Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie

Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki

PRACA MAGISTERSKA

ŁUKASZ DZIEDZIA

PORÓWNANIE JAKOŚCI UOGÓLNIENIA I EFEKTYWNOŚCI DZIAŁANIA RÓŻNYCH METOD KLASYFIKACJI Z SIECIAMI NEURONOWYMI SONN

PROMOTOR:
dr Adrian Horzyk

Kraków 2009

OŚWIADCZENIE AUTORA PRACY

OŚWIADCZAM, ŚWIADOMY ODPOWIEDZIALNOŚCI KARNEJ ZA POŚWIADCZENIE NIEPRAWDY, ŻE NINIEJSZĄ PRACĘ DYPLOMOWĄ WYKONAŁEM OSOBIŚCIE I SAMODZIELNIE, I NIE KORZYSTAŁEM ZE ŹRÓDEŁ INNYCH NIŻ WYMIENIONE W PRACY.

. . . . . .

PODPIS

AGH
University of Science and Technology in Krakow

Faculty of Electrical Engineering, Automatics, Computer Science and Electronics

MASTER OF SCIENCE THESIS

ŁUKASZ DZIEDZIA

COMPARISON OF GENERALIZATION QUALITY AND PROCESSING EFFICIENCY OF VARIOUS CLASSIFICATION METHODS WITH SONN NEURAL NETWORKS

SUPERVISOR:
Adrian Horzyk Ph.D

Krakow 2009

...

Spis treści

1. Wstęp

Fascynacja zagadnieniem klasyfikacji danych nie jest zjawiskiem nowym. Ludzkość od stuleci próbuje klasyfikować rozmaite obiekty, dzięki czemu poznaje ich naturę i wykorzystuje zdobytą wiedzę do najróżniejszych celów. Wraz z rozwojem techniki komputerowej możliwości gromadzenia, przetwarzania i analizy danych nieustannie się polepszają, co stawia nowe wyzwania dla twórców oraz użytkowników algorytmów klasyfikacji. W związku z tym konieczne jest konstruowanie nowych oraz ulepszanie istniejących algorytmów tak, aby coraz skuteczniej radzić sobie z ogromną ilością danych i pozyskiwać cenne informacje w nich zawarte. Ponadto, wiele z istniejących algorytmów w swoim działaniu lub budowie przypomina niedościgniony wzór „naturalnego komputera" – ludzki mózg, co sprawia, że badania nad tymi metodami klasyfikacji są niezwykle interesujące.

