy, dedicated to the nanopore data processing (*De Koning et al., 2020*). NanoGalaxy is an extension of the aforementioned Galaxy Tools, in the area of support for nanopore data. In some ways NanoForms and NanoGalaxy seems to have similar features and functionalities. Similar to standard Galaxy Tools, NanoGalaxy seems to be more powerful but on the other hand it is aimed at more advanced user with more bioinformatics skills. In our subjective opinion NanoForms is easier to use for non-bioinformatics users but on the other hand can be treated as a quite bordered “black box”. NanoGalaxy is more complex and has more functionalities, but getting familiar with the numerous available algorithms requires some bioinformatics experience. NanoGalaxy and NanoSPC (*Xu et al., 2020*) deliver similar results and has similar capabilities as our server.



**Figure 3** An example of a circular sequence, being assembled at the end, as visualized by the Bandage tool. Sometimes the long-read nanopore data gives a few separate genome fragments as sequencing output (contigs), so the hybrid assembly option, provided by NanoForms, can often resolve these ambiguities. This image is normally the last stage of NanoForms sequencing protocol, however all figures and statistics that arose on the course of sequencing, are delivered to the user as a report.

Full-size  DOI: [10.7717/peerj.13056/fig-3](https://doi.org/10.7717/peerj.13056/fig-3)

NanoSPC is focused only on Nanopore data and as a result it is not possible to perform *e.g.*, hybrid assembly there. It is also focused mostly on metagenomics, identification of pathogens and variant calling, but it seems to be easy to use also for users with limited dry-lab knowledge.

Another tool we tested was Patric ([Gillespie et al., 2011](https://www.patricbiodata.org/)). This platform provides bioinformatics analyses of all bacteria, with special focus on pathogens. It supports hybrid genome assembly in the formula of short + long reads (PacBio and nanopore are supported). [Table 1](#) provides the comprehensive summary of the features about the quoted services, compared based on NanoForms functionality, in terms of nanopore data processing. We decided not to include “demultiplexing reads support” in our table as the most current version of MinKnow software (the native software to Oxford Nanopore) supports this feature already.

As we were pursuing this project in mid-2020, Oxford Nanopore announced the availability of their EPI2ME (<http://epi2me.nanoporetech.com/;requireslogin>) cloud-based workflow to process raw nanopore data. The platform’s intended use is only nanopore data analysis, without options for sequence assembly or trimming. After performing the

**Table 1** NanoForms server vs. other services: the comparison. Our service successfully fills in the gap for NGS genome assembly, in regard to the fully automated (but interactive) pipelined nanopore data processing.

Feature	CGE	Enterobase	Galaxy tools	EPI2ME	NanoPipe	Patric	Nano galaxy	NanoSPC	NanoForms
Raw data processing	+	+	+	+	+	+	+	+	+
Interactive interface	+	+	+	+	+	+	+	+	+
QA, reports	+	+	+	+	+	+	+	+	+
Free for academics	+	+	+	+	+	+	+	+	+
Sequence assembly	-	+	+	-	+ <sup>a</sup>	+	+	+	+
Qualitative analyses: images	-	+	+	+	+	+	+	+	+
Nanopore data processing	+	-	+	+	+	+	+	+	+
Tools connected into the pipeline	-	+	+/- <sup>b</sup>	-	-	+/- <sup>b</sup>	+	+	+
Hybrid assembly: nanopore + Illumina	-	-	+	-	-	+	+	-	+
Special nanopore visualization tools	-	-	+	+	-	-	+	+	+
Summary report generated	-	-	-	-	+	-	-	+	+
Ease of use <sup>c</sup>	+	+/-	+/-	+	+	+	+	+	+
Dry-lab knowledge unnecessary <sup>c</sup>	+	+	-	-	+	-	-	+	+

**Notes.**<sup>a</sup>But not the novo assembly.<sup>b</sup>There are no automatic pipelines, but users can develop them themselves.<sup>c</sup>Subjective assessments of the authors.

base calling, reads are uploaded to the server *via* the EPI2ME Agent, which is a stand-alone program. The user can pick one of the several workflows (ie. microbiological classification or human genome analysis) which are triggered and executed in real time. Also in the table, the reader might find the characteristics of the last tool we reviewed: NanoPipe (*Shabardina et al., 2019*).

## CONCLUSIONS

In summary, our NanoForms server, freely available for all academics, bridges the high-speed of prokaryotic genome assembly with an intuitive, interactive interface. According to the Oxford Nanopore MinION specification product page, it is sufficient for the user to have a mid-range laptop and the device to obtain the sequence of the sample. No further resources are needed and the user can continue the genomic analyses after a short break in sequencing with the use of NanoForms service.

## ACKNOWLEDGEMENTS

MP wants to thank Przemyslaw Wroblewski and Giovanni Mazzocco who (with Dariusz Plewczynski) pioneered the NanOnline web server at CeNT University of Warsaw, implementing the initial idea for nanopore genome assemblies of human genomes preceding the release of the NanoForms server. MP would also like to thank to Kacper Kroczek for additional testing and for checking documentation consistency on the server



homepage. In addition, MP would like to thank Michal Madera and the SoftSystem Sp. z o.o. (Ltd.) company for providing some parts of the code necessary to the server and for delivering auxiliary server infrastructure.

## ADDITIONAL INFORMATION AND DECLARATIONS

### Funding

The work of Dominik Strzalka, Michal Pietal, Michal Cmil, Marta Sochacka-Pietal, Michal Wronski and Anna Czmil as well as the paper itself, was financed by Subcarpathian Center for Innovation (Podkarpackie Centrum Innowacyjności - PCI), Teofila Lenartowicza 4 Street, 35-051 Rzeszów, Poland, under the grant no. F3\_116 contract no. 05/PRZ/1/DG/PCI/2019. “Oxford Nanopore technology: optimization of enzymes and analysis of genomic data for commercial applications”. Michal Pietal and Dariusz Plewczynski were supported by the National Science Center grant (2018/02/X/NZ2/00622) “Identification of structural variants in the human genome using long fragments from next generation sequencing, based on Oxford Nanopore technology”. Dariusz Plewczynski was supported by Polish National Science Centre (2019/35/O/ST6/02484), and the Foundation for Polish Science co-financed by the European Union under the European Regional Development Fund (TEAM to Dariusz Plewczynski) “Three-dimensional Human Genome structure at the population scale: computational algorithm and experimental validation for lymphoblastoid cell lines of selected families from 1000 Genomes Project”. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

### Grant Disclosures

The following grant information was disclosed by the authors:

Subcarpathian Center of Innovation (PCI): 05/PRZ/1/DG/PCI/2019.

Oxford Nanopore technology: optimization of enzymes and analysis of genomic data for commercial applications.

National Science Center: 2018/02/X/NZ2/00622.

Identification of structural variants in the human genome using long fragments from next generation sequencing, based on Oxford Nanopore technology.

Polish National Science Centre: 2019/35/O/ST6/02484.

Foundation for Polish Science co-financed by the European Union under the European Regional Development Fund (TEAM to Dariusz Plewczynski).

Three-dimensional Human Genome structure at the population scale: computational algorithm and experimental validation for lymphoblastoid cell lines of selected families from 1000 Genomes Project.

### Competing Interests

The authors declare there are no competing interests.

### Author Contributions

- Anna Czmil, Sylwester Czmil and Michal Cmil analyzed the data, prepared figures and/or tables, application programming, and approved the final draft.

- Michal Wronski analyzed the data, authored or reviewed drafts of the paper, application programming, and approved the final draft.
- Marta Sochacka-Pietal conceived and designed the experiments, performed the experiments, analyzed the data, authored or reviewed drafts of the paper, and approved the final draft.
- Tomasz Wołkowicz analyzed the data, prepared figures and/or tables, and approved the final draft.
- Dariusz Plewczynski and Dominik Strzalka conceived and designed the experiments, authored or reviewed drafts of the paper, and approved the final draft.
- Michal Pietal conceived and designed the experiments, authored or reviewed drafts of the paper, application programming, and approved the final draft.

### Data Availability

The following information was supplied regarding data availability:

The source code of the NanoForms server is available at Github: <https://github.com/czmilanna/nanoforms>. This is the source for standalone server installation. The server is available at <https://nanoforms.tech/>.

The *Bacillus subtilis* sequences are available at ENA: [SRX6978160](https://ena.ebi.ac.uk/ena/record/SRX6978160).

## REFERENCES

- Chen Z, Erickson DL, Meng J. 2020.** Benchmarking hybrid assembly approaches for genomic analyses of bacterial pathogens using Illumina and Oxford Nanopore sequencing. *BMC Genomics* **21**(1):1–21 DOI [10.1186/s12864-019-6419-1](https://doi.org/10.1186/s12864-019-6419-1).
- Chen S, Zhou Y, Chen Y, Gu J. 2018.** fastp: an ultra-fast all-in-one FASTQ preprocessor. *Bioinformatics* **34**(17):i884–i890 DOI [10.1093/bioinformatics/bty560](https://doi.org/10.1093/bioinformatics/bty560).
- Cock PJ, Fields CJ, Goto N, Heuer ML, Rice PM. 2010.** The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Research* **38**(6):1767–1771 DOI [10.1093/nar/gkp1137](https://doi.org/10.1093/nar/gkp1137).
- Cock PJ, Grüning BA, Paszkiewicz K, Pritchard L. 2013.** Galaxy tools and workflows for sequence analysis with applications in molecular plant pathology. *PeerJ* **1**:e167 DOI [10.7717/peerj.167](https://doi.org/10.7717/peerj.167).
- Da Veiga Leprevost F, Grüning BA, Alves Aflitos S, Röst HL, Uszkoreit J, Barsnes H, Vaudel M, Moreno P, Gatto L, Weber J, Bai M. 2017.** BioContainers: an open-source and community-driven framework for software standardization. *Bioinformatics* **33**(16):2580–2582 DOI [10.1093/bioinformatics/btx192](https://doi.org/10.1093/bioinformatics/btx192).
- Davis MP, Van Dongen S, Abreu-Goodger C, Bartonicek N, Enright AJ. 2013.** Kraken: a set of tools for quality control and analysis of high-throughput sequence data. *Methods* **63**(1):41–49 DOI [10.1016/j.ymeth.2013.06.027](https://doi.org/10.1016/j.ymeth.2013.06.027).
- De Coster W, D’Hert S, Schultz DT, Cruts M, Van Broeckhoven C. 2018.** NanoPack: visualizing and processing long-read sequencing data. *Bioinformatics* **34**(15):2666–2669 DOI [10.1093/bioinformatics/bty149](https://doi.org/10.1093/bioinformatics/bty149).

- De Coster W, De Rijk P, De Roeck A, De Pooter T, D’Hert S, Strazisar M, Slegers K, Van Broeckhoven C. 2019. Structural variants identified by Oxford Nanopore PromethION sequencing of the human genome. *Genome Research* **29**(7):1178–1187 DOI [10.1101/gr.244939.118](https://doi.org/10.1101/gr.244939.118).
- De Koning W, Miladi M, Hiltemann S, Heikema A, Hays J, Flemming S, Van den Beek M, Mustafa D, Backofen R, Grüning B, Stubbs A. 2020. NanoGalaxy: nanopore long-read sequencing data analysis in Galaxy. *GigaScience* **10**(9):giaa105.
- Escobar-Zepeda A, Godoy-Lozano EE, Raggi L, Segovia L, Merino E, Gutiérrez-Rios RM, Juarez K, Licea-Navarro AF, Pardo-Lopez L, Sanchez-Flores A. 2018. Analysis of sequencing strategies and tools for taxonomic annotation: defining standards for progressive metagenomics. *Scientific Reports* **8**(1):1–13.
- Garlapati D, Charankumar B, Ramu K, Madeswaran P, Murthy MR. 2019. A review on the applications and recent advances in environmental DNA (eDNA) metagenomics. *Reviews in Environmental Science and Bio/Technology* **18**(3):389–411 DOI [10.1007/s11157-019-09501-4](https://doi.org/10.1007/s11157-019-09501-4).
- Gillespie JJ, Wattam AR, Cammer SA, Gabbard JL, Shukla MP, Dalay O, Driscoll T, Hix D, Mane SP, Mao C, Nordberg EK. 2011. PATRIC: the comprehensive bacterial bioinformatics resource with a focus on human pathogenic species. *Infection and Immunity* **79**(11):4286–4298 DOI [10.1128/IAI.00207-11](https://doi.org/10.1128/IAI.00207-11).
- Gloor GB, Hummelen R, Macklaim JM, Dickson RJ, Fernandes AD, MacPhee R, Reid G. 2010. Microbiome profiling by illumina sequencing of combinatorial sequence-tagged PCR products. *PLOS ONE* **5**(10):e15406 DOI [10.1371/journal.pone.0015406](https://doi.org/10.1371/journal.pone.0015406).
- Goldstein S, Beka L, Graf J, Klassen JL. 2019. Evaluation of strategies for the assembly of diverse bacterial genomes using MinION long-read sequencing. *BMC Genomics* **20**(1):1–17 DOI [10.1186/s12864-018-5379-1](https://doi.org/10.1186/s12864-018-5379-1).
- Jain M, Olsen HE, Paten B, Akeson M. 2016. The Oxford Nanopore MinION: delivery of nanopore sequencing to the genomics community. *Genome Biology* **17**(1):239 DOI [10.1186/s13059-016-1103-0](https://doi.org/10.1186/s13059-016-1103-0).
- Kawalek A, Kotecka K, Modrzejewska M, Gawor J, Jagura-Burdzy G, Bartosik AA. 2020. Genome sequence of *Pseudomonas aeruginosa* PAO1161, a PAO1 derivative with the ICE Pae 1161 integrative and conjugative element. *BMC Genomics* **21**(1):1–12 DOI [10.1186/s12864-019-6419-1](https://doi.org/10.1186/s12864-019-6419-1).
- Larsen MV, Joensen KG, Zankari E, Ahrenfeldt J, Lukjancenko O, Kaas RS, Roer L, Leekitcharoenphon P, Saputra D, Cosentino S, Thomsen MCF. 2017. The CGE tool box. In: *Applied genomics of foodborne pathogens*. Cham: Springer, 65–90.
- Lin Y, Yuan J, Kolmogorov M, Shen MW, Chaisson M, Pevzner P. 2016. Assembly of long error-prone reads Using de Bruijn graphs. *Proceedings of the National Academy of Sciences of the United States of America* **113**(52):E8396–E8405.
- Martín-Hernández GC, Müller B, Chmielarz M, Brandt C, Hölzer M, Viehweger A, Passoth V. 2021. Chromosome-level genome assembly and transcriptome-based annotation of the oleaginous yeast *Rhodotorula toruloides* CBS 14. *bioRxiv*.

- McIntyre AB, Ounit R, Afshinnekoo E, Prill RJ, Hénaff E, Alexander N, Minot SS, Danko D, Foux J, Ahsanuddin S, Tighe S. 2017. Comprehensive benchmarking and ensemble approaches for metagenomic classifiers. *Genome Biology* **18**(1):1–19 DOI [10.1186/s13059-016-1139-1](https://doi.org/10.1186/s13059-016-1139-1).
- Mikheenko A, Prjibelski A, Saveliev V, Antipov D, Gurevich A. 2018. Versatile genome assembly evaluation with QUAST-LG. *Bioinformatics* **34**(13):i142–i150 DOI [10.1093/bioinformatics/bty266](https://doi.org/10.1093/bioinformatics/bty266).
- Moss EL, Maghini DG, Bhatt AS. 2020. Complete, closed bacterial genomes from microbiomes using nanopore sequencing. *Nature Biotechnology* **38**(6):701–707 DOI [10.1038/s41587-020-0422-6](https://doi.org/10.1038/s41587-020-0422-6).
- Ondov BD, Bergman NH, Phillippy AM. 2011. Interactive metagenomic visualization in a Web browser. *BMC Bioinformatics* **12**(1):1–10 DOI [10.1186/1471-2105-12-1](https://doi.org/10.1186/1471-2105-12-1).
- Pearson WR. 1990. Rapid and sensitive sequence comparison with FASTP and FASTA. *Methods Enzymol* **183**:63–98 DOI [10.1016/0076-6879\(90\)83007-v](https://doi.org/10.1016/0076-6879(90)83007-v).
- Rhoads A, Au KF. 2015. PacBio sequencing and its applications. *Genomics, Proteomics & Bioinformatics* **13**(5):278–289 DOI [10.1016/j.gpb.2015.08.002](https://doi.org/10.1016/j.gpb.2015.08.002).
- Risse J, Thomson M, Patrick S, Blakely G, Koutsovoulos G, Blaxter M, Watson M. 2015. A single chromosome assembly of *Bacteroides fragilis* strain BE1 from Illumina and MinION nanopore sequencing data. *Gigascience* **4**(1):s13742–015.
- Seemann T. 2014. Prokka: rapid prokaryotic genome annotation. *Bioinformatics* **30**(14):2068–2069 DOI [10.1093/bioinformatics/btu153](https://doi.org/10.1093/bioinformatics/btu153).
- Shabardina V, Kischka T, Manske F, Grundmann N, Frith MC, Suzuki Y, Makołowski W. 2019. NanoPipe—a web server for nanopore MinION sequencing data analysis. *GigaScience* **8**(2):gij169.
- Simon HY, Siddle KJ, Park DJ, Sabeti PC. 2019. Benchmarking metagenomics tools for taxonomic classification. *Cell* **178**(4):779–794 DOI [10.1016/j.cell.2019.07.010](https://doi.org/10.1016/j.cell.2019.07.010).
- Ulahannan N, Pendleton M, Deshpande A, Schwenk S, Behr JM, Dai X, Tyler C, Rughani P, Kudman S, Adney E, Tian H. 2019. Nanopore sequencing of DNA concatemers reveals higher-order features of chromatin structure. *bioRxiv* 833590.
- Wang M, Fu A, Hu B, Tong Y, Liu R, Liu Z, Gu J, Xiang B, Liu J, Jiang W, Shen G. 2020. Nanopore targeted sequencing for the accurate and comprehensive detection of SARS-CoV-2 and other respiratory viruses. *Small* **16**(32):2002169 DOI [10.1002/sml.202002169](https://doi.org/10.1002/sml.202002169).
- Wick RR, Judd LM, Gorrie CL, Holt KE. 2017a. Completing bacterial genome assemblies with multiplex MinION sequencing. *Microbial Genomics* **3**(10):e000132.
- Wick RR, Judd LM, Gorrie CL, Holt KE. 2017b. Unicycler: resolving bacterial genome assemblies from short and long sequencing reads. *PLOS Computational Biology* **13**(6):e1005595 DOI [10.1371/journal.pcbi.1005595](https://doi.org/10.1371/journal.pcbi.1005595).
- Wick RR, Schultz MB, Zobel J, Holt KE. 2015. Bandage: interactive visualisation of de novo genome assemblies. *Bioinformatics* **31**(20):3350–3352 DOI [10.1093/bioinformatics/btv383](https://doi.org/10.1093/bioinformatics/btv383).