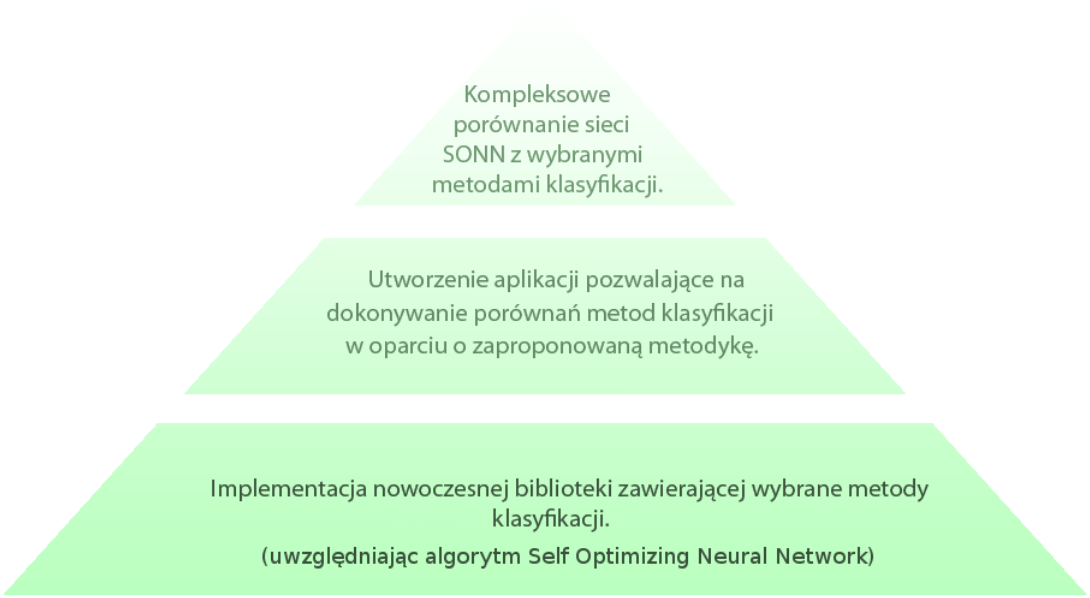
1.1. Motywacja

Osobistą motywacją do podjęcia tego tematu pracy było zainteresowanie metodami sztucznej inteligencji, dziedziną eksploracji danych oraz związanymi z tym zagadnieniami. Praca ta jest naturalną konsekwencją wybranego kierunku studiów oraz projektów i prac podejmowanych w trakcie edukacji. Dobór technologii oraz wybranych metod inżynierii oprogramowania, dzięki którym udało się zrealizować pracę, stanowi element podsumowania zdobytej wiedzy i doświadczeń. Realizując tą pracę intencją autora było zwiększenie oraz ugruntowanie wiadomości z podejmowanych dziedzin oraz dodanie skromnego wkładu do badań naukowych prowadzonych na Akademii Górniczo-Hutniczej w Krakowie.

1.2. Cele oraz zarys pracy

Tematem pracy jest dokonanie analizy porównawczej jakości uogólnienia oraz efektywności działania wybranych metod klasyfikacji z algorytmem *Self Optimizing Neural Networks* (SONN) autorstwa dr Adriana Horzyka, który od lat rozwijany jest w ramach badań naukowych na AGH w Krakowie. W niniejszej pracy analizowany jest algorytm SONN w wersji 3 (SONN-3) i ilekroć w pracy będzie mowa o algorytmie SONN to dotyczy to wersji SONN-3.

Wyniki uzyskane w pracy mają dostarczyć informacji o możliwościach i ograniczeniach algorytmu, które mogą posłużyć do rozwoju kolejnych wersji tego algorytmu.



Rysunek 1.1: Etapy pracy nad aplikacją, główne cele

Czynności niezbędne do wykonania analizy porównawczej wybranych metod klasyfikacji z sieciami SONN można opisać, jako proces składający się z trzech głównych etapów (patrz rysunek 1.1) opisanych w poniższych sekcjach.

1.2.1. Biblioteka algorytmów klasyfikacji

W celu uzyskania miarodajnych wyników porównań algorytmy muszą zostać zaimplementowane w tej samej technologii. Istnieje wiele porównań rozmaitych algorytmów, ale najczęściej przeprowadzane są z wykorzystaniem różnych aplikacji, które różnią się technologią wykonania, jakością, czy innymi szczegółami wpływającymi na powstawanie różnic w zakresie czasu tworzenia klasyfikatora. Ponadto zdarza się, że algorytmy w istniejących narzędziach wspierających eksplorację danych nie są dokładnie opisane, przez co trudno użytkownikom zweryfikować otrzymane rezultaty.

Istotnym elementem tej pracy jest etap implementacji spójnej biblioteki zawierającej wybrane algorytmy na wspólnej platformie technologicznej, przy jednoczesnym zadbaniu o udostępnienia informacji o parametrach i szczegółach implementacyjnych algorytmów. W ramach tego etapu konieczna jest analiza oraz implementacja wybranych algorytmów klasyfikacji danych. Jakość wykonania czynności w tym etapie jest kluczowa dla dalszych prac, stąd bardzo istotne jest zastosowanie odpowiednich narzędzi inżynierii oprogramowania opisanych w rozdziale 5. „Część praktyczna”.

1.2.2. Metodyka tworzenia porównań, aplikacja

Na tym etapie prac najważniejszym zadaniem jest analiza zagadnień związanych z procesem porównywania metod klasyfikacji. W oparciu o zebrane informacje konieczne jest zaproponowanie i opisanie metodyki dokonywania porównań pozwalającej na rzetelne skonfrontowanie algorytmu SONN z wybranymi metodami klasyfikacji. Na podstawie nakreślonej metodyki należy zaimplementować praktyczną, nowoczesną aplikację wspierającą proces porównania algorytmów. Aplikacja wspierająca porównywanie powinna wykorzystywać wcześniej zaimplementowaną bibliotekę algorytmów.

1.2.3. Porównanie wybranych metod klasyfikacji z algorytmem SONN

Ostatnim etapem realizowanym w ramach pracy jest wykorzystanie utworzonej aplikacji do wykonania porównań metod klasyfikacji, będących istotą pracy. Dokonując porównań należy odpowiednio dobrać zbiory danych, różne konfiguracje algorytmów oraz metod wstępnego przetwarzania danych. Praca powinna zostać zakończona przedstawieniem rezultatów oraz ich interpretacją, a także wskazaniem potencjalnych możliwości udoskonalenia algorytmu SONN.

1.3. Opis rozdziałów

Rozdział 1 „Wstęp” – wprowadzenie do tematyki pracy, przedstawienie celów oraz zakresu pracy.

Rozdział 2 „Eksploracja danych i zagadnienie klasyfikacji” – wprowadzenie teoretyczne do dziedziny eksploracji danych, a w szczególności zagadnienia klasyfikacji będącego przedmiotem pracy.