- Wood DE, Lu J, Langmead B. 2019.** Improved metagenomic analysis with Kraken 2. *Genome Biology* **20(1)**:257 DOI [10.1186/s13059-019-1891-0](https://doi.org/10.1186/s13059-019-1891-0).
- Xu Y, Yang-Turner F, Volk D, Crook D. 2020.** NanoSPC: a scalable, portable, cloud compatible viral nanopore metagenomic data processing pipeline. *Nucleic Acids Research* **48(W1)**:W366–W371 DOI [10.1093/nar/gkaa413](https://doi.org/10.1093/nar/gkaa413).
- Zhou Z, Alikhan NF, Mohamed K, Fan Y, Achtman M, Brown D, Chattaway M, Dallman T, Delahay R, Kornschöber C, Pietzka A. 2020.** The EnteroBase user's guide, with case studies on Salmonella transmissions, Yersinia pestis phylogeny, and Escherichia core genomic diversity. *Genome Research* **30(1)**:138–152 DOI [10.1101/gr.251678.119](https://doi.org/10.1101/gr.251678.119).





Contents lists available at ScienceDirect

SoftwareX

journal homepage: [www.elsevier.com/locate/softx](http://www.elsevier.com/locate/softx)

Original software publication

# GPR: A Python implementation of an extremely simple classifier based on fuzzy logic and gene expression programming



Anna Czmil<sup>\*</sup>, Jacek Kluska, Sylwester Czmil

The Faculty of Electrical and Computer Engineering, Rzeszow University of Technology, Powstancow Warszawy 12, Rzeszow, 35-959, Poland

## ARTICLE INFO

### Article history:

Received 12 July 2022

Received in revised form 13 December 2022

Accepted 8 March 2023

### Keywords:

Fuzzy rule-based classifier

Gene expression programming

Interpretability

## ABSTRACT

In this work, we present a Python-based implementation of an extremely simple classifier (GPR), which combines gene expression programming (GEP) features and the algebraic representation of the 'if-then' fuzzy rules theory of the Takagi–Sugeno fuzzy inference system. Generated fuzzy metarules are highly interpretable and suitable for many applications. We provide an open-source Python implementation of the GPR algorithm to enable the use of the algorithm without any commercial software tools and open access to the research community. We also added enhancements to improve the readability and interpretability of the rules.

© 2023 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

## Code metadata

Current code version	v1.0.0
Permanent link to code/repository used for this code version	<a href="https://github.com/ElsevierSoftwareX/SOFTX-D-22-00195">https://github.com/ElsevierSoftwareX/SOFTX-D-22-00195</a>
Code Ocean compute capsule	<a href="https://codeocean.com/capsule/6784302/tree/v1">https://codeocean.com/capsule/6784302/tree/v1</a>
Legal Code License	MIT
Code versioning system used	git
Software code languages, tools, and services used	Python, Deap, Geppy, Numpy
Compilation requirements, operating environments & dependencies	python ≥ 3.8
If available Link to developer documentation/manual	<a href="https://gpr-algorithm.readthedocs.io/en/latest/">https://gpr-algorithm.readthedocs.io/en/latest/</a>
Support email for questions	<a href="mailto:czmilanna@gmail.com">czmilanna@gmail.com</a>

## 1. Motivation and significance

Classification is a data mining technique that is applied to predict the membership of groups for data instances. There are various classification methods to solve classification problems. The most popular classification algorithms are logistic regression (LR), naive Bayes (NB), k-nearest neighbors (KNN), decision tree (DT), support vector machines (SVM), gene expression programming (GEP) [1,2], etc. However, many classification methods do not provide intelligible fuzzy or non-fuzzy classification rules. In this paper, we are particularly interested in fuzzy rule-based systems (FRBs) that act as classifiers automatically generated from data using GEP methods.

The fundamental problem concerning classifier design is constructing accurate and interpretable (transparent) models. It is

necessary to reach a compromise between the interpretability of its rules and the accuracy [3]. Measures of the accuracy of such systems are straightforward and well known. However, interpretability acknowledged as the main advantage of fuzzy rule-based systems is challenging to measure. It depends on several factors, mainly the model structure, the shape of the membership functions of fuzzy sets, the number of rules, the number of features, the number of linguistic terms, etc. The choice of appropriate interpretability measures remains an open problem [4].

Many algorithms have been proposed in the literature to construct fuzzy rule-based classifiers [5–10]. They produce more or less complicated fuzzy rules. However, the models built by these algorithms do not employ two fuzzy sets with linear membership functions, as GPR does. Usually, they use more fuzzy sets with complicated membership functions, mainly when they come from genetic programming-based methods. Similarly to [11,12], the GPR algorithm does not adjust the membership functions of fuzzy

<sup>\*</sup> Corresponding author.

E-mail address: [czmilanna@gmail.com](mailto:czmilanna@gmail.com) (Anna Czmil).

<https://doi.org/10.1016/j.softx.2023.101362>

2352-7110/© 2023 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

sets because their adjustment can degrade the comprehensibility of fuzzy rule-based systems. Therefore, GPR can be viewed as an extremely simple algorithm since it uses only two fuzzy sets with linear and complementary membership functions for every continuous feature.

The article [13] proposed the design of a very simple binary data-driven classifier called GPR. Its performance was tested on 16 datasets and compared with 22 other classification algorithms. The results show that this classifier is among the best classifiers when the quality criterion is the area under the ROC curve and the classification accuracy. This classifier is based on fuzzy rules and has the following features:

1. The modeled dataset may contain continuous and categorical input variables (features),
2. The number of features in the dataset does not affect the complexity of the metarules,
3. The classification procedure provides highly interpretable fuzzy metarules that are equivalent to algebraic expressions,
4. The algebraic expressions are obtained using the GEP technique,
5. The user can set a priori the number of metarules and their complexity.

The above advantages make the GPR classifier attractive for many applications, e.g., finding classification rules for medical datasets [14–16]. The main drawback of the GPR implementation described in [13] is that it relies on commercial software tools: GeneXproTools and GeneXproServer from Gepsoft Limited (<https://www.gepsoft.com/>). Furthermore, the user must manually extract metarules from the entire set of solutions in the form of algebraic equations. Extracting metarules is a simple task, but it requires commitment on the user's part. Therefore, we have implemented this algorithm in Python and provided a freeware version to users instead of commercial software. Additionally, we have made modifications/improvements to the original GPR algorithm so that the software user gets the following additional benefits:

1. The process of selecting linguistic “if-then” metarules is entirely automatic; the user only provides data records containing real numbers from the interval [0,1] and/or labels from the set {0,1},
2. The antecedents of the generated metarules may contain additional linguistic terms “low”, “high”, “medium” and the linguistic hedge “very”, which are logically comprehensible [17],
3. Each rule's support (certainty or confidence factor) is generated.

## 2. Software description

### 2.1. The GPR algorithm description

A comprehensive study of the GPR algorithm was described in [13]. Below, we briefly describe the idea of this algorithm by assuming the most interesting case, when the original dataset contains  $n$ -dimensional real input vectors (data records) with coordinates (features), say,  $y_k$ , ( $k = 1, \dots, n$ ), from finite intervals. All input vectors should be transformed into distinct points in a hypercube  $I^n = [0, 1]^n$ . Let us assume that the P1-TS fuzzy rule-based system models our dataset and consists of several fuzzy “if-then” metarules [18–20]. The metarule is equivalent to many single “if-then” fuzzy rules, where the antecedent of any single rule refers to all input variables  $y_1, \dots, y_n$ . In contrast, the antecedent of the metarule refers to the proper subset of the set

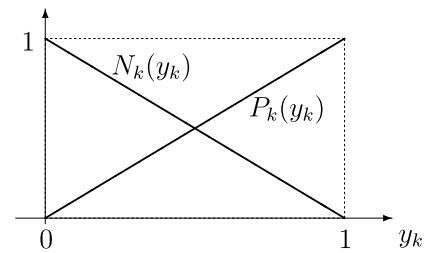


Fig. 1. Membership functions of fuzzy sets defined for continuous normalized inputs.

$\{y_1, \dots, y_n\}$ . Every feature  $y_k$  used in a rule or metarule refers to one of two fuzzy sets (linguistic variables). The membership function of the first fuzzy set is  $P_k(y_k) = y_k$ , and the second is  $N_k(y_k) = 1 - P_k(y_k)$ , for  $k = 1, \dots, n$ , (see Fig. 1).

If all consequents of the metarules of the considered P1-TS system are from the set  $\{0, 1\}$ , then all system variables can be easily interpreted from the multivalued logic point of view. For example, if  $y_k \leq \theta$ , where  $\theta$  is a threshold (usually,  $\theta = 0.5$ ), then  $y_k$  is interpreted as “almost false” (Low or L); otherwise, as “almost true” (High or H). An insightful theorem is presented in [13]. It says that, for the inputs  $y_k \in [0, 1]$  of the P1-TS system, after transforming all input variables from the set  $\{y_1, \dots, y_n\}$  into the set of new inputs:  $\{x_1, \dots, x_{2n}\}$ , such that  $x_{2k-1} = y_k$ , and  $x_{2k} = 1 - y_k$ , for  $k = 1, \dots, n$ , the crisp output  $S$  of this system can be expressed by the sum of products of variables “ $x_{(\cdot)}$ ” as follows

$$S = \sum_{r=1}^M \prod_{k \in K_r} x_k, \tag{1}$$

where the products  $\prod_{k \in K_r} x_k$  refer to the continuous features of such data records that correspond to the class label “1”. The subsets  $K_1, \dots, K_M \subset \{1, \dots, 2n\}$  contain some indices, and usually,  $1 \leq M \leq 2^{2n} - 1$ . Additionally, any algebraic expression in the form of (1) can be interpreted as a system of metarules that defines some P1-TS rule-based system. Therefore, the main problem we need to solve is finding an expression in the form of Eq. (1) for a given dataset. It should be noted that the number of possible solutions to our problem in the form of (1) is enormous, so we propose to use the GEP algorithm [2] to solve it. Our data may also contain records with categorical attributes (labels). However, we do not discuss this (simpler) case, as the details appear in [13]. In contrast to [13], we do not use a commercial implementation of the GEP algorithm in this contribution.

### 2.2. Software functionalities

GPR is written in Python 3 and uses the Deap and Geppy evolutionary computation frameworks and the NumPy numerical package [21–23]. The core functionality of the algorithm is implemented within the GPR class, which allows you to create an instance of the class and call the two most important methods, i.e., fit() and predict(). These methods follow the Scikit-learn library interface and are applied successively to train the algorithm and predict using the already trained algorithm. Since the results are returned similarly, it is possible to assess the prediction error with the methods from the module sklearn.metrics. The GPR class contains a private method \_init\_primitive\_set() that produces a set of possible linguistic terms “is\_high” and “is\_low” for each attribute. These attributes are multiplied together to form a gene. Many genes are linked to a chromosome, which is the sum of



**Table 1**  
Inputs and their complements expressed in fuzzy logic.

z1	z2	y1 is Low	y2 is Low	y1 is High	y2 is High	label
-1.000	2.500	1.000	0.900	0.000	0.100	1
2.500	2.000	0.125	1.000	0.875	0.000	0
3.000	7.000	0.000	0.000	1.000	1.000	1
-0.200	6.300	0.800	0.140	0.200	0.860	1
0.500	5.000	0.625	0.400	0.375	0.600	1

the values of these genes using `generate_chromosome()`. Another private method is `_init_generate_population_function()` which initializes the functions that Geppy invokes to generate genes, chromosomes, and entire populations. The method `_init_toolbox()` is responsible for initializing all available operations, i.e., mutations, selection, and crossovers applied by Geppy to modify, select, and move the individuals. `_init_stats()` register all possible statistical functions that will be applied to the data in each subsequent generation to observe the results of individual populations. `_init_evaluation_function()` initializes the evaluation function used to assess the degree of fit of the chromosome to the input data. The next two key functions are `_compile_chromosome()` and `_compliment_samples()`. The first of them is used to translate a chromosome in the form of literals into a Python function that can be used for classification. The second generates the complements of the input variables from the final dataset. The property `_best_fit()` returns the most fit individual that ever lived in the population during evolution; `_best_fit_function()` is a combination of the property `_best_fit()` and the function `_compile_chromosome()`. This combined function can be used for classification based on the most suitable individual. The functions `_shorten_terminals()` and `_translate_terminal()` are used to truncate terminals in rules using the linguistic term “medium” and hedge “very” and translate literals into human-readable markings in the form of linguistic “if-then” rules. The function `ranking()` counts the occurrences of each of the attributes of `_best_fit()` and generates a ranking of these attributes. The function `rules()` generates rules based on `_best_fit()`.

### 3. Illustrative examples

#### 3.1. Example 1

To demonstrate how the GPR algorithm works, let us consider five raw data records described by two input variables  $(z_1, z_2) \in [-1, 3] \times [2, 7] \subset \mathbb{R}^2$  which are labeled by the class  $C1 \in \{0, 1\}$  as described in [13]. Every data record was rescaled into the range  $[0, 1]$  using the object `MinMaxScaler` from Scikit-learn. Table 1 shows the complemented inputs for normalized inputs which have the following form:  $Low(y) = 1 - y$  and  $High(y) = y$  for  $y \in [0, 1]$ .

We apply GPR to generate the rules shown in Listing 1. First, we initialize the random number generator with a seed of 1 to ensure repeatability. Then, we define labels and attributes according to Table 1. The attributes are then re-scaled to the range  $[0, 1]$  using the object `MinMaxScaler`, because the GPR algorithm requires normalized data. To create a new instance of GPR, the `feature_names` parameter is required. Other parameters such as `max_n_of_rules`, `max_n_of_and` or `verbose` sequentially refer to the settings of the maximum number of rules, the maximum number of AND operators in a single rule, and whether or not to print the statistics. They are optional and have default values (see documentation at <https://gpr-algorithm.readthedocs.io/>). The user can also set other parameters:

- `target_names` that display the class names in the rules instead of the generic 0 and 1;

- `n_populations` and `n_generations`, which represent the number of generations and the number of populations, respectively;
- `threshold` that converts a score into a prediction; if the score is greater than the threshold (set as default to 0.5), we predict 1; otherwise, we predict 0;
- `base_pb` that indicates the probability of triggering an operation on the chromosome;
- `eval_fun` is used in a fitness function to calculate the match of the chromosome with the training data in the same way described in the original article. The user can change it to his own function, which takes two arrays as input parameters; the first consists of ground truth (correct) labels and the second one of predicted labels. This function returns a single number representing the score.

GPR has a similar interface to other classification algorithms available in Scikit-learn, i.e., have the `fit()` and `predict()` methods in the same format. The method `fit()` trains the GPR model according to the training data. It requires a list of normalized attributes and a list of labels and returns the fitted model. The method `predict()` performs classification on normalized samples and returns class labels for those attributes. The method `rules()` returns linguistic “if-then” metarules, which can be printed on the console after completing the fitting. The results obtained by GPR can be evaluated using methods from the module `sklearn.metrics`, e.g. `accuracy_score()`, `auc()`, etc.

**Listing 1:** An example usage of GPR on artificial data.

```

1 import random
2 import numpy as np
3 from sklearn.metrics import accuracy_score
4 from sklearn.preprocessing import MinMaxScaler
5 from gpr_algorithm import GPR
6
7 random.seed(1)
8
9 labels = np.array(
10     [1, 0, 1, 1, 1]
11 )
12 attributes = np.array(
13     [[-1.0, 2.5], [2.5, 2.0], [3.0, 7.0], [-0.2, 6.3],
14      [0.5, 5.0]]
15 )
16 attributes_normalized = MinMaxScaler().fit_transform(
17     attributes)
18
19 gpr = GPR(
20     feature_names=['y1', 'y2'],
21     max_n_of_rules=2, max_n_of_and=2,
22     verbose=False
23 )
24 gpr.fit(attributes_normalized, labels)
25 predicted_labels = gpr.predict(attributes_normalized)
26
27 print('Rules:')
28 for rule in gpr.rules:
29     print(rule)
30
31 print('Accuracy:')
32 print(accuracy_score(labels, predicted_labels))

```

After executing the code shown in Listing 1, we obtained the following rules:

**IF y2 is High THEN 1 | Support : 0.6400**

**IF y1 is Low THEN 1 | Support : 0.6062**

**ELSE 0**

The IF part of each rule is the premise (antecedent) of the rule and refers to one attribute (feature) or more attributes. These attributes are connected using the logical AND operator. The

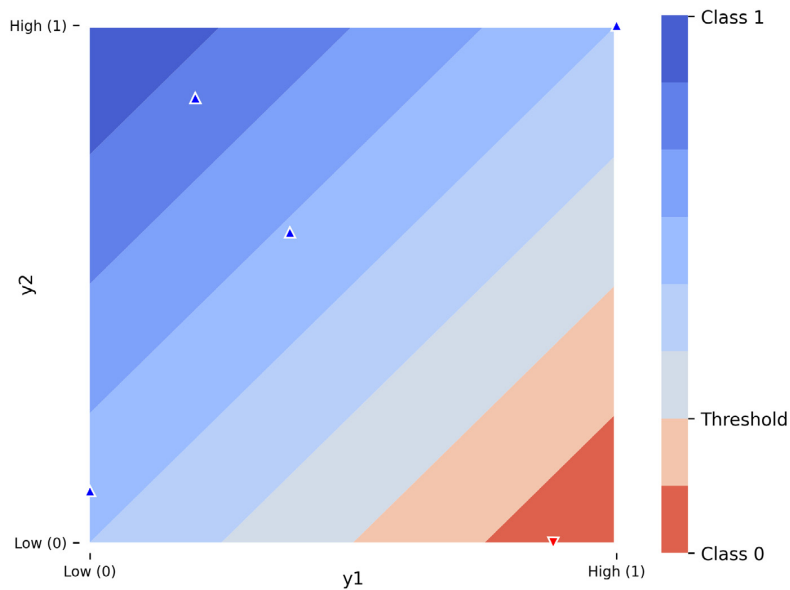


Fig. 2. The decision system boundaries visualization of the GPR algorithm fitted on 5 data records from Table 1, (Threshold = 0.5).

THEN part is the conclusion (consequent of the rule) and refers to a specific class. The resulting rules are the same as in [13].

The degree of confidence in the rule is determined for each rule, which is a number in the interval [0, 1], where 0 means no confidence, while 1 means the highest confidence. Thus, for rule “IF y2 is High THEN 1”, the support is determined according to columns “y2 is High” and “label” in Table 1 as:

$$(0.100 + 1.000 + 0.860 + 0.600)/4 = 0.6400.$$

Similarly, the support for the second rule “IF y1 is Low THEN 1”, is calculated according to columns “y1 is Low” and “label” in Table 1 as:

$$(1.000 + 0.000 + 0.800 + 0.625)/4 = 0.6062.$$

Fig. 2 illustrates the decision boundaries of the GPR classifier in the feature space (y1, y2) for the considered dataset from Table 1 and the obtained fuzzy rules. The numerical values of the variables y1 and y2 are from the unity interval [0, 1].

The abscissa in Fig. 2 corresponds to the degree to which y1 is High (or Low). For example, “y1 is Low” means that the numerical value of the first original feature, i.e., z1 in Table 1, is near minimal. The ordinate corresponds to the degree to which y2 is High (or Low), e.g., “y2 is High”, which means that the numerical value of the second original feature, i.e., z2 in Table 1, is near the maximal one. Of course, High is the logical complement of Low and vice versa.