Rozdział 3 „Metody klasyfikacji” – opis metod klasyfikacji danych implementowanych oraz analizowanych w ramach pracy, z uwzględnieniem algorytmu *Self Optimizing Neural Networks*.

Rozdział 4 „Metodologia porównań algorytmów klasyfikacji” – opis zagadnień związanych z porównywaniem metod klasyfikacji. Opis procesu oraz kluczowych koncepcji wykorzystanych w pracy.

Rozdział 5 „Część praktyczna” – prezentacja aplikacji, techniczne aspekty zaimplementowanej biblioteki oraz aplikacji, opis stosowanych metod inżynierii oprogramowania.

Rozdział 6 „Wyniki porównań metod klasyfikacji” – prezentacja wyników przeprowadzonych porównań.

Rozdział 7 „Podsumowanie” – zebranie wniosków dotyczących porównań, analiza możliwości rozwoju systemu oraz algorytmu SONN.

Dodatek A – zawartość dołączonej płyty CD.

2. Eksploracja danych i zagadnienie klasyfikacji

Algorytmy klasyfikacji, w tym sieci SONN, stanowią istotny fragment dziedziny nazywanej *data mining*, która w związku z lawinowym przyrostem informacji staje się obszarem nauki coraz bardziej popularnym i pożądanym. W języku polskim występuje wiele tłumaczeń zwrotu *data mining*. W niniejszym rozdziale i całym opracowaniu używana będzie nazwa eksploracja danych.

W rozdziale „Eksploracja danych i zagadnienie klasyfikacji" zaprezentowana zostanie dziedzina eksploracji danych, omówione zostaną związane z nią metody oraz szczegółowo omówione zostanie zagadnienie klasyfikacji, będące przedmiotem niniejszej pracy.

2.1. Eksploracja danych

Eksploracja danych jest czynnością naturalną dla każdego człowieka. Ludzki mózg potrafi analizować i rozpoznawać nawet bardzo skomplikowane wzorce danych. Dla przykładu można wymienić zdolność do rozpoznawania ludzi po głosie, rysach twarzy czy innych cechach osobniczych. Zatem dlaczego powstała osobna dziedzina nauki poświęcona tak naturalnej dla człowieka czynności?

2.1.1. Geneza

Ludzkość generuje oraz przechowuje coraz większe ilości danych. Dane pochodzą z różnych źródeł, do których można zaliczyć chociażby centra badawcze, instytucje rządowe, firmy, czy popularny Internet. Aktualnie szacuje się, że co każde trzy lata ilość przechowywanych danych zostaje podwojona [1]. Bardzo prawdopodobne jest również to, że z czas ten będzie się skracał wraz z rozwojem technologii.

Ogromne ilości danych, których analiza wykracza poza możliwości człowieka stały się powodem do powstania dziedziny eksploracji danych, której zadaniem jest dostarczanie metod i narzędzi pomocnych w procesie analizy danych pod kątem wyszukiwania istotnych informacji w dużych zbiorach danych.

2.1.2. Obszary zastosowań

Eksploracja danych znajduje zastosowanie w wielu dziedzinach nauki stając się istotnym czynnikiem wpływającym na ich rozwój. Do obszarów zastosowań eksploracji danych należą:

- Medycyna – Zagadnienie eksploracji danych w medycynie jest chętnie wykorzystywane, tym bardziej, że wpływa na zwiększenie skuteczności leczenia, czy zapobiegania rozwoju groźnych chorób.

  Przykład: Zastosowanie niektórych metod pozwala na tworzenie systemów wspierających diagnostykę medyczną (jednym z przykładów zastosowania może być diagnoza płuc w celu wykrycia zmian nowotworowych).

- Biznes – jest to dziedzina, w której eksploracja danych znalazła wiele praktycznych zastosowań, a umiejętnie stosowana może doprowadzić do zwiększenia zysków oraz dynamiki rozwoju przedsiębiorstwa.

  Przykład: Bardzo przydatną cechą nowoczesnych systemów wspierających zarządzanie relacjami z klientami (*Customer Relationship Management*, CRM) jest możliwość dokonania analizy skupień grup klientów, czy dokonywania analizy koszykowej w celu usprawnienia kampanii marketingowej przedsiębiorstwa.

- Technika – podobnie jak w przypadku wyżej wymienionych obszarów zastosowań metody eksploracji danych znalazły swoje zastosowanie w wielu gałęziach techniki.

  Przykład: Diagnoza stanu skomplikowanej infrastruktury technicznej (np. sieci energetyczne) odbywa się za pomocą metod eksploracji danych.