Suppose y represents a variable from the interval [0,1]. The following linguistic interpretation applies to the antecedent part, i.e., the “If” part, of any rule (metarule):

$$(y \text{ is High}) \text{ AND } (y \text{ is High}) = (y \text{ is very High}),$$

$$(y \text{ is Low}) \text{ AND } (y \text{ is Low}) = (y \text{ is very Low}),$$

$$(y \text{ is Low}) \text{ AND } (y \text{ is High}) = (y \text{ is Medium}),$$

$$(y \text{ is High}) \text{ AND } (y \text{ is Low}) = (y \text{ is Medium}),$$

where the sign “=” means equivalency; and the logical operator AND corresponds to the algebraic multiplication operation (\*). Such interpretation follows the analytical theory of fuzzy systems [13,19].

### 3.2. Example 2

Listing 2 presents the usage of GPR for the Breast Cancer Wisconsin (BCW) dataset [24]. It contains 699 records labeled as class 1 for malignant breast cancer and 0 for benign. Data records describe nine features that differ significantly between benign and malignant cases.

The GPR algorithm generated the following outcome:

**IF Bare Nuclei is High THEN Malignant | Support : 0.7340**

**IF Uniformity of Cell Size is High THEN Malignant | Support : 0.6192**

**ELSE Benign**

The received rules are similar to the rules presented in [13]. Fig. 3 shows the decision boundaries of the GPR classifier in the feature space (Bare Nuclei, Uniformity of Cell Size) in the BCW dataset.

### 3.3. Example 3

Listing 3 demonstrates the usage of GPR on Haberman’s survival dataset. The dataset includes 306 cases from a study examining the survival of breast cancer patients who underwent surgery. The following attributes describe the data records: age of the patient, year of operation, and number of positive axillary nodes detected. The goal is to determine whether a patient survived five years or longer or if the patient died within five years.

The GPR algorithm generated the following outcome:

**IF Age is Medium THEN Survived | Support : 0.2108**

**IF Positive is High THEN Survived | Support : 0.1434**

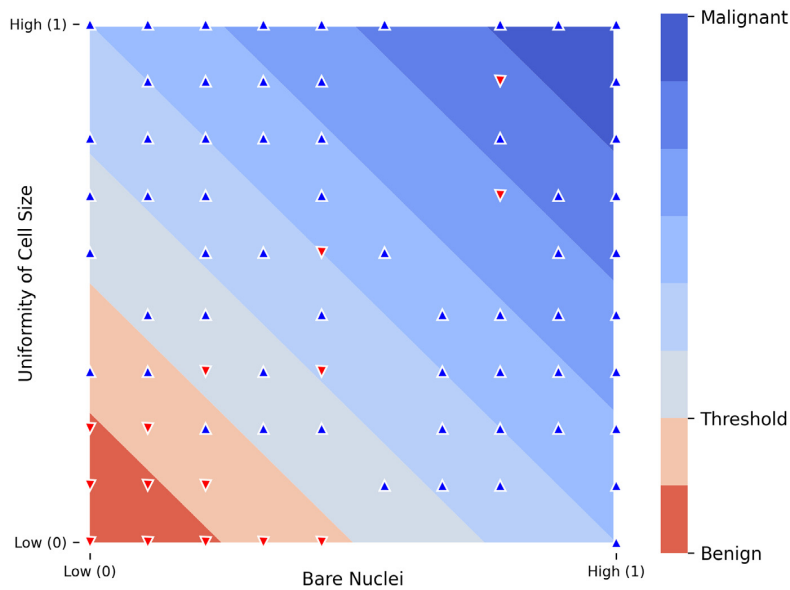


Fig. 3. The decision system boundaries visualization of the GPR algorithm fitted on 699 records in the BCW dataset, (Threshold=0.5).

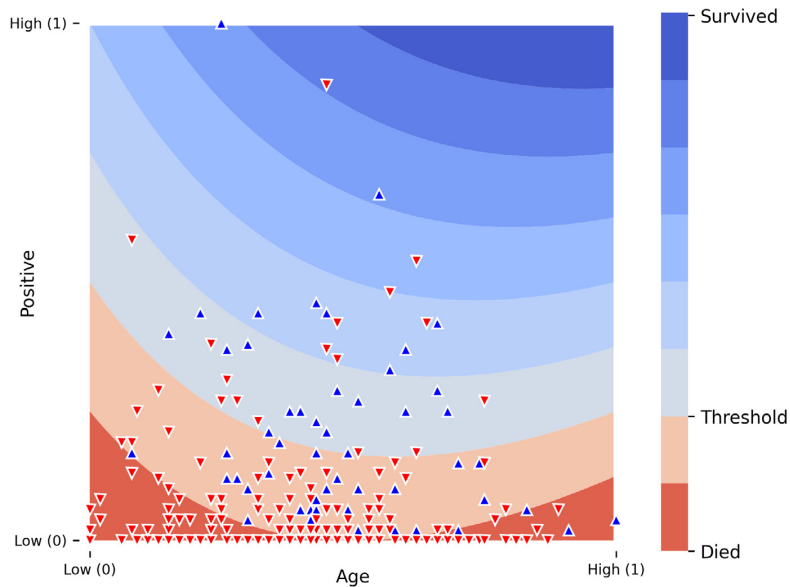


Fig. 4. The decision system boundaries visualization of the GPR algorithm fitted on 306 records in the Haberman dataset, (Threshold=0.5).

**IF Age is High AND Positive is High THEN Survived |**  
**Support : 0.0609**  
**ELSE Died**  
 which is equivalent to the following expression:  
**Age is High \* Age is Low + Positive is High**  
**+ Age is High \* Positive is High**

In this expression, the attribute age occurs three times, and the attribute positive occurs twice. By dividing the number of occurrences of a given terminal by the number of all terminals in the expression, we obtain its rank value. For the expression above, the importance values are as follows:

**Age: 0.6, Positive:0.4.**

Fig. 4 illustrates the decision boundaries of the GPR classifier in the feature space (Positive, Age) in the Haberman dataset.

**Listing 2:** An example usage of GPR on the breast cancer Wisconsin dataset.

```

1 import random
2 from pathlib import Path
3 import numpy as np
4 import pandas as pd
5 from sklearn.metrics import accuracy_score
6 from sklearn.preprocessing import MinMaxScaler
7 from gpr_algorithm import GPR
8
9 random.seed(0)
10 df = pd.read_csv(
11     Path(__file__).parent.joinpath('data').joinpath('
12         bcw.csv')
13 )
14 target_names = ['benign', 'malignant']
15 feature_names = [
16     'Clump Thickness', 'Uniformity of Cell Size',
17     'Uniformity of Cell Shape', 'Marginal Adhesion',
18     'Single Epithelial Cell Size', 'Bare Nuclei',
19     'Bland Chromatin', 'Normal Nucleoli', 'Mitoses'
20 ]
21 labels = df['Class'].values
22 labels[labels == 2] = 0
23 labels[labels == 4] = 1
24 attributes = df[feature_names].values
25 attributes_normalized = MinMaxScaler().fit_transform(
26     attributes)
27
28 gpr = GPR(
29     target_names=target_names,
30     feature_names=feature_names,
31     max_n_of_rules=3,
32     max_n_of_and=3,
33     n_generations=20,
34     n_populations=20,
35     verbose=False
36 )
37
38 gpr.fit(attributes_normalized, labels)
39 pred_labels = gpr.predict(attributes_normalized)

```

#### 4. Impact

The paper [13], shows that the data-driven GPR algorithm is among the best classifiers when the quality criterion is the area under the ROC curve or the classification accuracy. It ranks high for its direct competitors in the form of classifiers that provide interpretable models. This classifier is an extension of the P1-TS rule-based system [18–20]; it corresponds to the well-known and widely used Takagi–Sugeno–Kang fuzzy system [25]. Thus, we think that our proposed implementation will interest researchers interested in fuzzy logic and its applications. Classifiers play an essential role in machine learning methods. The GPR algorithm provides very simple and highly interpretable fuzzy “if-then” metarules. Thus, it is compatible with the currently intensively developed explainable artificial intelligence. This fact strengthens the argument that a Python implementation will enjoy popularity in the machine-learning community. Unlike the previous implementation, our software tool requires minimal user intervention, i.e., it only requires normalizing the data to the interval [0,1] and/or encoding the labels for the categorical data with numbers from the set {0,1}.

#### 5. Conclusions

We have presented GPR, a Python-based implementation of an extremely simple classifier, which combines the features of GEP and the algebraic representation of the “if-then” fuzzy rules theory of the Takagi–Sugeno fuzzy inference system. We aim to

**Listing 3:** An example of the use of GPR on the Haberman survival dataset.

```

1 import random
2 from pathlib import Path
3 import numpy as np
4 import pandas as pd
5 from sklearn.metrics import accuracy_score
6 from sklearn.preprocessing import MinMaxScaler
7 from gpr_algorithm import GPR
8
9 random.seed(0)
10 df = pd.read_csv(
11     Path(__file__).parent.joinpath('data').joinpath('
12         haberman.csv')
13 )
14 target_names = ['Died', 'Survived']
15 feature_names = [
16     'Age', 'Year', 'Positive'
17 ]
18
19 labels = df['Survival'].astype("category").cat.codes.
20     values
21 attributes = df[feature_names].values
22 attributes_normalized = MinMaxScaler().fit_transform(
23     attributes)
24
25 gpr = GPR(
26     target_names=target_names,
27     feature_names=feature_names,
28     max_n_of_rules=5,
29     max_n_of_and=5,
30     n_generations=20,
31     n_populations=20,
32     verbose=False
33 )
34
35 gpr.fit(attributes_normalized, labels)
36 pred_labels = gpr.predict(attributes_normalized)

```

widen the accessibility of this algorithm through an open-source Python implementation. Our implementation of GPR shares a consistent interface with Scikit-learn. Example code snippets of GPR usage and output results are explained and detailed to demonstrate how it works and give an overview of its capabilities. The code is available on GitHub under the MIT license.

#### CRedit authorship contribution statement

**Anna Czmił:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Project administration. **Jacek Kluska:** Conceptualization, Methodology, Validation, Formal analysis, Resources, Writing – original draft, Writing – review & editing, Visualization, Supervision, Funding acquisition. **Sylwester Czmił:** Software, Resources, Data curation, Writing – original draft, Writing – review & editing.

#### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Data availability

We have used publicly available data.

#### Acknowledgment

The work was financed by funds for the maintenance and research potential development (UPB) of the Rzeszow University of Technology.

## References

- [1] Witten IH, Frank E, Hall MA. Data mining : Practical machine learning tools and techniques. Morgan Kaufmann; 2011, p. 629. <http://dx.doi.org/10.1016/C2009-0-19715-5>.
- [2] Ferreira C. Gene expression programming : Mathematical modeling by an artificial intelligence. Berlin: Springer-Verlag; 2006, <http://dx.doi.org/10.1007/3-540-32849-1>.
- [3] Rudziński F. A multi-objective genetic optimization of interpretability-oriented fuzzy rule-based classifiers. Appl Soft Comput 2016;38:118–33. <http://dx.doi.org/10.1016/j.asoc.2015.09.038>.
- [4] Gacto M, Alcalá R, Herrera F. Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures. Inform Sci 2011;181(20):4340–60. <http://dx.doi.org/10.1016/j.ins.2011.02.021>, Special Issue on Interpretable Fuzzy Systems.
- [5] Chi Z, Yan H, Pham T. Fuzzy algorithms: With applications to image processing and pattern recognition, Vol. 10. World Scientific; 1996.
- [6] del Jesus M, Hoffmann F, Navascues L, Sanchez L. Induction of fuzzy-rule-based classifiers with evolutionary boosting algorithms. IEEE Trans Fuzzy Syst 2004;12(3):296–308. <http://dx.doi.org/10.1109/TFUZZ.2004.825972>.
- [7] Ishibuchi H, Yamamoto T. Rule weight specification in fuzzy rule-based classification systems. IEEE Trans Fuzzy Syst 2005;13(4):428–35. <http://dx.doi.org/10.1109/TFUZZ.2004.841738>.
- [8] Hühn J, Hüllermeier E. FURIA: An algorithm for unordered fuzzy rule induction. Data Min Knowl Discov 2009;19(3):293–319. <http://dx.doi.org/10.1007/s10618-009-0131-8>.
- [9] Alcalá-Fdez J, Alcalá R, Herrera F. A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning. IEEE Trans Fuzzy Syst 2011;19(5):857–72. <http://dx.doi.org/10.1109/TFUZZ.2011.2147794>.
- [10] Cózar J, Fernández A, Herrera F, Gámez JA. A metahierarchical rule decision system to design robust fuzzy classifiers based on data complexity. IEEE Trans Fuzzy Syst 2019;27(4):701–15. <http://dx.doi.org/10.1109/TFUZZ.2018.2866967>.
- [11] Ishibuchi H, Yamamoto T, Nakashima T. Hybridization of fuzzy GBML approaches for pattern classification problems. IEEE Trans Syst Man Cybern B 2005;35(2):359–65. <http://dx.doi.org/10.1109/TSMCB.2004.842257>.
- [12] Nakashima T, Schaefer G, Yokota Y, Ishibuchi H. A weighted fuzzy classifier and its application to image processing tasks. Fuzzy Sets and Systems 2007;158(3):284–94. <http://dx.doi.org/10.1016/j.fss.2006.10.011>, Image Processing.
- [13] Kluska J, Madera M. Extremely simple classifier based on fuzzy logic and gene expression programming. Inform Sci 2021;571:560–79. <http://dx.doi.org/10.1016/j.ins.2021.05.041>.
- [14] Kluska J, Kusy M, Obrzut B. The classifier for prediction of peri-operative complications in cervical cancer treatment. In: Rutkowski L, et al., editors. ICAISC 2014. Cham: Springer International Publishing; 2014, p. 143–54. [http://dx.doi.org/10.1007/978-3-319-19324-3\\_18](http://dx.doi.org/10.1007/978-3-319-19324-3_18).
- [15] Chen T, Shang C, Su P, Keravnou-Papailiou E, Zhao Y, Antoniou G, et al. A decision tree-initialised neuro-fuzzy approach for clinical decision support. Artif Intell Med 2021;111:1–13. <http://dx.doi.org/10.1016/j.artmed.2020.101986>.
- [16] Czmil A, Czmil S, Mazur D. A method to detect type 1 diabetes based on physical activity measurements using a mobile device. Appl Sci 2019;9(12). <http://dx.doi.org/10.3390/app9122555>, URL <https://www.mdpi.com/2076-3417/9/12/2555>.
- [17] Zadeh LA. A fuzzy-set-theoretic interpretation of linguistic hedges. J Cybern 1972;2(3):4–34. <http://dx.doi.org/10.1080/01969727208542910>.
- [18] Kluska J. Selected applications of P1-TS fuzzy rule-based systems. In: Rutkowski L, Korytkowski M, Scherer R, Tadeusiewicz R, Zadeh LA, Zurada JM, editors. Artificial intelligence and soft computing. Cham: Springer International Publishing; 2015, p. 195–206.
- [19] Kluska J. Analytical methods in fuzzy modeling and control. Studies in fuzziness and soft computing, Berlin, Heidelberg: Springer; 2009, <http://dx.doi.org/10.1007/978-3-540-89927-3>.
- [20] Kluska J. Transformation lemma on analytical modeling via Takagi-Sugeno fuzzy system and its applications. In: Rutkowski L, Tadeusiewicz R, Zadeh LA, Zurada JM, editors. Artificial Intelligence and Soft Computing – ICAISC 2006. Berlin, Heidelberg: Springer Berlin Heidelberg; 2006, p. 230–9.
- [21] Fortin F-A, De Rainville F-M, Gardner M-A, Parizeau M, Gagné C. DEAP: Evolutionary algorithms made easy. J Mach Learn Res 2012;13:2171–5.
- [22] Gao S. Geppy: A Python framework for gene expression programming. 2020, <http://dx.doi.org/10.5281/zenodo.3946297>.
- [23] Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, et al. Array programming with NumPy. Nature 2020;585(7825):357–62. <http://dx.doi.org/10.1038/s41586-020-2649-2>.
- [24] Wolberg WH, Mangasarian OL. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. Proc Natl Acad Sci 1990;87(23):9193–6. <http://dx.doi.org/10.1073/pnas.87.23.9193>.
- [25] Takagi T, Sugeno M. Fuzzy identification of systems and its applications to modeling and control. IEEE Trans Syst Man Cybern 1985;SMC-15(1):116–32. <http://dx.doi.org/10.1109/TSMC.1985.6313399>.



Article

# Comparative Study of Fuzzy Rule-Based Classifiers for Medical Applications

Anna Czmil 

The Faculty of Electrical and Computer Engineering, Rzeszow University of Technology, Powstancow Warszawy 12, 35-959 Rzeszow, Poland; czmilanna@gmail.com

**Abstract:** The use of machine learning in medical decision support systems can improve diagnostic accuracy and objectivity for clinical experts. In this study, we conducted a comparison of 16 different fuzzy rule-based algorithms applied to 12 medical datasets and real-world data. The results of this comparison showed that the best performing algorithms in terms of average results of Matthews correlation coefficient (MCC), area under the curve (AUC), and accuracy (ACC) was a classifier based on fuzzy logic and gene expression programming (GPR), repeated incremental pruning to produce error reduction (Ripper), and ordered incremental genetic algorithm (OIGA), respectively. We also analyzed the number and size of the rules generated by each algorithm and provided examples to objectively evaluate the utility of each algorithm in clinical decision support. The shortest and most interpretable rules were generated by 1R, GPR, and C45Rules-C. Our research suggests that GPR is capable of generating concise and interpretable rules while maintaining good classification performance, and it may be a valuable algorithm for generating rules from medical data.

**Keywords:** fuzzy rule-based system; interpretability; clinical decision support; medical diagnostic systems



**Citation:** Czmil, A. Comparative Study of Fuzzy Rule-Based Classifiers for Medical Applications. *Sensors* **2023**, *23*, 992. <https://doi.org/10.3390/s23020992>

Academic Editors: Aleksandra Kawala-Sterniuk, Grzegorz Marcin Wójcik and Waldemar Bauer

Received: 22 November 2022

Revised: 20 December 2022

Accepted: 13 January 2023

Published: 15 January 2023



**Copyright:** © 2023 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Accurate diagnosis of patients with various illnesses and diseases is a challenging area of medical research. The key is predicting an outbreak of a disease, preventing the progression of chronic disease and saving lives if patients receive medical treatment immediately after diagnosis [1]. However, even the most experienced physician can become confused when a disease has several symptoms similar to another condition. A patient may also have a set of symptoms that can indicate various diseases, and these symptoms may not be easily quantifiable. When these symptoms occur, physicians at different professional and clinical levels can differ in their diagnosis, potentially resulting in a misdiagnosis. Moreover, patients are often uncertain of their symptoms, making the diagnosis more difficult. Therefore, computers have become crucial for medical diagnosis and prognosis in providing consistent results, especially with the growing amount of medical information [2]. However, machines cannot fully replace expert knowledge. Combining human expertise and computational models for advanced data analysis helps narrow the gap between acquiring and understanding data, which is vital for medical research. Experts need tools to transform raw and complex data into easily interpretable information, but the output of the algorithm alone is not sufficient for making an accurate diagnosis; expert knowledge is also required [3]. As diagnostic decision-making becomes more complex, developing highly effective and reliable medical decision support systems (MDSS) to support the complex and evolving diagnostic process is challenging [1].