- Biotechnologia – ze względu na duża ilość danych występujących w badaniach biotechnologicznych eksploracja danych jest bardzo skutecznym narzędziem wspierającym badania w zakresie tej dziedziny.

  Przykład: eksploracja danych w biotechnologii jest pomocna przy analizie zależności między różnymi strukturami DNA a prawdopodobieństwem wystąpienia raka.

- Inne – eksploracja danych jest lub może być wykorzystywana wszędzie tam, gdzie mamy do czynienia ze zbiorami danych, których rozmiary sprawiają, że niemożliwa staje się ich analiza przez człowieka.

2.2. Metody eksploracji danych

Rozróżnia się kilka metod występujących w wachlarzu narzędzi eksploracji danych. Najważniejsze z nich zostaną opisane poniżej.

2.2.1. Reguły asocjacyjne

Reguły asocjacyjne stosowane są w celu znalezienia zależności pomiędzy występowaniem wzorców w zbiorze danych. Częstym przykładem zastosowania reguł asocjacyjnych jest analiza koszykowa, dzięki której sprzedawca może wyznaczyć reguły występujące w procesie kupna jego produktów.

Przykład: Sprzedawca oferujący produkty A, B, C, D, E po zastosowaniu algorytmów wyszukujących reguły asocjacyjne na zbiorze archiwalnych transakcji może uzyskać informacje o zależnościach występujących przy kupnie produktów, np. kupując produkty B, C kupowany jest także produkt E. Takie algorytmy wykorzystywane są w wielu nowoczesnych serwisach zakupowych (np. amazon.com).

2.2.2. Regresja

Z regresją mamy do czynienia, gdy na podstawie informacji o próbkach ze zbioru danych (gdzie każda z próbek określona jest przez zestaw cech i związana jest z pewną, ciągłą wartością wynikową) chcemy znaleźć zależność pozwalającą na przewidzenie, jaka wartość wynikowa związana będzie z nową, nieznaną wcześniej próbką określoną przez taki sam zestaw cech.

Przykład: Regresja jest wykorzystywana do przewidywania szeregów czasowych (na przykład w analizie giełdowej).

2.2.3. Analiza skupień

Kolejną metodą, dzięki której wyszukać możemy cenne informacje w dużym zbiorze danych jest analiza skupień. Dzięki niej próbki danych opisane przez wiele cech możemy pogrupować zgodnie z pewnym, nieznanym wcześniej podobieństwem (wydzielić klastry).

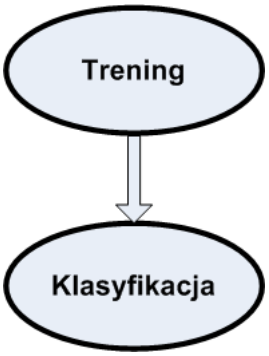
Przykład: Analiza skupień wykorzystywana jest między innymi przez przeglądarkę internetową Clusty.org [2] do semantycznego grupowania wyników wyszukiwań.

2.3. Klasyfikacja

Zagadnienie klasyfikacji możemy scharakteryzować jako próbę znalezienia mechanizmu, pozwalającego na przyporządkowanie próbki danych określonej przez pewien zestaw cech do jednej ze znanych klas. Taki mechanizm nosi nazwę klasyfikatora i jego działanie uzależnione jest od wybranej koncepcji, natomiast proces tworzenia klasyfikatora nazywa się najczęściej uczeniem lub treningiem. Matematyczne sformułowanie zagadnienia klasyfikacji znajduje się w podrozdziale 2.3.1, natomiast opis metod analizowanych w ramach pracy w rozdziale 3.

Warto zwrócić uwagę na podobieństwo klasyfikacji z zagadnieniem analizy skupień, dla obu podejść celem jest określenie przynależności nieznanego wzorca do pewnej grupy. Najistotniejszą różnicą pomiędzy tymi dwoma podejściami jest brak zdefiniowanego zbioru klas dla zagadnienia analizy skupień (*ang. unsupervised learning*), podczas gdy metody klasyfikacji wymagają sprecyzowania klasy dla każdego z występujących w problemie wzorców (*ang. supervised learning*).

Bardzo często w literaturze problem klasyfikacji definiuje się jako zagadnienie dwuetapowe, przedstawione schematycznie na rysunku 2.1.



Rysunek 2.1: Dwuetapowość procesu klasyfikacji

Najczęstszym sposobem wykorzystywania algorytmów klasyfikacji jest wytrenowanie najlepszego klasyfikatora dla danego problemu (zbioru danych), a następnie wielokrotne stosowanie go do klasyfikacji danych wcześniej nieznanych, na których klasyfikator nie był uczony, licząc na jego zdolność do uogólnienia zdobytej w trakcie treningu wiedzy.