Although data analytics for healthcare is gaining recognition rapidly, there are still limitations associated with machine learning algorithms that are black boxes. These algorithms contain a complex mathematical function, e.g., support vector machines (SVMs), or require an understanding of the distance function and the representation space, e.g., k-nearest



neighbors (KNN), which are very challenging to explain and to be understood by experts in practical applications. However, the application of black-box algorithms in medicine has raised concerns in the academic community due to their opacity and lack of trustworthiness [4]. To summarize the performance of a model, it is necessary to report several metrics, since no single metric captures all the desired properties. Nevertheless, tools such as CACP simplify this task by allowing the assessment of classification efficiency, reproducibility, and statistical reliability while maintaining the objectivity of model comparisons [5].

Classification quality is crucial, but it is also essential to understand how a record is classified. MDSS rely on knowledge management to obtain clinical advice based on multiple factors in patient-related data. In these applications, models based on patterns, rules, or decision trees are more useful and easier for experts to comprehend in practical applications. In particular, rule-based systems (RBS) represent knowledge in the form of a set of rules that suggest what to do in various situations. They consist of a set of “if-then” rules, a set of facts, and interpreters that control the application of the rules. The idea of an expert system is to use the experience and facts in a knowledge base and encode it into a set of rules. If the expert system has access to the same data, it will behave similarly to the expert. RBS are straightforward models that can be adapted and applied to numerous problems [6]. Rule-based systems are known as white-box models because they provide a model closer to human language, making them easy for experts to understand [7]. The interpretability of a classification model is particularly important for MDSS. When designing classifiers, it is crucial to reach a compromise between interpretability and accuracy [8]. Accuracy is a well-known method for validating machine learning models in classification problems due to its popularity and relative simplicity. However, there is no widely accepted measure of the interpretability of machine learning models [9]. As it depends on several factors, mainly the structure of the model, the shape of the membership functions, the number of rules, attributes, and linguistic terms, it can be difficult to measure [8].

After introducing fuzzy rule-based systems (FRBS), which are models based on fuzzy sets proposed by Zadeh [10], many of their applications emerged in different areas such as artificial intelligence, robotics, decision-making, expert systems, power engineering, and medicine [11–14]. A fuzzy logic approach is an effective way to represent and understand data containing both patient information and clinical reasoning used by physicians to conclude patients' health that is inherently uncertain and vague in medical problems. It has proven to be a powerful tool in developing decision support systems, such as rule-based medical decision support systems [3]. Many algorithms have been proposed for designing FRBS, including one rule (1R), C4.5 and its extensions, the exemplar-aided constructor of hyperrectangle (EACH), and repeated incremental pruning to produce error reduction (Ripper). 1R is a simple algorithm that uses a single rule to make predictions [15]. C4.5 and its extensions are decision tree learning algorithms that use fuzzy logic to make decisions at each node of the tree [16]. EACH is a clustering algorithm that uses fuzzy logic to group data into clusters [17], and Ripper is an algorithm that uses fuzzy logic to prune, or remove, unnecessary rules from a fuzzy rule-based system [18]. Genetic algorithms have been successfully applied to the generation of fuzzy rules and the adjustment of the membership functions of fuzzy sets [19]. Examples of these algorithms include hybrid decision tree-genetic algorithm (DT\_GA), which combines a decision tree learning algorithm with a genetic algorithm [20], and the oblique decision tree with evolutionary learning (DT\_Oblique), which uses evolutionary learning to improve the performance of an oblique decision tree [21]. Other examples include structural learning algorithm in a vague environment (SLAVEv0) and its extensions, which use genetic algorithms to learn the structure of a fuzzy rule-based system [22], the classifier based on fuzzy logic and gene expression programming (GPR) that combines fuzzy logic with gene expression programming to generate fuzzy rules for classification tasks [8], and hierarchical decision rules (Hider), which use genetic algorithms to generate fuzzy rules for classification tasks [23]. Organizational co-evolutionary algorithm for classification (OCEC) is another example of a genetic algorithm applied to fuzzy rule-based systems. This algorithm uses co-evolutionary



learning, in which multiple populations of solutions are evolved simultaneously, to improve the performance of a fuzzy classifier [24]. Ordered incremental genetic algorithm (OIGA) [25] and Pittsburgh genetic interval rule learning algorithm (PGIRLA) [26] are both examples of genetic algorithms that are specifically designed for learning fuzzy rules. These algorithms use genetic operations to generate and refine a set of fuzzy rules that can be used to make decisions.

## 2. Related Work

Fuzzy logic is used extensively for medical applications by researchers for diagnosis and classification. For example, Aamir et al. used a fuzzy rule-based algorithm to predict the severity of diabetes in patients [27]. Adeli and Neshat found that a fuzzy rule-based algorithm was effective in diagnosing heart disease from electrocardiogram (ECG) data [28]. Improtta et al. utilized a fuzzy rule-based algorithm for the evaluation of renal function in posttransplant patients [29]. Rotshtein proposed an approach for building a fuzzy expert system for the differential diagnosis of ischemia heart disease [30]. Mohammadpour et al. determined the accuracy of fuzzy rule-based classification that could non-invasively predict CAD based on the myocardial perfusion scan test and clinical-epidemiological variables [31]. Al-Dmour et al. presented the usage of fuzzy logic techniques in a warning system to categorize patients' status or medical conditions [32]. RBS and FRBS have also been used to develop many MDSS in recent decades [31,33–46]. These systems represent the symptoms of MDSS patients and are based on an inference algorithm to process the information using linguistic terms. Domain knowledge is embedded as rules in the knowledge base.

Many studies demonstrate the potential of using different fuzzy rule-based algorithms in medical applications while simultaneously comparing different fuzzy algorithms. Steimann investigated the impact of fuzzy set theory on medical artificial intelligence and pointed out its most appreciated features [47]. Gupta et al. reviewed various fuzzy models that are being used in healthcare systems for making decisions. Mousavi et al. proposed an intelligent classification algorithm using a fuzzy rule-based approach to classify medical datasets and compared it with selected fuzzy rule-based algorithms [48]. Kluska and Madera proposed a new design for a very simple data-driven binary classifier and conducted an empirical study of its performance using other state-of-the-art algorithms and datasets from multiple disciplines, including medicine [8]. There are also many reviews in the literature on various fuzzy rule-based systems [49–52]. These works highlight important contributions, current trends, and challenges in the field.

Among the different reviews in the literature, choosing the type of fuzzy rule-based algorithm for particular medical applications remains a challenging task. The comparison of available algorithms is not straightforward, as researchers use various datasets and criteria for their evaluations. Another challenge is selecting an appropriate metric to evaluate the calculated results. Available research has not yet comprehensively investigated the validity of the outcomes of fuzzy rule-based algorithms using a wide range of available algorithms and metrics. Therefore, this study has two main objectives. First, we compare all commonly used, state-of-the-art algorithms and assess their performance. The comparison is made against the results of all selected algorithms compared in every dataset, calculated using 10-fold cross-validation. Our findings demonstrate a ranking of the algorithms in terms of the most popular performance metrics. Second, we analyze fuzzy rule-based classifiers in terms of rules' size metrics and provide examples of rules generated by every algorithm to objectively determine which of these algorithms is worth using when applied to issues in clinical decision support. The use of some of those algorithms in the field of medicine is novel.

The remainder of the paper is structured as follows. Section 3 provides the details of the experimental datasets. Section 4 describes the applied fuzzy rule-based classification algorithms and their settings. Section 5 presents the classification assessment methods.

Then in Section 6, the experimental results of the comparison are presented. Finally, Section 7 contains a discussion, observations, and conclusions.

### 3. Experimental Datasets

This article focuses on the medical applications of fuzzy rule-based classifiers, so only medical data are considered. Datasets were downloaded from the KEEL—dataset repository [53], and actual medical data were collected during other scientific research, as detailed below. We used standard classification datasets without missing values. Each dataset defines a supervised classification problem, and each example has some nominal and numerical attributes and a nominal output attribute. The datasets have different levels of class imbalance. Table 1 presents a summary of the datasets, including the number of records, attributes, classes, and class imbalance.

**Table 1.** Summary of datasets used in experiments.

	Dataset	Records	Attributes	Classes	Class Imbalance	Source
1	Appendicitis	106	7	2	0.2471	KEEL
2	Breast cancer	277	9	2	0.4133	KEEL
3	Haberman	306	3	2	0.3600	KEEL
4	Heart	270	13	2	0.8000	KEEL
5	Hepatitis	80	19	2	0.1940	KEEL
6	Mammographic	830	5	2	0.9438	KEEL
7	Saheart	462	9	2	0.5298	KEEL
8	Spectfheart	267	44	2	0.2594	KEEL
9	WDBC	569	30	2	0.5938	KEEL
10	Wisconsin	683	9	2	0.5383	KEEL
11	Complications	107	8	2	0.8136	Real
12	Diabetes	230	9	2	1.0000	Real

#### 3.1. Appendicitis

The dataset includes 7 medical measures taken from 106 patients, along with a class label that indicates whether the patient has appendicitis (label 1) or not (label 0) according to the research by S. M. Weiss and C. A. Kulikowski [54].

#### 3.2. Breast Cancer

The dataset of 277 instances with no missing values is characterized by 9 attributes provided by the Institute of Ljubljana Oncology. These attributes include both linear and nominal values, e.g., age, tumor nodes, and tumor size.

#### 3.3. Haberman

The dataset contains 306 records described by 3 attributes of a study on the survival of patients who had undergone breast cancer surgery at Billings Hospital at the University of Chicago between 1958 and 1970. The task is to predict whether the patient survived for five years or more after surgery (positive) or died within five years (negative).

#### 3.4. Heart

The heart disease database includes 270 instances with 13 attributes, each labeled with a class label indicating the absence (1) or presence (2) of heart disease. This dataset can be used to analyze various factors and characteristics that may be associated with heart disease.

#### 3.5. Hepatitis

The dataset contains information on 80 patients affected by hepatitis, including a mixture of 19 integer and real-valued attributes. The task is to predict whether these patients will die (1) or survive (2).

### 3.6. Mammographic

The dataset includes 5 attributes related to the severity (benign or malignant) of a mammographic mass lesion in 830 patients, based on the characteristics of BI-RADS and the patient's age.

### 3.7. Saheart

The dataset contains information on 462 men living in a high-risk region for coronary heart disease in the Western Cape, South Africa. It is characterized by 9 attributes. The class label indicates whether the person has coronary heart disease: negative (0) or positive (1).

### 3.8. Spectfheart

The dataset contains information on the diagnosis of single proton emission computed tomography (SPECT) images of the heart in 267 patients. Each record is described by 44 attributes, and each patient is classified into one of two categories: normal (0) or abnormal (1).

### 3.9. Wisconsin Diagnosis Breast Cancer (WDBC)

The dataset contains 569 records with 30 features computed from a digitized image of a breast mass. These attributes describe the characteristics of the cell nuclei present in the image. The task is to predict whether the tumor found is benign or malignant.

### 3.10. Wisconsin Breast Cancer Original (Wisconsin)

The dataset contains 9 attributes with 683 cases from a study of patients who had undergone breast cancer surgery. The task is to predict whether the detected tumor is benign (2) or malignant (4).

### 3.11. Complications

The dataset contains 107 cases of perioperative complications of radical hysterectomy in patients with cervical cancer described by 8 attributes. The task is to determine the presence or absence of perioperative complications [13].

### 3.12. Diabetes

Data was collected from 230 schoolchildren between the ages of 6 and 18 under the care of a pediatric diabetes clinic. It contains 9 parameters, including weekly physical activity parameters. The task is to determine the presence or absence of type 1 diabetes [55].

## 4. Fuzzy Rule-Based Classification Algorithms

This section contains descriptions of the classification algorithms used in these experiments. The algorithms implementations, except for GPR, come from KEEL Included Algorithms [53] and belong to the Rule Learning for Classification family. We used a custom implementation of GPR [56], and set the parameters to default values.

### 4.1. One Rule (1R-C)

1R is an algorithm that ranks attributes according to their error rate, with the attribute with the lowest error rate chosen for the decision tree. The range of values for the selected attribute is then divided into several disjoint intervals, with the number of intervals determined by the value of the SMALL parameter. Finally, the algorithm uses these intervals to create a one-level decision tree, which is a tree with a single decision node that classifies objects based on the chosen attribute [15]. The SMALL parameter was set to 6.

### 4.2. C4.5 (C4.5-C)

C4.5-C is probably the most widely used machine learning algorithm for generating a decision tree [16]. It is an extension of Quinlan's earlier ID3 algorithm [57]. The pruned parameter that determines whether the algorithm will prune the decision tree was set

to TRUE. The confidence parameter determines the minimum confidence required for a rule to be considered significant, and in this case it was set to 0.25. The instances per leaf parameter determines the minimum number of instances that must be present at a leaf node and it was set to 2.

#### 4.3. C4.5Rules (C45Rules-C)

C45Rules-C is an algorithm that reads the decision tree or trees produced by C4.5 and generates a set of rules for each tree and all trees together [57,58]. The confidence factor, item sets per leaf, and threshold parameters can be adjusted to fine-tune the generated rules for optimal performance. In the current implementation, the confidence factor was set to 0.25, the item sets per leaf parameter was set to 2, and the threshold was set to 10.

#### 4.4. C4.5Rules Simulated Annealing Version (C45RulesSA-C)

C45RulesSA-C is a version of the C45Rules-C algorithm with a general-purpose local search method called Simulated Annealing that generates an approximate solution within a range close to the current solution and accepts the approximate solution if the objective function improves [57,58]. The user-defined parameters such as confidence, item sets per leaf, and threshold are used to fine-tune the generated rules, while the max coldings, max trials, mu, phi, and alpha parameters are used to control the behavior of the Simulated Annealing method. In the current implementation, these parameters were set to 0.25, 2, 10, 10, 0.5, 0.5, and 0.5 respectively.

#### 4.5. Hybrid Decision Tree-Genetic Algorithm (DT\_GA-C)

DT\_GA-C is a hybrid decision tree/genetic algorithm method that allows discovering knowledge from data expressed as easy-to-interpret high-level classification rules [20]. A genetic algorithm aims to generate rules covering examples belonging to small disjuncts, whereas a conventional decision tree algorithm aims to produce rules covering examples of large disjuncts. The user-defined parameters of DT\_GA-C, such as confidence was set to 0.25, the instances per leaf parameter was set to 2, and the genetic algorithm approach parameter was set to GA-LARGE-SN. The threshold S to consider a small disjunct parameter was set to 10, the number of total generations for the GA parameter was set to 50, and the number of chromosomes in the population parameter was set to 200. Crossover probability was set to 0.8, and the mutation probability parameter was set to 0.01.

#### 4.6. Oblique Decision Tree with Evolutionary Learning (DT\_Oblique-C)

DT\_Oblique-C uses evolutionary algorithms to optimize split criteria during constructing oblique trees [21]. This allows the algorithm to quickly and efficiently find high-quality split criteria that accurately classify the data. In the current implementation, the number of total generations for the genetic algorithm was set to 25, indicating that the algorithm will run for up to 25 generations before stopping.

#### 4.7. Exemplar-Aided Constructor of Hyperrectangles (EACH-C)

EACH-C implements the nested generalized exemplar (NGE) theory. It makes predictions and classifications based on examples that it has seen in the past. The algorithm compares new examples with those it has seen before and finds the closest example in memory. Distance measure aims to determine what is closest [17]. The feature adjustment rate was set to 0.2, and the use second chance parameter was set to TRUE.

#### 4.8. Classifier Based on Fuzzy Logic and Gene Expression Programming (GPR)

GPR is an extremely simple classifier that consists of highly interpretable fuzzy metarules [8]. It uses only two fuzzy sets with linear and complementary membership functions for every continuous feature. The number of populations was set to 500, the number of generations was set to 10, threshold was set to 0.5, and the probability of triggering an operation on the chromosome was set to 0.1.

#### 4.9. Hierarchical Decision Rules (Hider-C)

Hider-C uses an approach based on evolutionary algorithms to learn rules in continuous and discrete domains. The algorithm produces a hierarchical set of rules. It uses real and binary coding for individuals in the population [23]. The population size, number of generations, mutation probability and cross percent parameters are used to control the behavior of the genetic algorithm component. In this case, these parameters are set to 0.25, 100, 100, 0.5, and 80 respectively. The extreme mutation probability, prune examples factor, penalty factor, and error coefficient parameters are used to fine-tune the generated rules and control the behavior of the decision tree component of DT\_Oblique-C. In this case, the extreme mutation probability is set to 0.05, the prune examples factor is set to 0.05, the penalty factor is set to 1, and the error coefficient is set to 0.

#### 4.10. New Structural Learning Algorithm in a Vague Environment (NSLV-C)

NSLV-C is an extension of the iterative scheme of SLAVE that aims to improve the efficiency of the learning process by obtaining complete rules in each iteration and reducing the learning time [59]. It modifies the iterative scheme and the genetic algorithm to remove the bias of the class order and find the best rule in each iteration without fixing the class. We set the study parameters in this study as follows: the population size was set to 100, the maximum number of iterations allowed without change was set to 500, the binary mutation probability was set to 0.01, the integer mutation probability was set to 0.01, the real mutation probability was set to 1.0, and the crossover probability was set to 1.0.

#### 4.11. Organizational Co-Evolutionary Algorithm for Classification (OCEC-C)

OCEC-C causes the evolution of sets of examples and, finally, extracts rules from these sets at the end of the evolutionary process [24]. Due to the differences between the individuals in traditional evolutionary algorithms and organizations formed from these sets of examples, three evolutionary operators and a selection mechanism have been developed for realizing the evolutionary operations performed on organizations. It prevents evolutionary processes from producing meaningless rules. The number of total generations was set to 500, and the number of migrating/exchanging members was set to 1.0.

#### 4.12. Ordered Incremental Genetic Algorithm (OIGA-C)

OIGA-C address incremental training of input attributes for classifiers [25]. OIGA learns input attributes one after another, and the resulting classification rule sets are also incrementally evolved to accommodate the new attributes. The attributes are arranged in different orders when their discriminating abilities are evaluated. The parameters were set as follows: the mutation probability was set to 0.01, the crossover rate was set to 1.0, the population size was set to 200, the number of rules was set to 30, the stagnation limit was set to 30, the generation limit was set to 200, the survivors percent was set to 0.5, and the attribute order was set to descendent.

#### 4.13. Pittsburgh Genetic Interval Rule Learning Algorithm (PGIRLA-C)

PGIRLA-C uses genetic algorithms with real genes to evolve the classification rule sets. The rule sets are evolved by genetic algorithms using the Pittsburgh approach [26]. We set the number of generations to 5000, the population size to 61, the crossover probability to 0.7, the mutation probability to 0.5, and the number of rules to 20.

#### 4.14. Repeated Incremental Pruning to Produce Error Reduction (Ripper-C)

Ripper-C is a rule-based classification algorithm proposed by Cohen that derives a set of rules from the training set that match or exceed the performance of decision trees [18]. The three stages of RIPPER-C are growing, pruning, and optimizing. The grow\_pct parameter was set to 0.66, and k to 2.

#### 4.15. Structural Learning Algorithm in a Vague Environment v0 (SLAVEv0-C)

SLAVEv0-C is a classifier based on fuzzy rules that is generated evolutionarily. Fuzzy rules are evolved for each two-class problem using a Michigan iterative learning approach and integrated using the fuzzy round-robin class binarization scheme [22]. The parameters were set as follows: the population size was set to 20, the number of iterations allowed without change was set to 500, the mutation probability was set to 0.5, the crossover probability was set to 0.1, and lambda was set to 0.8.

#### 4.16. Structural Learning Algorithm in a Vague Environment 2 (SLAVE2-C)

SLAVE2-C is a modification of the original SLAVE learning algorithm, including new genetic operators to reduce learning time, improve understanding of the rules obtained, and a new way to penalize the rules in the iterative approach that allows the system's behavior to improve [60]. The following parameters were set: the population size was set to 20, the number of iterations allowed without change was set to 500, the binary mutation probability was set to 0.5, the binary crossover probability was set to 0.1, the real mutation probability was set to 1.0, the real crossover probability was set to 0.2, and lambda was set to 0.8.

### 5. Performance Metrics

The selection of metrics that measure the performance of algorithms is an essential step in machine learning approaches. Each metric has specific characteristics and measures properties that may be different from the predicted results. The metrics used to evaluate the performance of the proposed work are listed below.

Accuracy (ACC) is calculated by dividing the number of correctly classified samples by the total number of samples in the evaluation dataset. If the model's predictions for a sample exactly match the true labels for that sample, the subset accuracy is 1.0; otherwise, it is 0.0. The fraction of correct predictions over  $n_{\text{samples}}$  is calculated using the *accuracy\_score* function from the *sklearn.metrics* module defined as follows:

$$\text{ACC}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1(\hat{y}_i = y_i) \quad (1)$$

where  $\hat{y}_i$  is the predicted value of the  $i$ -th sample,  $y_i$  is the true value for that sample, and  $1(x)$  is the indicator function.

Precision (Pre) is calculated as the ratio of correctly classified samples to all samples assigned to a particular class. Pre is the ability of the classifier to not label a negative sample as positive. It is bounded between 0 and 1, where 1 is the best possible value and 0 is the worst possible value. The metrics for each label, and averages weighted by support, are calculated. It is defined by:

$$\text{Pre} = \frac{1}{\sum_{l \in L} |y_l|} \sum_{l \in L} |y_l| P(y_l, \hat{y}_l) \quad (2)$$

where  $y$  the set of true (sample, label) pairs,  $\hat{y}$  the set of predicted (sample, label) pairs,  $L$  the set of labels,  $y_l$  the subset of  $y$  with label  $l$ ,  $\hat{y}_s$  and  $\hat{y}_l$  are subsets of  $\hat{y}$ ,  $P(A, B) := \frac{|A \cap B|}{|B|}$  for some sets  $A$  and  $B$ . It is calculated using the *precision\_score* method from the *sklearn.metrics* module.

Sensitivity (Sen) (also known as the Recall) is calculated as the ratio between correctly classified positive samples and all samples assigned to the positive class. Sen is the ability of the classifier to correctly classify all positive samples as positive. It is defined as follows:

$$\text{Sen} = \frac{1}{\sum_{l \in L} |y_l|} \sum_{l \in L} |y_l| R(y_l, \hat{y}_l) \quad (3)$$

where  $y$  the set of true (sample, label) pairs,  $\hat{y}$  the set of predicted (sample, label) pairs,  $L$  the set of labels,  $y_l$  the subset of  $y$  with label  $l$ ,  $\hat{y}_s$  and  $\hat{y}_l$  are subsets of  $\hat{y}$ ,  $R(A, B) := \frac{|A \cap B|}{|A|}$ . It is calculated using the *recall\_score* method from the *sklearn.metrics* module.

Other performance metrics are calculated using the well-known confusion matrix which consists of four entries: the true positives (TP), false negatives (FN), false positives (FP), and true negatives (TN) [61], as follows:

$$\mathbf{M} = \begin{pmatrix} \text{TP} & \text{FN} \\ \text{FP} & \text{TN} \end{pmatrix} \quad (4)$$

True Positives (TP) refer to the number of samples correctly classified as positive, e.g., the number of records that have breast cancer correctly predicted as having breast cancer. True Negatives (TN) refer to the samples correctly classified as negative, e.g., the number of records without breast cancer correctly predicted to be non-breast cancer. False Positives (FP) refer to the samples incorrectly classified as positive, e.g., the number of samples without breast cancer incorrectly predicted to have breast cancer. False Negatives (FN) refer to the samples incorrectly classified as negative, e.g., the number of records containing breast cancer is incorrectly predicted not to have breast cancer.

Specificity (Spe) is calculated as the ratio between correctly classified negative samples and all samples classified as negative. Spe is bounded to  $[0, 1]$ , where 1 represents perfect predictions of the negative class and 0 represents incorrect predictions of all samples in the negative class. It is defined by:

$$\text{Spe} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (5)$$

Area Under ROC Curve (AUC) measures the ability of a classifier to distinguish between classes and is used to summarize the ROC curves. The higher AUC, the better model's performance in distinguishing between the positive and negative classes. The ROC curve is plotted with Sen against the false positive rate (FPR, calculated as  $1 - \text{Spe}$ ). Sen is on the  $y$ -axis, and FPR is on the  $x$ -axis.

Matthews Correlation Coefficient (MCC) is a correlation coefficient between true and predicted classes. It reaches a high value only if the classifier achieves good results in all entries in the confusion matrix. MCC is bounded to  $[-1, 1]$ , where 1 represents a perfect prediction, 0 random guessing, and  $-1$  represents total disagreement between prediction and observation [62]. MCC has become popular research applied in machine learning due to its favorable properties in the case of imbalanced classes. It is defined as follows:

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \quad (6)$$

Weighted Metric (WM) is a single performance indicator for multiple metrics that was proposed in this study to make it easier to compare algorithms and select the optimal algorithm:

$$\text{WM} = \frac{30 \times \text{AUC} + 50 \times \text{Sen} + 5 \times (\text{ACC} + \text{Pre} + \text{Spe} + \text{MCC})}{100} \quad (7)$$

According to some studies [63], the AUC is one of the most significant measures of a classifier's performance, so that it was included with a weight of 0.3. The Sen term is also often used in health care and medical research to describe the confidence in results and utility of testing. Therefore, it was weighted with 0.5 when calculating WM, and other metrics were weighted with 0.05.

## 6. Experimental Results

A fuzzy rule-based algorithms' performance is evaluated in this section. Algorithms compared include: AdaBoost.NC-C, CART-C, C45-C, C45Rules-C, C45RulesSA-C, Chi-



RW-C, EACH-C, FH-GBML-C, FURIA-C, DT\_GA-C, MPLCS-C, DT\_Oblique-C, OIGA-C, OCEC-C, 1R-C, and GPR. Table 2 shows the average results of ACC, AUC, Pre, Sen, and Spe obtained on all datasets using 10-fold cross-validation. The results in Table 2 are sorted in descending order based on MCC, and the three best results for each metric are highlighted in bold.

**Table 2.** Results of a comparison of fuzzy rule-based algorithms.

No.	Algorithm	MCC	ACC	AUC	Spe	Pre	Sen	WM
1	GPR	<b>0.459 ± 0.342</b>	<b>0.807 ± 0.281</b>	0.720 ± 0.171	<b>0.792 ± 0.125</b>	0.772 ± 0.167	<b>0.792 ± 0.125</b>	<b>0.753 ± 0.145</b>
2	OIGA-C	<b>0.457 ± 0.337</b>	<b>0.860 ± 0.253</b>	0.714 ± 0.172	<b>0.793 ± 0.114</b>	<b>0.782 ± 0.152</b>	<b>0.793 ± 0.114</b>	<b>0.755 ± 0.138</b>
3	Ripper-C	<b>0.452 ± 0.319</b>	0.676 ± 0.243	<b>0.730 ± 0.162</b>	0.735 ± 0.158	<b>0.780 ± 0.139</b>	0.735 ± 0.158	0.718 ± 0.164
4	C45RulesSA-C	0.449 ± 0.343	0.752 ± 0.255	<b>0.727 ± 0.172</b>	0.769 ± 0.140	0.776 ± 0.147	0.769 ± 0.140	0.740 ± 0.157
5	OCEC-C	0.447 ± 0.323	0.753 ± 0.221	<b>0.726 ± 0.164</b>	0.753 ± 0.145	0.771 ± 0.145	0.753 ± 0.145	0.730 ± 0.156
6	NSLV-C	0.446 ± 0.338	0.791 ± 0.298	0.716 ± 0.171	<b>0.795 ± 0.122</b>	0.771 ± 0.148	<b>0.795 ± 0.122</b>	<b>0.752 ± 0.141</b>
7	C45Rules-C	0.446 ± 0.340	0.738 ± 0.273	0.724 ± 0.173	0.768 ± 0.142	<b>0.777 ± 0.141</b>	0.768 ± 0.142	0.737 ± 0.159
8	DT GA-C	0.442 ± 0.329	0.799 ± 0.267	0.712 ± 0.163	0.784 ± 0.116	0.775 ± 0.138	0.784 ± 0.116	0.746 ± 0.135
9	SLAVE2-C	0.438 ± 0.338	0.792 ± 0.296	0.712 ± 0.170	0.786 ± 0.123	0.769 ± 0.148	0.786 ± 0.123	0.746 ± 0.144
10	C45-C	0.438 ± 0.343	0.785 ± 0.264	0.710 ± 0.171	0.782 ± 0.128	0.772 ± 0.146	0.782 ± 0.128	0.743 ± 0.147
11	Hider-C	0.414 ± 0.336	0.797 ± 0.274	0.693 ± 0.167	0.767 ± 0.138	0.763 ± 0.144	0.767 ± 0.138	0.728 ± 0.150
12	DT Oblique-C	0.402 ± 0.346	0.741 ± 0.222	0.703 ± 0.173	0.745 ± 0.146	0.754 ± 0.149	0.745 ± 0.146	0.715 ± 0.160
13	SLAVEv0-C	0.394 ± 0.374	0.761 ± 0.315	0.691 ± 0.182	0.772 ± 0.137	0.749 ± 0.161	0.772 ± 0.137	0.727 ± 0.158
14	PGIRLA-C	0.327 ± 0.337	<b>0.819 ± 0.269</b>	0.655 ± 0.165	0.716 ± 0.193	0.668 ± 0.239	0.716 ± 0.193	0.681 ± 0.172
15	EACH-C	0.264 ± 0.340	0.621 ± 0.417	0.626 ± 0.165	0.662 ± 0.185	0.675 ± 0.238	0.662 ± 0.185	0.630 ± 0.180
16	1R-C	0.228 ± 0.331	0.652 ± 0.378	0.610 ± 0.160	0.703 ± 0.162	0.636 ± 0.211	0.703 ± 0.162	0.645 ± 0.160

GPR achieved the highest MCC of  $0.459 \pm 0.342$ , while 1R-C achieved the lowest MCC of  $0.228 \pm 0.331$ . In terms of ACC, OIGA-C ( $0.860 \pm 0.253$ ), PGIRLA-C ( $0.819 \pm 0.269$ ), and GPR ( $0.807 \pm 0.281$ ) obtained the best results. Ripper-C had the highest AUC score of  $0.730 \pm 0.162$ , followed by C45RulesSA-C and OCEC-C. The best Spe obtained NSLV-C ( $0.795 \pm 0.122$ ), OIGA-C ( $0.793 \pm 0.114$ ), and GPR ( $0.792 \pm 0.125$ ). OIGA-C achieved the highest Pre of  $0.782 \pm 0.152$ . According to Sen, NSLV-C achieved the highest results ( $0.795 \pm 0.122$ ), followed by OIGA-C ( $0.793 \pm 0.114$ ) and GPR ( $0.792 \pm 0.125$ ), and EACH-C achieved the worst performance ( $0.662 \pm 0.185$ ). OIGA-C, GPR, and NSLV-C had the highest WM scores of  $0.755 \pm 0.138$ ,  $0.753 \pm 0.145$ , and  $0.752 \pm 0.141$ , respectively. The algorithms with the lowest WM were EACH-C ( $0.630 \pm 0.180$ ), 1R-C ( $0.645 \pm 0.160$ ), and PGIRLA-C ( $0.681 \pm 0.172$ ).

The box plot in Figure 1 shows MCC of each algorithm in all datasets subjected to 10-fold cross-validation. The results are sorted in descending order by the median of the MCC. OIGA-C had the highest MCC among all the fuzzy rule-based algorithms tested. SLAVE2-C had the second-highest MCC, while GPR had the third-highest MCC. The plot also shows several outliers that decrease the average results of the algorithms.

The box plot in Figure 2 shows the AUC of each algorithm in all datasets subjected to 10-fold cross-validation. The results are sorted in descending order by the median of the AUC. The best results were obtained by the OCEC-C algorithm, followed by C45RulesSA-C, OIGA-C, GPR, and Ripper-C. The plot also shows several outliers that decrease the average results of the algorithms. The 1R-C and EACH-C algorithms are at the bottom of the list.

The box plot in Figure 3 shows the ACC of each algorithm in all datasets subjected to a 10-fold cross-validation. The results are sorted in descending order by the median of the ACC. GPR was found to be the most accurate among all the fuzzy rule-based algorithms tested. Therefore, GPR is a good choice for general use. SLAVE2-C was ranked second and NSLV-C was ranked third. The 1R-C and EACH-C algorithms again took the last two places, similar to their positions in rankings for MCC and AUC.



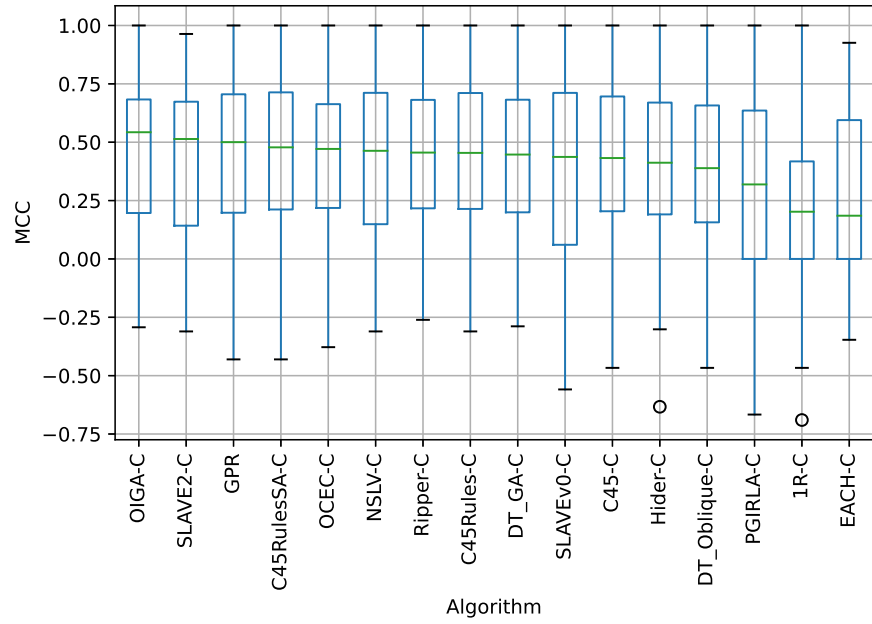


Figure 1. Distribution of the MCC values for each algorithm in all datasets.

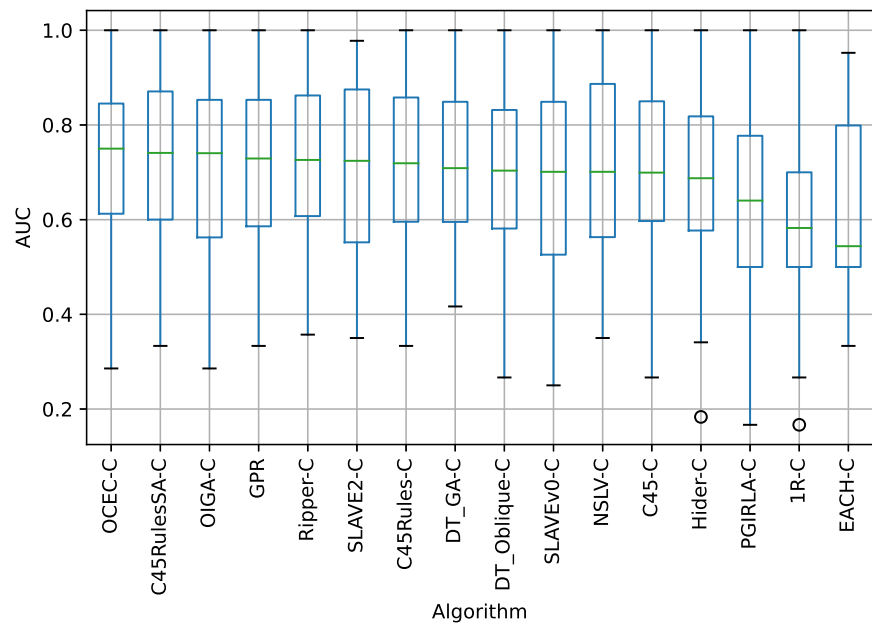
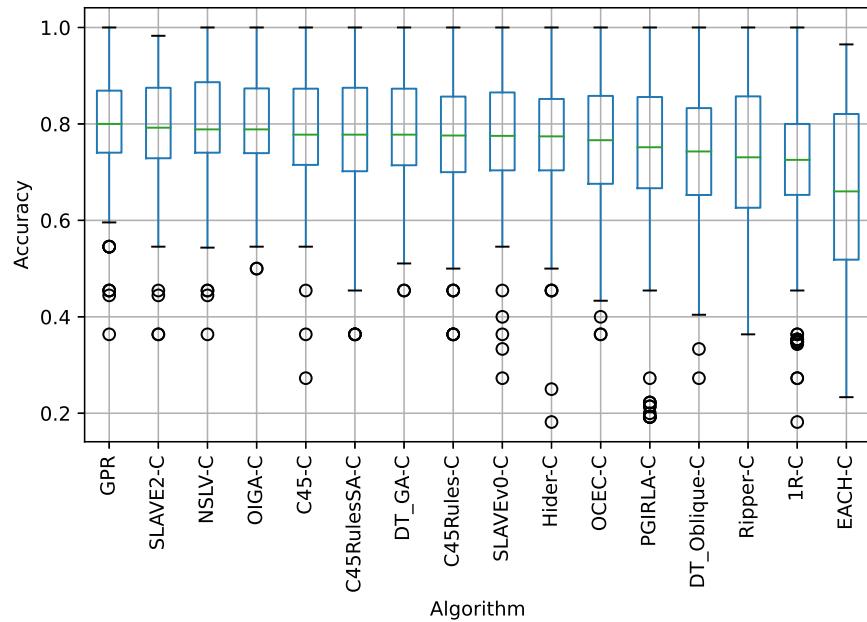


Figure 2. Distribution of the AUC values for each algorithm in all datasets.