Podobnie jak wcześniej opisywane metody, również klasyfikacja danych stosowana jest w wielu dziedzinach. Potwierdzeniem tego może być zestaw zbiorów danych wykorzystywanych przy porównaniach metod klasyfikacji w tej pracy, który został opisany w rozdziale 4.2. Nie jest kompletna lista możliwych obszarów zastosowań, a raczej jej skromny wycinek, który jednak powinien obrazować wszechstronność w zakresie możliwości wykorzystania zagadnienia klasyfikacji.

2.3.1. Matematyczne sformułowanie zagadnienia klasyfikacji

Jeżeli:

X - przestrzeń cech

Y - zbiór znanych klas

Dany jest n-elementowy zbiór treningowy:

$$\{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), ..., (\mathbf{x_n}, y_n)\}, \text{ gdzie:}$$

$\mathbf{x_i} \in X$ - wektor cech określający i-ty element zbioru danych (wzorzec)

$y_i \in Y$ - element ze zbioru dostępnych klas, mówiący do jakiej klasy i-ty obiekt należy
To zagadnienie klasyfikacji polega na znalezieniu takiego odwzorowania (klasyfikatora):

$$k : X \longrightarrow Y,$$

które dla dowolnego nieznanego wektora $\mathbf{x} \in X$ określi poprawnie, do jakiej klasy $y \in Y$ dany wektor należy.

3. Metody klasyfikacji

Spośród wielu popularnych metod klasyfikacji wybranych zostało kilka, które zostaną porównane z metodą SONN. Podczas wybierania algorytmów klasyfikacji do zestawu zaimplementowanego oraz stosowanego w porównaniach stosowano następujące kryteria:

- Popularność metody – istotne jest, aby dokonywana analiza było miarodajna oraz porównywalna. Osiągnięcie tego jest możliwe dzięki możliwości odniesienia wyników testów metody SONN do wyników algorytmów, które są rozpoznawalne i często występują w różnego rodzaju publikacjach i testach.

- Skuteczność metody – należy pamiętać, że algorytmy klasyfikacji stanowią narzędzia, które mają skutecznie wspierać inne dziedziny. Jest to istotny argument przemawiający za koniecznością tworzenia porównań takich, jak opisane w niniejszej pracy. Takie podejście wymusza skupienie uwagi na metodach osiągających dobre rezultaty, które mają szanse zostać zastosowane w praktycznych zagadnieniach.

- Własne zainteresowanie – spośród wielu istniejących algorytmów ostatecznie do implementacji wybrane zostały te, które okazały się szczególnie interesujące. Do takich metod niewątpliwie należą SVM i RBFN. W porównaniach nie znalazła się metoda perceptronu wielowarstwowego (*Multilayer Perceptron*, MLP), która była przedmiotem innego realizowanego przeze mnie projektu [3].

Ponieważ nie sposób było zaimplementować wszystkich istniejących metod, duży nacisk położony został na implementację biblioteki tworzonej w ramach pracy tak, aby możliwe było rozszerzanie istniejącego kodu o nowe algorytmy. Szczegóły tego rozwiązania omówione zostaną w rozdziale 5.

W dalszej części tego rozdziału opisane zostaną metody zaimplementowane w ramach pracy. Na szczególną uwagę zasługują informacje o szczegółach implementacyjnych, parametrach oraz różnych wariantach metod. Gdy będzie to możliwe zilustrowane zostaną przykłady działania algorytmu.

3.1. Metoda k najbliższych sąsiadów

Metoda k najbliższych sąsiadów (*k Nearest Neighbours*, kNN)[4] jest jedną z najprostszych metod klasyfikacji. Dzięki swojej prostocie wysoka interpretowalność[1] nadaje się doskonale jako narzędzie dydaktyczne podczas poznawania zagadnienia klasyfikacji. Również dzięki prostocie algorytm ten stał się bardzo popularny w dziedzinie eksploracji danych, co przemawia za sensownością porównywania innych metod z tym algorytmem.

3.1.1. Opis metody

Metoda k najbliższych sąsiadów bazuje na prostej zasadzie:

> „wzorzec danych należy do tej klasy, do których należy największa liczba spośród k sąsiadujących z nim wzorców".

Trening metody k najbliższych sąsiadów sprowadza się do zapamiętania zbioru uczącego (metoda *lazy*). Jest to dość specyficzne dlatego, że większość algorytmów klasyfikacji w fazie treningu wykonuje czasochłonne obliczenia. Konsekwencją tej specyfiki jest wydłużony czas klasyfikacji w porównaniu z innymi metodami.