**Figure 3.** Distribution of ACC values for each algorithm in all datasets.

Table 3 presents the result of comparing the fuzzy rule-based classifier. They are compared in terms of the following metrics, which are calculated as averages for every algorithm in every dataset: ANC—the average number of characters per rule in the dataset, ANR—the average number of rules in the dataset, ANA—the average number of attributes per rule in dataset, ANUA—the average number of unique attributes per rule in dataset. The results are sorted in ascending order by ANC. 1R-C generated an ANC of 106.54 with a small average number of rules on the dataset (ANR of 3.31) and a small average number of attributes on the dataset (ANA of 3.31). However, it achieved the worst results for MCC, AUC, and other performance metrics, as shown in Table 2. The comparison result places GPR near 1R-C also as an algorithm providing an extremely simple and concise set of metarules. Its simplicity is expressed as an ANC of 156.23, which is over 208 times smaller for GPR than for OIGA-C, which achieved the best results in terms of the WM (Table 2). GPR generates an ANR of 4.0 and ANA of 6.69 while maintaining high MCC, ACC, and other performance metrics, as shown in Table 2. DT Oblique-C generated the most complicated rules (ANC of 32457.38, ANA of 1059.08).

Table 4 presents examples of linguistic “if-then” fuzzy rules generated by fuzzy rule-based classifiers on the real Diabetes dataset. The results are sorted alphabetically. Parsing the algorithms’ output files ensured that all the compared rules had the same format. The number of digits in the ranges was not modified and depends on the KEEL implementation. The table also provides information on the number and length of the generated rules. In terms of syntax, GPR generated the shortest and most understandable rules, whereas EACH-C generated the lowest number of rules. OCEC-C generated the largest number of rules, while OIGA-C generated the largest number of characters. The study’s findings suggest that a structure based on four features is at the limit of human processing capacity and such a rule is very hard to understand [64]. Therefore, using algorithms containing several or several dozen attributes is challenging.

**Table 3.** Comparison of fuzzy rule-based classifiers in terms of rules’ size metrics.

	Algorithm	ANC	ANR	ANA	ANUA
1	1R-C	106.54	3.31	3.31	1.00
2	GPR	156.23	4.00	6.69	5.31
3	C45Rules-C	392.08	8.38	18.85	6.46
4	C45RulesSA-C	557.62	9.77	28.08	6.15
5	EACH-C	695.384	2.00	23.46	11.85
6	NSLV-C	824.08	8.92	28.46	8.23
7	Ripper-C	981.31	16.15	51.31	8.85
8	C45-C	1425.31	11.46	57.23	6.62
9	DT GA-C	2703.38	18.08	123.00	10.38
10	SLAVE2-C	4593.85	12.38	154.62	13.31
11	SLAVEv0-C	5101.92	14.69	168.08	13.38
12	PGIRLA-C	6330.54	18.69	115.31	12.31
13	Hider-C	11,468.85	18.08	425.15	11.08
14	OCEC-C	12,188.08	83.23	772.46	13.31
15	OIGA-C	20,958.08	30.00	399.23	15.85
16	DT Oblique-C	32,457.38	61.15	1059.08	11.69

**Table 4.** Example of “if-then” fuzzy rules generated by fuzzy rule-based classifiers on the real Diabetes dataset.

Algorithm	Rules Generated for the Diabetes Dataset	Number of Rules	Rules Length
1R-C	IF step count = [13072.0 , 55333.0) THEN 0 IF step count = [55333.0 , 58288.0) THEN 1 IF step count = [58288.0 , 60294.0) THEN 0 IF step count = [60294.0 , 114655.0] THEN 1	4	172
C45-C	IF step count <= 60837.000000 AND vigorous <= 128.750000 AND weight <= 80.500000 THEN 0 IF step count <= 60837.000000 AND vigorous <= 128.750000 AND weight > 80.500000 THEN ...	12	1828
C45Rules-C	IF height>1.61 AND age>14.0 AND weight<=52.0 THEN 1 IF vigorous>128.75 AND vigorous<=319.5 AND age>8.0 AND moderate>214.916666666667 THEN 1 IF step count>60837.0 THEN 1 ...	8	400
C45RulesSA-C	IF height>1.61 AND age>14.0 AND weight<=52.0 THEN 1 IF vigorous>128.75 AND vigorous<=319.5 AND age>8.0 AND moderate>214.916666666667 THEN 1 IF step count>60837.0 THEN 1 ...	8	400
DT GA-C	IF step count <= 60837.0 AND vigorous <= 128.75 AND weight <= 80.5 THEN 0 IF step count <= 60837.0 AND vigorous <= 128.75 AND weight > 80.5 THEN 1 IF step count <= 60837.0 ...	19	2856
DT Oblique-C	IF -1.0*step count + 60837.0 >= 0 AND -1.0*vigorous + 128.75 >= 0 AND -1.0*weight + 80.5 >= 0 AND -1.0*height + 1.87 >= 0 AND 168.486174002403*sex + -178.36864022034422*age + ...	30	8625
EACH-C	IF age in [6.0 , 18.0] AND weight in [19.3 , 98.8] AND height in [1.15 , 1.88] AND step count in [13072.0 , 60837.0] AND sedentary in [1343.166666666667 , 7813.333333333333] AND l...	2	603
GPR	IF step count is High THEN 1 IF vigorous is High AND sedentary is High THEN 1 ELSE 0	3	87

Table 4. Cont.

Algorithm	Rules Generated for the Diabetes Dataset	Number of Rules	Rules Length
Hider-C	IF age = [7.5, 17.5) AND weight = [29.15, 65.7) AND step count = [ , 55096.5) AND sedentary = [2270.0833333333335, 4964.9166666666664) AND light = [356.875, 1330.8333333333335) AND...	14	3595
NSLV-C	IF step count = { VeryLow Low} THEN 0 IF step count = { High VeryHigh} THEN 1 IF age = { Low High VeryHigh} AND moderate = { Low VeryHigh} THEN 1	3	145
OCEC-C	IF step count = 3 THEN 1 IF age = 2 AND sedentary = 1 THEN 1 IF sex = 0 AND vigorous = 1 THEN 1 IF sex = 0 AND step count = 1 AND light = 1 THEN 0 IF height = 2 AND ste...	62	6763
OIGA-C	IF 1.6699878586619132 < sex < 1.1982191470913168 AND 9.4429624945491 < age < 16.56761035848586 AND 67.72250192298611 < weight < 85.23233850170679 AND 1.859257826523217 < height ...	30	14312
PGIRLA-C	IF sedentary = [3801.8675692824663, 5006.615988626676] AND light = [1162.1170959360238, 2362.4439084883884] AND moderate = [414.0390532025578, 474.55751714327096] AND vigorous ...	19	4340
Ripper-C	IF step count<=60837.0 AND height<=1.58 THEN 0 IF step count<=60837.0 AND moderate<=119.0 THEN 0 IF step count<=60837.0 AND vigorous<=127.5 AND height>1.64 AND moderate>123...	9	467
SLAVE2-C	IF age = { VeryLow Medium} AND weight = { Medium} AND height = { High VeryHigh} AND step count = { VeryLow Low} AND sedentary = { Medium} AND light = { Low} AND moderate = { Low...	8	2098
SLAVEv0-C	IF step count = { VeryLow Low} THEN 0 IF age = { VeryLow Low Medium VeryHigh} AND height = { VeryLow Low Medium VeryHigh} AND step count = { Medium} AND sedentary = { Medium} ...	11	2814

The results indicate that GPR generates the shortest and most interpretable rules while still achieving good classification performance. As a result, we decided to use the Wilcoxon signed-rank test to statistically compare the results of GPR with those of other fuzzy rule-based algorithms. Table 5 presents the results of the Wilcoxon signed-rank test. The results of GPR and fuzzy rule-based algorithms for the MCC, AUC, and ACC measurements were compared.  $X$  denotes a vector containing the mean values of the MCC (or AUC and ACC) measure for the GPR algorithm, as calculated from ten random stratified folds for each dataset.  $Y_i$  denotes a vector containing the corresponding values for the  $i$ th algorithm tested on exactly the same folds. The index  $i$  represents the name of the algorithm, where  $i$  belongs to the set: {1R-C, C45-C, C45Rules-C, C45RulesSA-C, DT\_GA-C, DT\_Oblique-C, EACH-C, Hider-C, NSLV-C, OCEC-C, OIGA-C, PGIRLA-C, Ripper-C, SLAVE2-C, SLAVEv0-C}. Table 5 shows the probability ( $p$ -value) of a two-sided paired Wilcoxon test for the null hypothesis  $H_0$  that the difference  $(X - Y_i)$  follows a distribution with a zero median. The two-sided  $p$ -value is calculated by doubling the most significant one-sided value.

According to the results in Table 5, for the MCC measure, the Wilcoxon signed-rank test fails to reject the null hypothesis of no significant difference in the mean values of MCC at the significance level of  $\alpha = 0.05$  when comparing GPR to the following nine algorithms: C45-C, C45Rules-C, C45RulesSA-C, DT\_GA-C, NSLV-C, OCEC-C, OIGA-C, Ripper-C, and SLAVE2-C. However, according to the results in Table 5, the null hypothesis can be rejected at the 5% level when comparing GPR to the following six algorithms: 1R-C, DT\_Oblique-C, EACH-C, Hider-C, PGIRLA-C, and SLAVEv0-C. Thus, the alternative hypothesis  $H_1$  is accepted: there is a significant difference in the mean values of MCC for

GPR compared to the 1R-C, DT\_Oblique-C, EACH-C, Hider-C, PGIRLA-C, and SLAVEv0-C algorithms. According to Wilcoxon's rank test (Table 5) and the distribution of MCC values as shown in Figure 1, from the perspective of the MCC criterion, GPR is worse at the significance level of  $\alpha = 0.05$  than OIGA-C, and SLAVE2-C. For the same reasons, GPR is better than the following six algorithms: 1R-C, DT\_Oblique-C, EACH-C, Hider-C, PGIRLA-C, and SLAVEv0-C.

**Table 5.** Comparison of fuzzy rule-based classifiers and GPR with Wilcoxon's signed-rank test.

No.	Algorithm	MCC <i>p</i> -Value	AUC <i>p</i> -Value	ACC <i>p</i> -Value
1	1R-C	0.0000	0.0000	0.0000
2	C45-C	0.8475	0.6052	0.2622
3	C45Rules-C	0.8690	0.0899	0.0027
4	C45RulesSA-C	0.6243	0.0583	0.0123
5	DT GA-C	0.9265	0.6479	0.3322
6	DT Oblique-C	0.0026	0.0592	0.0000
7	EACH-C	0.0000	0.0000	0.0000
8	Hider-C	0.0056	0.0016	0.0022
9	NSLV-C	0.5980	0.8152	0.7802
10	OCEC-C	0.0725	0.8430	0.0000
11	OIGA-C	0.6399	0.3130	0.8192
12	PGIRLA-C	0.0003	0.0004	0.0001
13	Ripper-C	0.5355	0.4273	0.0000
14	SLAVE2-C	0.2653	0.2346	0.4621
15	SLAVEv0-C	0.0012	0.0014	0.0023

According to the results in Table 5, for the AUC measure, the Wilcoxon signed-rank test fails to reject the null hypothesis of no significant difference in the mean values of AUC at the significance level of  $\alpha = 0.05$  when comparing GPR to the following algorithms: C45-C, C45Rules-C, C45RulesSA-C, DT\_GA-C, DT\_Oblique-C, NSLV-C, OCEC-C, OIGA-C, Ripper-C, and SLAVE2-C. Considering the *p*-values for the AUC measure in Table 5 and the distribution of AUC values for each algorithm across all datasets and 10 cross-validation folds shown in Figure 2 it can be concluded that GPR is worse than OCEC-C, C45RulesSA-C, and OIGA-C, but better than 1R-C, EACH-C, Hider-C, PGIRLA-C, Ripper-C, and SLAVEv0-C.

According to the results in Table 5, for the ACC measure, the Wilcoxon signed-rank test fails to reject the null hypothesis of no significant difference in the mean values of ACC at the significance level of  $\alpha = 0.05$  when comparing GPR to the following algorithms: C45-C, DT\_GA-C, NSLV-C, OIGA-C, and SLAVE2-C. Based on the *p*-values for the ACC measure in Table 5 and the distributions of ACC values shown in Figure 3, GPR is superior at the significance level of  $\alpha = 0.05$  to the following algorithms: 1R-C, C45Rules-C, C45RulesSA-C, DT\_Oblique-C, EACH-C, Hider-C, OCEC-C, PGIRLA-C, Ripper-C, and SLAVEv0-C.

## 7. Discussion and Conclusions

Machine learning can be used to improve the accuracy and objectivity of clinical experts in clinical decision-support systems. Generated rules can help identify the most likely diagnosis and show how individual attributes contributed to the decision. However, it can be difficult to select the most relevant rules from the many that are generated, especially when they contain numerous attributes and are difficult to interpret. It is important to choose the appropriate algorithm for the task at hand to ensure the best results. This paper has proposed a comparative study of fuzzy rule-based algorithms that were applied to issues in the field of clinical decision support. The proposed comparison begins with applying 16 different rule-based fuzzy logic algorithms: 1R-C, C45-C, C45Rules-C, C45RulesSA-C, DT\_GA-C, DT\_Oblique-C, EACH-C, GPR, Hider-C, NSLV-C, OCEC-C, OIGA-C, PGIRLA-C, Ripper-C, SLAVE2-C, SLAVEv0-C to 12 clinical datasets and generation of rules. We calculated performance metrics such as MCC, ACC, AUC, Spe,

Pre, Sen, and WM based on the results obtained and compared them. Based on the WM criterion, which takes into account the results obtained from all metrics, the best algorithms are OIGA-C, GPR, and NSLV-C, and the worst are EACH-C, 1R-C, and PGIRLA-C. Then, we presented the MCC, ACC, and AUC values distribution for each algorithm in all datasets. The average length of the rules in the dataset, the average number of rules in the dataset, and the average number of attributes and unique attributes per rule were also included in the comparison. We also presented rules generated for a Diabetes dataset considering the number of rules, their length, and their syntax. Most interpretable rules were generated by 1R, GPR and C45Rules-C. The longest and most complicated rules were generated by DT\_Oblique-C, OIGA-C and OCEC-C. In conclusion, algorithms that achieve high classification results tend to generate very complex and lengthy rules (such as OIGA-C), while algorithms that produce simpler rules often have lower classification results (like 1R-C).

The research indicates that GPR generates the shortest and most interpretable rules while still achieving good classification performance. As a result, we decided to test GPR statistically using the Wilcoxon signed-rank test. It was performed to compare the means of every rule-based fuzzy logic classifier and GPR. According to the results of this test and the distribution of ACC values for each rule-based fuzzy logic algorithm in all datasets, the GPR algorithm outperformed at the significance level of  $\alpha = 0.05$  the 1R-C, C45Rules-C, C45RulesSA-C, DT\_Oblique-C, EACH-C, Hider-C, OCEC-C, PG1RLA-C, Ripper-C, and SLAVEvO-C algorithms. Considering all the results, we can conclude that GPR can be used successfully for generating rules from medical data.

However, theoretical results, particularly those related to the “no free lunch” theorem [65], state that in the general case no algorithm can outperform every other algorithm in all possible tasks. In other words, there is no one-size-fits-all solution to all problems. The GPR algorithm also has some drawbacks. For example, it uses a genetic algorithm to generate metarules, which can be computationally intensive and slow to converge, especially for large and complex problems. Furthermore, GPR requires the normalization of continuous input data to the interval [0, 1], encoding of all data (continuous and categorical), and the adoption of a threshold for the discriminant function (with a default value of 0.5). The selection of a fitting function for the evolutionary algorithm (such as accuracy or sensitivity) is also required.

This study has a few limitations that should be considered when interpreting the results. First, we did not conduct a memory requirement test or measure run time. Second, we use the default values for the hyperparameters, which could potentially be adjusted to improve performance. Furthermore, the performance of the algorithms was tested only on medical datasets with a relatively small number of records, so the results may not be representative for larger datasets.

One potential area for future research is to conduct further research on the impact of memory requirements and run time. Another idea for future research is to include a greater number of algorithms and real-world datasets obtained through cooperation with various medical organizations. To make our findings more accessible and user-friendly, we also intend to develop a user interface based on our open-source code. This interface will enable medical professionals to easily generate rules for specific medical problems and display them in a unified way, using the most appropriate algorithm for the task at hand. Through these efforts, we hope to enhance the utility and impact of our work in the field of medical decision-making.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** All reported results can be found at <https://github.com/czmilanna/rules>, accessed on 12 January 2023.

**Conflicts of Interest:** The author declares no conflict of interest.

### Abbreviations

The following abbreviations are used in this manuscript:

ACC	accuracy
ANA	average number of attributes per rule in dataset
ANC	average number of characters per rule in dataset
ANR	average number of rules on dataset
ANUA	average number of unique attributes per rule in dataset
AUC	area under ROC curve
C45RulesSA-C	C4.5Rules simulated annealing version
DT_GA-C	hybrid decision tree-genetic algorithm
DT_Oblique-C	oblique decision tree with evolutionary learning
EACH-C	exemplar-aided constructor of hyperrectangles
FN	false negatives
FP	false positives
FPR	false positive rate
FRBS	fuzzy rule-based systems
GPR	classifier based on fuzzy logic and gene expression programming
Hider-C	hierarchical decision rules
KNN	k-nearest neighbors
MCC	Matthews correlation coefficient
MDSS	medical decision support systems
NSLV-C	New SLAVE
OCEC-C	organizational co-evolutionary algorithm for classification
OIGA-C	ordered incremental genetic algorithm
PGIRLA-C	Pittsburgh genetic interval rule learning algorithm
Pre	precision
RBS	rule-based systems
Ripper-C	repeated incremental pruning to produce error reduction
Sen	sensitivity
SLAVEv0-C	structural learning algorithm in a vague environment
SVM	support vector machine
TN	true negatives
TP	true positives
WDBC	Wisconsin diagnosis breast cancer
WM	weighted metric
Wisconsin	Wisconsin breast cancer (original)

### References

1. Yan, H.; Jiang, Y.; Zheng, J.; Peng, C.; Li, Q. A multilayer perceptron-based medical decision support system for heart disease diagnosis. *Expert Syst. Appl.* **2006**, *30*, 272–281. [\[CrossRef\]](#)
2. Malmir, B.; Amini, M.; Chang, S.I. A medical decision support system for disease diagnosis under uncertainty. *Expert Syst. Appl.* **2017**, *88*, 95–108. [\[CrossRef\]](#)
3. Casalino, G.; Castellano, G.; Castiello, C.; Pasquadibisceglie, V.; Zaza, G. A Fuzzy Rule-Based Decision Support System for Cardiovascular Risk Assessment. In *Fuzzy Logic and Applications*; Springer International Publishing: Cham, Switzerland, 2019; pp. 97–108. [\[CrossRef\]](#)
4. Durán, J.M.; Jongsma, K.R. Who is afraid of black box algorithms? On the epistemological and ethical basis of trust in medical AI. *J. Med. Ethics* **2021**, *47*, 329–335. [\[CrossRef\]](#)
5. Czmil, S.; Kluska, J.; Czmil, A. CACP: Classification Algorithms Comparison Pipeline. *SoftwareX* **2022**, *19*, 101134. [\[CrossRef\]](#)
6. Grosan, C.; Abraham, A. *Intelligent Systems*; Intelligent Systems Reference Library, Springer: New York, NY, USA, 2011; p. 148.
7. Loyola-González, O. Black-Box vs. White-Box: Understanding Their Advantages and Weaknesses From a Practical Point of View. *IEEE Access* **2019**, *7*, 154096–154113. [\[CrossRef\]](#)
8. Kluska, J.; Madera, M. Extremely simple classifier based on fuzzy logic and gene expression programming. *Inf. Sci.* **2021**, *571*, 560–579. [\[CrossRef\]](#)

9. Kliegr, T.; Bahník, Š.; Fürnkranz, J. A review of possible effects of cognitive biases on interpretation of rule-based machine learning models. *Artif. Intell.* **2021**, *295*, 103458. [CrossRef]
10. Zadeh, L. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353. [CrossRef]
11. Kluska, J. Selected Applications of P1-TS Fuzzy Rule-Based Systems. In *Lecture Notes in Computer Science, Proceedings of the International Conference on Artificial Intelligence and Soft Computing, Zakopane, Poland, 14–18 June 2015*; Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 195–206.
12. Dec, G.; Drahus, G.; Mazur, D.; Kwiatkowski, B. Forecasting Models of Daily Energy Generation by PV Panels Using Fuzzy Logic. *Energies* **2021**, *14*, 1676. [CrossRef]
13. Kluska, J.; Kusy, M.; Obrzut, B. The Classifier for Prediction of Peri-operative Complications in Cervical Cancer Treatment. In *Lecture Notes in Computer Science, Proceedings of the International Conference on Artificial Intelligence and Soft Computing, Zakopane, Poland, 14–18 June 2015*; Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 143–154.
14. Al-shami, T.M. (2, 1)-Fuzzy sets: Properties, weighted aggregated operators and their applications to multi-criteria decision-making methods. *Complex Intell. Syst.* **2022**. [CrossRef]
15. Holte, R. Very simple classification rules perform well on most commonly used datasets. *Mach. Learn.* **1993**, *11*, 63–91. [CrossRef]
16. Witten, I.H.; Frank, E.; Hall, M.A. *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed.; Morgan Kaufmann Series in Data Management Systems; Morgan Kaufmann: Amsterdam, The Netherlands, 2011.
17. Salzberg, S. A Nearest Hyperrectangle Learning Method. *Mach. Learn.* **1991**, *6*, 251–276. [CrossRef]
18. Cohen, W. Fast Effective Rule Induction. In *Lecture Notes on Multidisciplinary Industrial Engineering, Proceedings of the Twelfth International Conference on Management Science and Engineering Management, Tahoe City, CA, USA, 9–12 July 1995*; pp. 1–10.
19. Ouyang, C.S.; Lee, C.T.; Lee, S.J. An Improved Fuzzy Genetics-Based Machine Learning Algorithm for Pattern Classification. In Proceedings of the Second International Conference on Innovative Computing, Information and Control (ICICIC 2007), Kumamoto, Japan, 5–7 September 2007. [CrossRef]
20. Carvalho, D.; Freitas, A. A hybrid decision tree/genetic algorithm method for data mining. *Inf. Sci.* **2004**, *163*, 13–35. [CrossRef]
21. Cantú-Paz, E.; Kamath, C. Inducing oblique decision trees with evolutionary algorithms. *IEEE Trans. Evol. Comput.* **2003**, *7*, 54–68. [CrossRef]
22. Gonzalez, A.; Perez, R. Completeness and consistency conditions for learning fuzzy rules. *Fuzzy Sets Syst.* **1998**, *96*, 37–51. [CrossRef]
23. Aguilar-Ruiz, J.; Riquelme, J.; Toro, M. Evolutionary learning of hierarchical decision rules. *Trans. Syst. Man Cybern.—Part B Cybern.* **2003**, *33*, 324–331. [CrossRef]
24. Jiao, L.; Liu, J.; Zhong, W. An organizational coevolutionary algorithm for classification. *IEEE Trans. Evol. Comput.* **2006**, *10*, 67–80. [CrossRef]
25. Zhu, F.; Guan, S. Ordered incremental training with genetic algorithms. *Int. J. Intell. Syst.* **2004**, *19*, 1239–1256. [CrossRef]
26. Corcoran, A.; Sen, S. Using Real-Valued Genetic Algorithms to Evolve Rule Sets for Classification. In Proceedings of the 1st IEEE Conference on Evolutionary Computation, Orlando, FL, USA, 27–29 June 1994; pp. 120–124.
27. Aamir, K.M.; Sarfraz, L.; Ramzan, M.; Bilal, M.; Shafi, J.; Attique, M. A Fuzzy Rule-Based System for Classification of Diabetes. *Sensors* **2021**, *21*, 8095. [CrossRef]
28. Adeli, A.; Neshat, M. A fuzzy expert system for heart disease diagnosis. In Proceedings of the International MultiConference of Engineers and Computer Scientists 2010 Vol I, Hong Kong, 17–19 March 2010; pp. 134–139. Available online: [https://www.iaeng.org/publication/IMECS2010/IMECS2010\\_pp134-139.pdf](https://www.iaeng.org/publication/IMECS2010/IMECS2010_pp134-139.pdf) (accessed on 12 January 2023).
29. Improta, G.; Mazzella, V.; Vecchione, D.; Santini, S.; Triassi, M. Fuzzy logic-based clinical decision support system for the evaluation of renal function in post-Transplant Patients. *J. Eval. Clin. Pract.* **2019**, *26*, 1224–1234. [CrossRef]
30. Rotshtein, A. Design and Tuning of Fuzzy Rule-Based Systems for Medical Diagnosis. In *Fuzzy and Neuro-Fuzzy Systems in Medicine*; CRC Press: Boca Raton, FL, USA, 2017; pp. 243–290. [CrossRef]
31. Mohammadpour, R.A.; Abedi, S.M.; Bagheri, S.; Ghaemian, A. Fuzzy Rule-Based Classification System for Assessing Coronary Artery Disease. *Comput. Math. Methods Med.* **2015**, *2015*, 564867. [CrossRef] [PubMed]
32. Al-Dmour, J.A.; Sagahyoon, A.; Al-Ali, A.; Abusnana, S. A fuzzy logic-based warning system for patients classification. *Health Inform. J.* **2017**, *25*, 1004–1024. [CrossRef] [PubMed]
33. Adlassnig, K.P. Fuzzy Set Theory in Medical Diagnosis. *IEEE Trans. Syst. Man Cybern.* **1986**, *16*, 260–265. [CrossRef]
34. Wieben, O.; Afonso, V.X.; Tompkins, W.J. Classification of premature ventricular complexes using filter bank features, induction of decision trees and a fuzzy rule-based system. *Med. Biol. Eng. Comput.* **1999**, *37*, 560–565. [CrossRef] [PubMed]
35. Tsiouras, M.; Exarchos, T.; Fotiadis, D.; Kotsia, A.; Vakalis, K.; Naka, K.; Michalis, L. Automated Diagnosis of Coronary Artery Disease Based on Data Mining and Fuzzy Modeling. *IEEE Trans. Inf. Technol. Biomed.* **2008**, *12*, 447–458. [CrossRef]
36. Sanz, J.; Pagola, M.; Bustince, H.; Brugos, A.; Fernández, A.; Herrera, F. A case study on medical diagnosis of cardiovascular diseases using a Genetic Algorithm for Tuning Fuzzy Rule-Based Classification Systems with Interval-Valued Fuzzy Sets. In Proceedings of the 2011 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems (T2FUZZ), Paris, France, 11–15 April 2011; pp. 9–15. [CrossRef]



37. Hosseini, R.; Ellis, T.; Mazinani, M.; Dehmeshki, J. A genetic fuzzy approach for rule extraction for rule-based classification with application to medical diagnosis. In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), Athens, Greece, 5–9 September 2011; pp. 5–9.
38. Mala, I.; Akhtar, P.; Ali, T.J.; Zia, S.S. Fuzzy rule based classification for heart dataset using fuzzy decision tree algorithm based on fuzzy RDBMS. *World Appl. Sci. J.* **2013**, *28*, 1331–1335.
39. Sanz, J.A.; Galar, M.; Jurio, A.; Brugos, A.; Pagola, M.; Bustince, H. Medical diagnosis of cardiovascular diseases using an interval-valued fuzzy rule-based classification system. *Appl. Soft Comput.* **2014**, *20*, 103–111.
40. Jaćimović, J.; Krstev, C.; Jelovac, D. A rule-based system for automatic de-identification of medical narrative texts. *Informatica* **2015**, *39*, 45–53.
41. Sadeghzadeh, M. A New Method for Diagnosing Breast Cancer using Firefly Algorithm and Fuzzy Rule based Classification. In Proceedings of the 2017 IEEE 11th International Conference on Application of Information and Communication Technologies (AICT), Moscow, Russia, 20–22 September 2017; pp. 1–5. [[CrossRef](#)]
42. Davoodi, R.; Moradi, M.H. Mortality prediction in intensive care units (ICUs) using a deep rule-based fuzzy classifier. *J. Biomed. Inform.* **2018**, *79*, 48–59. [[CrossRef](#)]
43. Gu, X.; Zhang, C.; Ni, T. Feature Selection and Rule Generation Integrated Learning for Takagi-Sugeno-Kang Fuzzy System and its Application in Medical Data Classification. *IEEE Access* **2019**, *7*, 169029–169037. [[CrossRef](#)]
44. Karthikeyan, R.; Geetha, P.; Ramaraj, E. Rule Based System for Better Prediction of Diabetes. In Proceedings of the 2019 3rd International Conference on Computing and Communications Technologies (ICCCT), Chennai, India, 21–22 February 2019; pp. 195–203. [[CrossRef](#)]
45. Singh, N.; Singh, P. Medical Diagnosis of Coronary Artery Disease Using Fuzzy Rule-Based Classification Approach. In *Advances in Biomedical Engineering and Technology*; Springer: Singapore, 2020; pp. 321–330. [[CrossRef](#)]
46. Hossain, S.; Sarma, D.; Chakma, R.J.; Alam, W.; Hoque, M.M.; Sarker, I.H. A rule-based expert system to assess coronary artery disease under uncertainty. In *Communications in Computer and Information Science*; Springer: Singapore, 2020; pp. 143–159.
47. Steimann, F. On the use and usefulness of fuzzy sets in medical AI. *Artif. Intell. Med.* **2001**, *21*, 131–137. [[CrossRef](#)] [[PubMed](#)]
48. Mousavi, S.M.; Abdullah, S.; Niaki, S.T.A.; Banihashemi, S. An intelligent hybrid classification algorithm integrating fuzzy rule-based extraction and harmony search optimization: Medical diagnosis applications. *Knowl.-Based Syst.* **2021**, *220*, 106943. [[CrossRef](#)]
49. Varshney, A.K.; Torra, V. Literature Review of various Fuzzy Rule based Systems. *arXiv* **2022**, arXiv:2209.07175. [[CrossRef](#)]
50. Chandrasekar, R.; Neelu, K. Review of Fuzzy Rule Based Classification systems. *Res. J. Pharm. Technol.* **2016**, *9*, 1299.
51. Gilda, K.S.; Satarkar, S.L. Review of Fuzzy Systems through various jargons of technology. *J. Emerg. Technol. Innov. Res.* **2020**, *7*, 260–264.
52. Kar, S.; Das, S.; Ghosh, P.K. Applications of neuro fuzzy systems: A brief review and future outline. *Appl. Soft Comput.* **2014**, *15*, 243–259. [[CrossRef](#)]
53. Alcalá-Fdez, J.; Fernández, A.; Luengo, J.; Derrac, J.; García, S. KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. *J. Multiple Valued Log. Soft Comput.* **2011**, *17*, 255–287.
54. Weiss, S.M.; Kulikowski, C.A. *Computer Systems That Learn: Classification and Prediction Methods From Statistics, Neural Nets, Machine Learning, and Expert Systems*; Morgan Kaufmann: San Mateo, CA, USA, 1991.
55. Czmil, A.; Czmil, S.; Mazur, D. A Method to Detect Type 1 Diabetes Based on Physical Activity Measurements Using a Mobile Device. *Appl. Sci.* **2019**, *9*, 2555. [[CrossRef](#)]
56. Czmil, A. GPR: A Python Implementation of an Extremely Simple Classifier Based on Fuzzy Logic and Gene Expression Programming (Version 1.0.0) [Computer Software], 2022. Available online: <https://github.com/czmilanna/gpr-algorithm> (accessed on 12 January 2023).
57. Quinlan, J. *C4.5: Programs for Machine Learning*; Morgan Kaufmann: San Mateo, CA, USA, 1993.
58. Quinlan, J. MDL and Categorical Theories (Continued). In Proceedings of the Twelfth International Conference on Management Science and Engineering Management, Tahoe City, CA, USA, 9–12 July 1995; pp. 464–470.
59. González, A.; Perez, R. Improving the genetic algorithm of SLAVE. *Mathw. Soft Comput.* **2009**, *16*, 59–70.
60. Gonzalez, A.; Perez, R. SLAVE: A genetic learning system based on an iterative approach. *IEEE Trans. Fuzzy Syst.* **1999**, *7*, 176–191. [[CrossRef](#)]
61. Kulkarni, A.; Chong, D.; Batarseh, F.A. 5—Foundations of data imbalance and solutions for a data democracy. In *Data Democracy*; Batarseh, F.A., Yang, R., Eds.; Academic Press: Cambridge, MA, USA, 2020; pp. 83–106. [[CrossRef](#)]
62. Hicks, S.A.; Strümke, I.; Thambawita, V.; Hammou, M.; Riegler, M.A.; Halvorsen, P.; Parasa, S. On evaluation metrics for medical applications of artificial intelligence. *Sci. Rep.* **2022**, *12*, 5979. [[CrossRef](#)] [[PubMed](#)]
63. Huang, J.; Ling, C.X. Using AUC and accuracy in evaluating learning algorithms. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 299–310. [[CrossRef](#)]

64. Halford, G.S.; Baker, R.; McCredden, J.E.; Bain, J.D. How Many Variables Can Humans Process? *Psychol. Sci.* **2005**, *16*, 70–76. [[CrossRef](#)] [[PubMed](#)]
65. Wolpert, D.; Macready, W. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

## Streszczenie

Niniejsza rozprawa doktorska stanowi jednotematyczny cykl publikacji naukowych dotyczących zagadnień wykorzystania metod sztucznej inteligencji w celu usprawnienia procesu diagnostycznego w medycynie. Zagadnienia obejmowały problematykę zastosowania metod sztucznej inteligencji do konstrukcji wybranych systemów wspomagania decyzji medycznych. Uwzględniając powyższe, w pracy sformułowano hipotezę, która zakłada, że

*Możliwe jest wykorzystanie różnych metod sztucznej inteligencji do analizy danych medycznych i automatyzacji wybranych procesów diagnostycznych, pozwalające na uzyskanie akceptowalnych wyników z dokładnością i efektywnością nie gorszą niż innych istniejących metod znanych z literatury.*

Hipoteza została uprawdopodobniona przez realizację następujących zadań:

**Zadanie 1. Zastosowanie metod sztucznej inteligencji do klasyfikacji cukrzycy typu 1 na podstawie danych uzyskanych za pomocą nieinwazyjnych pomiarów aktywności fizycznej**

Zadanie zostało zrealizowane przez zastosowanie dziesięciu najpopularniejszych metod sztucznej inteligencji do klasyfikacji cukrzycy typu 1. Wyniki uzyskane przez każdy z wybranych algorytmów zostały zwalidowane za pomocą metryk wydajnościowych, a następnie porównane w celu wyboru optymalnego algorytmu do klasyfikacji.

**Zadanie 2. Opracowanie metody pozwalającej na automatyczne, jednoczesne rozpoznawanie i zliczanie czerwonych i białych krwinek oraz płytek krwi na podstawie zdjęć mikroskopowych z wykorzystaniem głębokich sieci neuronowych**

Zadanie zrealizowano przez wytrenowanie sieci RetinaNet do rozpoznawania i klasyfikacji trzech rodzajów komórek krwi, a następnie manualną ocenę wyników klasyfikacji czerwonych i białych krwinek oraz płytek krwi i obliczenie metryk wydajnościowych dla uzyskanych wyników. Ustalono optymalną wartość progową do zliczania wszystkich typów komórek jednocześnie, a otrzymane wyniki porównano z wynikami innych autorów zajmujących się tematyką liczenia komórek krwi.

**Zadanie 3. Opracowanie aplikacji umożliwiającej automatyzację procesu oceny, składania i identyfikacji sekwencji genomowych uzyskanych za pomocą nowych metod sekwencjonowania przy użyciu narzędzi korzystających z metod uczenia maszynowego**

Zadanie zostało zrealizowane dzięki wykonaniu implementacji aplikacji-serwera NanoForms, umożliwiającego ocenę jakości danych z sekwencjonowania nanoporowego oraz Illumina, a także składanie genomów bakteryjnych metodami de novo i hybrydową. Serwer został skonfigurowany oraz przekazany do niekomercyjnego użytku publicznego, a kod źródłowy aplikacji-serwera został udostępniony w postaci otwartego oprogramowania.

**Zadanie 4. Implementacja w języku Python klasyfikatora opartego na logice rozmytej i programowaniu ekspresji genów, służącego do generowania wysoce interpretowalnych reguł rozmytych**

Zadanie zrealizowano przez wykonanie implementacji klasyfikatora GPR w języku Python, który osiąga bardzo wysokie wyniki pod względem dokładności i pola pod krzywą ROC. W implementacji algorytmu zaproponowano wiele usprawnień pierwotnej wersji, a całość oprogramowania udostępniono na licencji gwarantującej dostęp do kodu źródłowego.

**Zadanie 5. Opracowanie narzędzia pozwalającego na eksperymentalne porównanie wybranych rozmytych algorytmów opartych na regułach do klasyfikacji danych medycznych**

Zadanie zostało zrealizowane przez przygotowanie porównania wyników metryk wydajnościowych osiąganych przez wybrane algorytmy oparte na regułach, przeprowadzonego na zbiorach danych medycznych. Przeprowadzono wiele analiz statystycznych i porównawczych, pozwalających na wybór najlepszych algorytmów regułowych w zastosowaniach medycznych.

**Słowa kluczowe:** sztuczna inteligencja, algorytmy interpretowalne, diagnostyka medyczna, metody sekwencjonowania następnej generacji, głębokie sieci neuronowe

## Abstract

This manuscript-based doctoral thesis addresses using artificial intelligence (AI) methods to improve the medical diagnostic process. The aim of the dissertation was to present the possibility of using artificial intelligence methods to enhance the quality and efficiency of the medical diagnostic process. Based on this premise, the paper formulates a hypothesis, which assumes that

*It is possible to use a variety of artificial intelligence methods to analyze medical data and automate selected diagnostic processes, which will produce interpretable results with accuracy and efficiency not worse than other existing methods known from the literature.*

The hypothesis was substantiated by achieving the following goals:

**Goal 1. Application of artificial intelligence methods to classify type 1 diabetes based on data obtained by noninvasive physical activity measurements**

This goal was met by applying ten of the most popular artificial intelligence methods to classify type 1 diabetes. The results obtained by each of the selected algorithms were validated using performance metrics and then compared to select the optimal algorithm to solve this problem.