Klasyfikacja konkretnego wzorca polega na wyznaczeniu k próbek ze zbioru treningowego znajdujących się najbliżej (w kontekście stosowanej metryki) badanego wzorca. Do najczęściej stosowanych metryk w algorytmie *k Nearest Neighbours* należą:

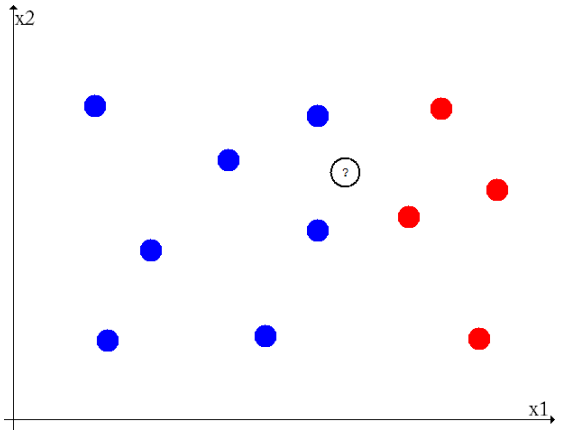
- Metryka euklidesowa
- Metryka Manhattan
- Metryka Hamminga

Ostatnim etapem jest określenie klasy, do jakiej należy analizowana próbka. W praktyce do najczęściej stosowanych metod wyboru klasy należą:

- *Majority Voting* (Głosowane na równych prawach) – próbka należy do tej klasy, do której należy najwięcej spośród k znalezionych sąsiadów.
- *Inverse Distance Voting* (Głosowanie z uwzględnieniem odległości) – dla każdej z klas występujących wśród znalezionych k sąsiadów obliczana jest suma odwróconych odległości od analizowanej próbki. Próbka danych zostaje zakwalifikowana do klasy, dla której obliczona suma jest największa.

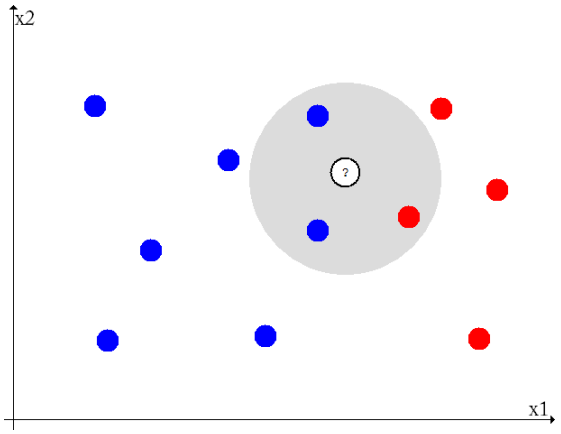
[1] Kryterium interpretowalności zostało zarysowane w rozdziale 4.5.4.

Działanie metody k najbliższych sąsiadów zostało zilustrowane dla przykładu dwuwymiarowego na rysunkach 3.1 oraz 3.2. Na pierwszym z nich widoczne są wzorce należące do dwóch różnych klas (kolory niebieski i czerwony) oraz klasyfikowany punkt (oznaczony znakiem zapytania), dla którego chcemy określić przynależność do jednej z klas.



Rysunek 3.1: Koncepcja działania metody kNN, problem

Na rysunku 3.2 zaznaczony został obszar zawierający k (w przykładzie $k = 3$) najbliższych sąsiadów.



Rysunek 3.2: Koncepcja działania metody kNN, rozwiązanie

Stosując metodę *Majority Voting* można stwierdzić, że dla $k = 3$ badany punkt będzie należał do klasy oznaczonej kolorem niebieskim.

3.1.2. Szczegóły implementacyjne oraz parametry metody

Zaimplementowany algorytm korzysta z metryki euklidesowej jako sposobu do wyznaczania odległości pomiędzy wektorami danych.

Zastosowana metoda wyboru klasy to bardzo popularna metoda *Majority Voting*.

Parametry związane z metodą k najbliższych sąsiadów przedstawione są w tabeli 3.1.

Tablica 3.1: Parametry metody kNN

Nazwa	Opis
k	Ilość wyszukiwanych najbliższych sąsiadów. Wraz ze wzrostem ilości sąsiadów zmniejsza się ryzyko zbyt zgrubnego oszacowania przynależności do klasy oraz zwiększa czas obliczeń. Należy jednak starannie dobierać wielkość tego parametru tak, aby zasięg poszukiwania sąsiedztwa nie był zbyt szeroki, co może doprowadzić do pogorszenia wyników klasyfikacji.

3.2. Support Vector Machines

Metoda *Support Vector Machines* (Maszyny Wektorów Wspierających, SVM)[5, 6, 7, 8, 9, 10, 11, 12] jest kolejną metodą klasyfikacji wybraną i zaimplementowaną w ramach niniejszej pracy. Jest to algorytm, który osiąga bardzo dobre wyniki w wielu testach, dlatego jest często stosowany w praktycznych zagadnieniach.