**Goal 2. Development of a method to automatically and simultaneously identify and count red and white blood cells and platelets from microscopic images using deep machine learning methods**

The goal was achieved by training RetinaNet to recognize and classify three types of blood cells, then manually evaluating the results of classifying red and white blood cells and platelets, and calculating performance metrics for the results obtained. The optimal threshold value was determined to count all cell types simultaneously, and the results obtained were compared with those of other authors working on blood counting.

**Goal 3. Development of an application to automate the process of evaluating, assembling, and identifying genomic sequences obtained by new sequencing methods using machine learning-based tools**

This goal was achieved by developing the NanoForms application-server, which allows the quality assessment of the ONT and Illumina sequencing data, as well as the

assembly of bacterial genomes using de novo and hybrid methods. The server was configured and made available for noncommercial public use, and the source code of the application-server is open source.

**Goal 4. A Python implementation of a classifier based on fuzzy logic and gene expression programming to generate highly interpretable fuzzy rules**

This goal was met by proposing a novel Python-based implementation of the GPR classifier, which achieves very high results in terms of accuracy and area under the ROC curve. The implementation of the algorithm includes several improvements to the original version, and all the software was made available under a license that guarantees access to the source code.

**Goal 5. Developing a tool to experimentally compare selected fuzzy rule-based algorithms for medical data classification**

This goal was achieved by designing an experimental comparison of the performance metrics achieved by selected rule-based algorithms, carried out on medical data sets. A number of statistical and comparative analyzes were performed, allowing the selection of the best rule-based algorithms for medical applications.

**Keywords:** artificial intelligence, interpretable algorithms, medical diagnostics, next-generation sequencing methods, deep neural networks

## Oświadczenia współautorów

Poniższy rozdział zawiera oświadczenia dotyczące indywidualnego wkładu merytorycznego autorki rozprawy oraz współautorów w przygotowanie, przeprowadzenie i opracowanie badań oraz przedstawienie prac w formie publikacji, a także informację procentowym wkładzie autorskim. Oświadczenia dotyczą kolejno następujących artykułów:

- [A-1] **A. Czmił**, S. Czmił, D. Mazur. *A method to detect type 1 diabetes based on physical activity measurements using a mobile device*. Applied Sciences 9 (12) (2019) 2555. <https://doi.org/10.3390/app9122555> (str. 149),
- [A-2] G. Drałus, D. Mazur, **A. Czmił**. *Automatic detection and counting of blood cells in smear images using RetinaNet*. Entropy 23 (11) (2021) 1522. doi:10.3390/e23111522 (str. 152),
- [A-3] **A. Czmił**, M. Wroński, S. Czmił, M. Sochacka-Piętal, M. Ćmił, J. Gawor, T. Wołkiewicz, D. Plewczyński, D. Strzałka, M. Piętal. *NanoForms: an integrated server for processing, analysis and assembly of raw sequencing data of microbial genomes, from Oxford Nanopore technology*. PeerJ, 2022, 10:e13056. doi:10.7717/peerj.13056 (str. 155),
- [A-4] **A. Czmił**, J. Kluska, S. Czmił. *GPR: A Python implementation of an extremely simple classifier based on fuzzy logic and gene expression programming*, SoftwareX, 2023; 22:101362. <https://doi.org/10.1016/j.softx.2023.101362> (str. 161).





## Oświadczenie współautorów publikacji

Niniejszym podaję procentowy wkład autorski w publikację pt. „*A method to detect type 1 diabetes based on physical activity measurements using a mobile device*”. Wkład Anny Czmił w powstanie publikacji obejmuje:

- współautorstwo koncepcji artykułu,
- współopracowanie metodologii i zadań badawczych,
- dobór algorytmów sztucznej inteligencji do eksperymentów z uwzględnieniem przyjętej metodologii,
- udział w przeprowadzeniu eksperymentów obliczeniowych,
- udział w opracowaniu i analizie wyników,
- udział w przygotowaniu rysunków i tabel,
- współredakcję pracy.

Procentowy wkład autorski Anny Czmił wynosi: 33,33 %.

<b>Imię i nazwisko współautora</b>	<b>Procentowy wkład autorski</b>	<b>Data i podpis współautora</b>
Sylwester Czmił	33,33 %	21.06.2023 Sylwester Czmił
Damian Mazur	33,33 %	21.06.2023 Damian Mazur

Rzeszów 21.06.2023.....

(miejsowość, data)

mgr inż. Sylwester Czmił

Wydział Elektrotechniki i Informatyki

Politechnika Rzeszowska im. Ignacego Łukasiewicza

## Oświadczenie współautorów publikacji

Jako współautor pracy pt. „*A method to detect type 1 diabetes based on physical activity measurements using a mobile device*” oświadczam, iż mój własny wkład merytoryczny w przygotowanie, przeprowadzenie i opracowanie badań oraz przedstawienie pracy w formie publikacji stanowi:

- współautorstwo koncepcji artykułu,
- współpracowanie metodologii i zadań badawczych,
- udział w przeprowadzeniu eksperymentów obliczeniowych,
- udział w opracowaniu i analizie wyników,
- udział w przygotowaniu rysunków i tabel,
- współredakcja pracy.

Mój udział procentowy w przygotowaniu publikacji określam jako 33,33 %.

Jednocześnie wyrażam zgodę na wykorzystanie w/w pracy jako część rozprawy doktorskiej mgr inż. Anny Czmił.

Sylwester Czmił  
.....  
(podpis oświadczającego)

Zzostawo 21.06.2023

(miejsowość, data)

dr hab. inż. Damian Mazur, prof. PRz  
Katedra Elektrotechniki i Podstaw Informatyki  
Politechnika Rzeszowska im. Ignacego Łukasiewicza

## Oświadczenie współautorów publikacji

Jako współautor pracy pt. „*A method to detect type 1 diabetes based on physical activity measurements using a mobile device*” oświadczam, iż mój własny wkład merytoryczny w przygotowanie, przeprowadzenie i opracowanie badań oraz przedstawienie pracy w formie publikacji stanowi:

- współautorstwo koncepcji artykułu,
- weryfikacja otrzymanych wyników,
- pozyskanie wsparcia finansowego,
- współredakcja pracy.

Mój udział procentowy w przygotowaniu publikacji określam jako 33,33 %.

Jednocześnie wyrażam zgodę na wykorzystanie w/w pracy jako część rozprawy doktorskiej mgr inż. Anny Czmił.

Damian Mazur



(podpis oświadczającego)

## Oświadczenie współautorów publikacji

Niniejszym podaję procentowy wkład autorski w publikację pt. „Automatic detection and counting of blood cells in smear images using RetinaNet”. Wkład Anny Czmił w powstanie publikacji obejmuje:

- współudział w implementacji oprogramowania służącego do identyfikacji i zliczania komórek na obrazach z rozmazów krwi,
- współpracowanie metodologii i przeprowadzenie części eksperymentów obliczeniowych,
- współpracowanie i analizę wyników,
- przygotowanie rysunków oraz tabel,
- współredakcję pracy.

Procentowy wkład autorski Anny Czmił wynosi: 33,33 %.

Imię i nazwisko współautora	Procentowy wkład autorski	Data i podpis współautora
Grzegorz Drałus	33,33 %	13.06.2023 
Damian Mazur	33,33 %	21.06.2023 

Rzeszów, 13.06.2023

(miejsowość, data)

dr inż. Grzegorz Drałus

Katedra Elektrotechniki i Podstaw Informatyki

Politechnika Rzeszowska im. Ignacego Łukasiewicza


## Oświadczenie współautorów publikacji

Jako współautor pracy pt. „Automatic detection and counting of blood cells in smear images using RetinaNet” oświadczam, iż mój własny wkład merytoryczny w przygotowanie, przeprowadzenie i opracowanie badań oraz przedstawienie pracy w formie publikacji stanowi:

- współautorstwo koncepcji artykułu,
- współudział w implementacji oprogramowania służącego do identyfikacji i zliczania komórek na obrazach z rozmazów krwi,
- współopracowanie metodologii i przeprowadzenie części eksperymentów obliczeniowych,
- współopracowanie i analizę wyników,
- współredakcję pracy.

Mój udział procentowy w przygotowaniu publikacji określam jako 33,33 %.

Jednocześnie wyrażam zgodę na wykorzystanie w/w pracy jako część rozprawy doktorskiej mgr inż. Anny Czmił.



(podpis oświadczającego)

Rzeszów 21.06.2023

(miejsowość, data)

dr hab. inż. Damian Mazur, prof. PRz  
Katedra Elektrotechniki i Podstaw Informatyki  
Politechnika Rzeszowska im. Ignacego Łukasiewicza

## Oświadczenie współautorów publikacji

Jako współautor pracy pt. „*Automatic detection and counting of blood cells in smear images using RetinaNet*” oświadczam, iż mój własny wkład merytoryczny w przygotowanie, przeprowadzenie i opracowanie badań oraz przedstawienie pracy w formie publikacji stanowi:

- współautorstwo koncepcji artykułu,
- weryfikacja otrzymanych wyników,
- pozyskanie wsparcia finansowego,
- współredakcja pracy.

Mój udział procentowy w przygotowaniu publikacji określam jako 33,33 %.

Jednocześnie wyrażam zgodę na wykorzystanie w/w pracy jako część rozprawy doktorskiej mgr inż. Anny Czmił.

Damian Mazur

(podpis oświadczającego)

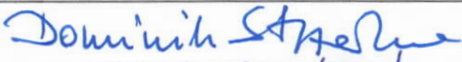
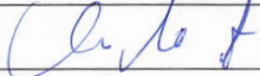
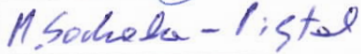
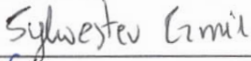



## Oświadczenie współautorów publikacji

Niniejszym podaję procentowy wkład autorski w publikację pt. „*NanoForms: an integrated server for processing, analysis and assembly of raw sequencing data of microbial genomes, from Oxford Nanopore technology*”. Wkład Anny Czmił w powstanie publikacji obejmuje:

- przygotowanie architektury aplikacji,
- implementację kluczowych funkcjonalności w aplikacji Nanoforms,
- przygotowanie widoków aplikacji,
- przygotowanie grafik,
- udział w analizie danych genomowych uzyskanych podczas sekwencjonowania i opracowaniu wyników,
- współredakcję pracy.

Procentowy wkład autorski Anny Czmił wynosi: 10 %.

Imię i nazwisko współautora	Procentowy wkład autorski	Data i podpis współautora
Dominik Strzałka	10 %	
Michał Piętał	10 %	
Marta Sochacka-Piętał	10 %	
Sylwester Czmił	10 %	
Michał Ćmił	10 %	
Michał Wroński	10 %	
Jan Gawor	10 %	
Tomasz Wołkowicz	10 %	
Dariusz Plewczynski	10 %	

Rzeszów M.DG. 23

(miejsowość, data)

dr hab. inż. Dominik Strzałka, prof. PRz  
Zakład Systemów Złożonych  
Politechnika Rzeszowska im. Ignacego Łukasiewicza

## Oświadczenie współautorów publikacji

Jako współautor pracy pt. „*NanoForms: an integrated server for processing, analysis and assembly of raw sequencing data of microbial genomes, from Oxford Nanopore technology*” oświadczam, iż mój własny wkład merytoryczny w przygotowanie, przeprowadzenie i opracowanie badań oraz przedstawienie pracy w formie publikacji stanowi:

- współpracowanie metodologii i eksperymentów,
- pozyskanie wsparcia finansowego,
- współredakcja pracy.

Mój udział procentowy w przygotowaniu publikacji określam jako 10 %.

Jednocześnie wyrażam zgodę na wykorzystanie w/w pracy jako część rozprawy doktorskiej mgr inż. Anny Czmił.

Dominik Strzałka

(podpis oświadczającego)



Rzeszów, 18 VI 2023 r.

(miejsowość, data)

dr Michał Piętał

Zakład Systemów Złożonych

Politechnika Rzeszowska im. Ignacego Łukasiewicza


## Oświadczenie współautorów publikacji

Jako współautor pracy pt. „*NanoForms: an integrated server for processing, analysis and assembly of raw sequencing data of microbial genomes, from Oxford Nanopore technology*” oświadczam, iż mój własny wkład merytoryczny w przygotowanie, przeprowadzenie i opracowanie badań oraz przedstawienie pracy w formie publikacji stanowi:

- zaproponowanie koncepcji artykułu,
- współpracowanie metodologii i eksperymentów,
- udział w implementacji kodu źródłowego aplikacji-serwera,
- pozyskanie wsparcia finansowego,
- współredakcja pracy.

Mój udział procentowy w przygotowaniu publikacji określam jako 10 %.

Jednocześnie wyrażam zgodę na wykorzystanie w/w pracy jako część rozprawy doktorskiej mgr inż. Anny Czmił.



(podpis oświadczającego)

Rzeszów, 19.06.2023.....

(miejsowość, data)

dr Marta Sochacka-Piętal  
Katedra Biotechnologii i Bioinformatyki  
Politechnika Rzeszowska im. Ignacego Łukasiewicza

## Oświadczenie współautorów publikacji

Jako współautor pracy pt. „*NanoForms: an integrated server for processing, analysis and assembly of raw sequencing data of microbial genomes, from Oxford Nanopore technology*” oświadczam, iż mój własny wkład merytoryczny w przygotowanie, przeprowadzenie i opracowanie badań oraz przedstawienie pracy w formie publikacji stanowi:

- współpracowanie metodologii i przeprowadzenie eksperymentów,
- udział w implementacji kodu źródłowego aplikacji-serwera,
- udział w analizie danych genomowych uzyskanych podczas sekwencjonowania i opracowaniu wyników,
- współredakcja pracy.

Mój udział procentowy w przygotowaniu publikacji określam jako 10 %.

Jednocześnie wyrażam zgodę na wykorzystanie w/w pracy jako część rozprawy doktorskiej mgr inż. Anny Czmił.

M. Sochacka-Piętal.....

(podpis oświadczającego)

Rzeszów 13.06.2023

(miejsowość, data)

mgr inż. Sylwester Czmił

Wydział Elektrotechniki i Informatyki

Politechnika Rzeszowska im. Ignacego Łukasiewicza

## Oświadczenie współautorów publikacji

Jako współautor pracy pt. „*NanoForms: an integrated server for processing, analysis and assembly of raw sequencing data of microbial genomes, from Oxford Nanopore technology*” oświadczam, iż mój własny wkład merytoryczny w przygotowanie, przeprowadzenie i opracowanie badań oraz przedstawienie pracy w formie publikacji stanowi:

- współpracowanie metodologii i eksperymentów,
- udział w implementacji kodu źródłowego aplikacji-serwera,
- przygotowanie rysunków i tabel,
- współredakcja pracy.

Mój udział procentowy w przygotowaniu publikacji określam jako 10 %.

Jednocześnie wyrażam zgodę na wykorzystanie w/w pracy jako część rozprawy doktorskiej mgr inż. Anny Czmił.

Sylwester Czmił

(podpis oświadczającego)

Rzeszów, 14 VI 2023r

(miejsowość, data)

mgr inż. Michał Ćmil  
Zakład Systemów Złożonych  
Politechnika Rzeszowska im. Ignacego Łukasiewicza

## Oświadczenie współautorów publikacji

Jako współautor pracy pt. „*NanoForms: an integrated server for processing, analysis and assembly of raw sequencing data of microbial genomes, from Oxford Nanopore technology*” oświadczam, iż mój własny wkład merytoryczny w przygotowanie, przeprowadzenie i opracowanie badań oraz przedstawienie pracy w formie publikacji stanowi:

- udział w analizie danych genomowych uzyskanych podczas sekwencjonowania i opracowaniu wyników,
- przygotowanie rysunków i tabel,
- współredakcja pracy,
- zapewnienie stałego wsparcia dla użytkowników aplikacji NanoForms.

Mój udział procentowy w przygotowaniu publikacji określam jako 10 %.

Jednocześnie wyrażam zgodę na wykorzystanie w/w pracy jako część rozprawy doktorskiej mgr inż. Anny Czmił.

Michał Ćmil

(podpis oświadczającego)

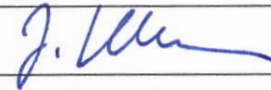
## Oświadczenie współautorów publikacji

Niniejszym podaję procentowy wkład autorski w publikację pt. „*GPR: A Python implementation of an extremely simple classifier based on fuzzy logic and gene expression programming*”.

Wkład Anny Czmił w powstanie publikacji obejmuje:

- współautorstwo koncepcji artykułu i metodologii,
- udział w implementacji algorytmu GPR w języku programowania Python,
- przeprowadzenie eksperymentów obliczeniowych,
- interpretację otrzymanych reguł klasyfikatora,
- walidację otrzymanych wyników,
- przygotowanie grafik,
- współredakcję pracy.

Procentowy wkład autorski Anny Czmił wynosi: 33,33 %.

Imię i nazwisko współautora	Procentowy wkład autorski	Data i podpis współautora
Jacek Kluska	33,33 %	21.06.2023 
Sylwester Czmił	33,33 %	21.06.2023 Sylwester Czmił

Rzeszów 21.06.2023

(miejsowość, data)

prof. dr hab. inż. Jacek Kluska  
Katedra Informatyki i Automatyki  
Politechnika Rzeszowska im. Ignacego Łukasiewicza

## Oświadczenie współautorów publikacji

Jako współautor pracy pt. „*GPR: A Python implementation of an extremely simple classifier based on fuzzy logic and gene expression programming*” oświadczam, iż mój własny wkład merytoryczny w przygotowanie, przeprowadzenie i opracowanie badań oraz przedstawienie pracy w formie publikacji stanowi:

- współautorstwo koncepcji artykułu i metodologii,
- nadzór nad planowaniem i realizacją poszczególnych zadań,
- udział w przygotowaniu rysunków i tabel,
- weryfikacja otrzymanych wyników,
- pozyskanie wsparcia finansowego,
- współredakcja pracy.

Mój udział procentowy w przygotowaniu publikacji określam jako 33,33 %.

Jednocześnie wyrażam zgodę na wykorzystanie w/w pracy jako część rozprawy doktorskiej mgr inż. Anny Czmił.



(podpis oświadczającego)



.....Rzeszów 21.06.2023.....

(miejsowość, data)

mgr inż. Sylwester Czmił

Wydział Elektrotechniki i Informatyki

Politechnika Rzeszowska im. Ignacego Łukasiewicza

## Oświadczenie współautorów publikacji

Jako współautor pracy pt. „*GPR: A Python implementation of an extremely simple classifier based on fuzzy logic and gene expression programming*” oświadczam, iż mój własny wkład merytoryczny w przygotowanie, przeprowadzenie i opracowanie badań oraz przedstawienie pracy w formie publikacji stanowi:

- współautorstwo koncepcji artykułu i metodologii,
- współpracowanie metodologii i zadań badawczych,
- udział w implementacji algorytmu GPR w języku programowania Python,
- współredakcja pracy.

Mój udział procentowy w przygotowaniu publikacji określam jako 33,33 %.

Jednocześnie wyrażam zgodę na wykorzystanie w/w pracy jako część rozprawy doktorskiej mgr inż. Anny Czmił.

.....Sylwester Czmił.....  
(podpis oświadczającego)