3.2.1. Opis metody

Istotą metody *Support Vector Machines* jest odnalezienie hiperpłaszczyzny, która podzieli przestrzeń cech na podprzestrzenie reprezentujące każdą z klas. Analizę tej metody warto rozpocząć od wyprowadzenia niezbędnych podstaw, na przykładzie najprostszego przypadku, a następnie rozszerzyć analizę o bardziej zaawansowane koncepcje, takie jak *Soft Margin*, *Kernel Trick*, czy metody osiągania wieloklasowości.

Rozważmy najprostszy przypadek. Mamy m-elementowy zbiór danych:

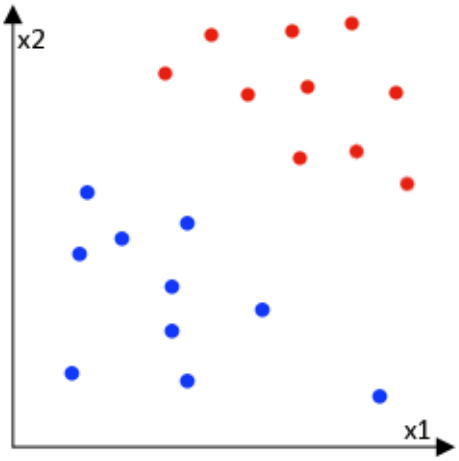
$$\{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), ..., (\mathbf{x_m}, y_m)\}, \text{ gdzie:}$$

$\mathbf{x_i}$ - wektor określający i-ty element zbioru

y_i - klasa i-tego elementu zbioru

Analizowany zbiór jest dwuklasowy (binarne zagadnienie klasyfikacji) oraz dane są liniowo separowane (założenie). Ilustracją takiego przykładu dla $n = 2$, gdzie n jest liczbą wymiarów przestrzeni cech może być rysunek 3.3.

Dla takiego przypadku istnieje nieskończenie wiele hiperpłaszczyzn, które potrafią podzielić przestrzeń cech na podprzestrzenie reprezentujące każdą z klas. Przykłady takich rozwiązań zostały zamieszczone na rysunku 3.4 jako proste w kolorze zielonym. Punkty znajdujące się na hiperpłaszczyźnie spełniają zależność:



Rysunek 3.3: Koncepcja działania metody SVM, liniowo separowalny (dwuklasowy) zbiór danych

$$\mathbf{wx_i} * b = 0, \text{ gdzie:}$$

w - normalna względem hiperpłaszczyzny

x - punkt w przestrzeni

Dzięki założeniu liniowej separowalności danych wszystkie punkty ze zbioru uczącego spełniają odpowiednio:

$$\mathbf{wx_i} + b \geq 1 \text{ , dla } y_i = 1 \tag{3.1}$$

$$\mathbf{wx_i} + b \leq 1 \text{ , dla } y_i = -1 \tag{3.2}$$

Co dla wszystkich punktów zbioru uczącego można zapisać jako:

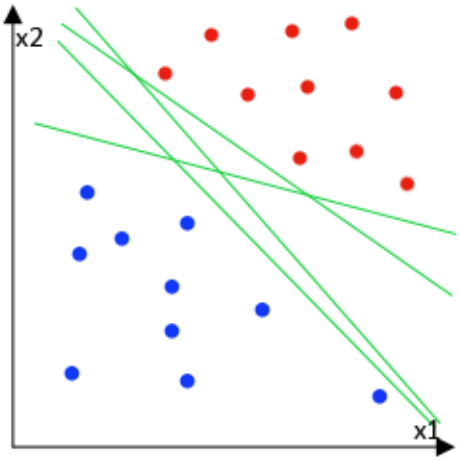
$$y_i(\mathbf{wx_i} + b) - 1 = 0 \tag{3.3}$$

Aby wybrać najlepsze rozwiązanie poszukuje się klasyfikatora, który podzieli przestrzeń z największym marginesem. Zastosowanie takiego rozwiązania motywowane jest założeniem, że wraz ze wzrostem szerokości marginesu zmniejsza się ryzyko popełnienia błędu w przypadku klasyfikacji nieznanych danych.

Można udowodnić [7], że maksymalizacja marginesu sprowadza się do minimalizacji funkcji celu:

$$Q(\mathbf{w}) = \mathbf{w}^T\mathbf{w} \text{ , przy ograniczeniach:} \tag{3.4}$$

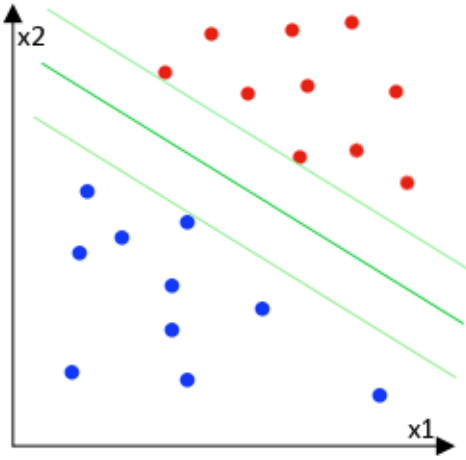
$$y_i(\mathbf{wx_i} + b) - 1 = 0 \tag{3.5}$$



Rysunek 3.4: Koncepcja działania metody SVM, przykłady możliwych rozwiązań

Tak postawiony problem można rozwiązać wieloma sposobami (metoda najszybszego spadku, metoda symulowanego wyżarzania). Bardzo często stosowanym sposobem jest zastosowanie metody programowania kwadratowego (*ang. Quadratic Programming*), które może być rozwiązane za pomocą algorytmu *Sequential Minimal Optimization* (SMO) [9, 10] opisanego w dalszej części tego rozdziału.

Znalezione zgodnie z tą metodą rozwiązanie przedstawion zostało na rysunku 3.5. Opisywany do tej pory przypadek nosi nazwę *Linear SVM* (LSVM).



Rysunek 3.5: Koncepcja działania metody SVM, klasyfikator liniowy z największym marginesem

Załóżmy, że dane nie są liniowo separowalne, co w praktyce jest bardzo częstym zjawiskiem. W takim przypadku zastosowanie metody maksymalnego marginesu przedstawione wcześniej nie jest uzasadnione. Istnieją dwa podejścia pozwalające na rozwiązanie problemu danych nieseparowalnych liniowo.

3.2.2. Soft Margin

Pierwszym podejściem stosowanym w przypadku danych nieseparowalnych liniowo jest metoda *Soft Margin*. Polega ona na dodaniu czynnika kary za odległość od hiperpłaszczyzny, dla każdego błędnie sklasyfikowanego wzorca. Zmodyfikowana funkcja celu ma postać:

$$Q(\mathbf{w}) = \mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{R}\varepsilon_i. \text{ , gdzie:} \tag{3.6}$$

C - współczynnik kary

R - ilość błędnie sklasyfikowanych próbek

ε_i - odległość i-tej błędnie sklasyfikowanej próbki od marginesu

3.2.3. Kernel Trick

Inne podejście pozwalające stosować metodę SVM dla zbiorów zawierających dane nieseparowalne liniowo nazywane jest *Kernel Trick* [11]. Podejście to polega na mapowaniu nie separowalnych liniowo próbek do przestrzeni o większej liczbie wymiarów, w której możliwe jest zastosowanie klasyfikatora liniowego (dane są separowalne liniowo). W tym celu stosuje się jedną z funkcji jądrowych (*ang. kernel function*). Do najczęściej używanych metod jądrowych w algorytmie SVM należą:

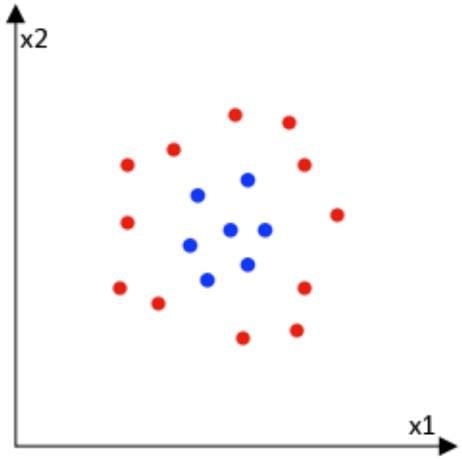
- funkcja gaussowska $K(\mathbf{u},\mathbf{v}) = e^{-(\frac{\|\mathbf{u}-\mathbf{v}\|^2}{2\sigma^2})}$
- funkcja wielomianowa $K(\mathbf{u},\mathbf{v}) = (\mathbf{uv}+1)^d$ lub $K(\mathbf{u},\mathbf{v}) = (\mathbf{uv})^d$
- funkcja sigmoidalna $K(\mathbf{u},\mathbf{v}) = tanh(\eta\mathbf{uv}+\rho)$

Na rysunkach 3.6 i 3.7 przedstawiono schematycznie zasadę działania *Kernel Trick* dla przypadku dwuwymiarowego. Rysunek 3.6 przedstawia zbiór danych nieseparowalnych liniowo, w którym jak w poprzednich przypadkach występują dwie klasy (zagadnienie binarne).

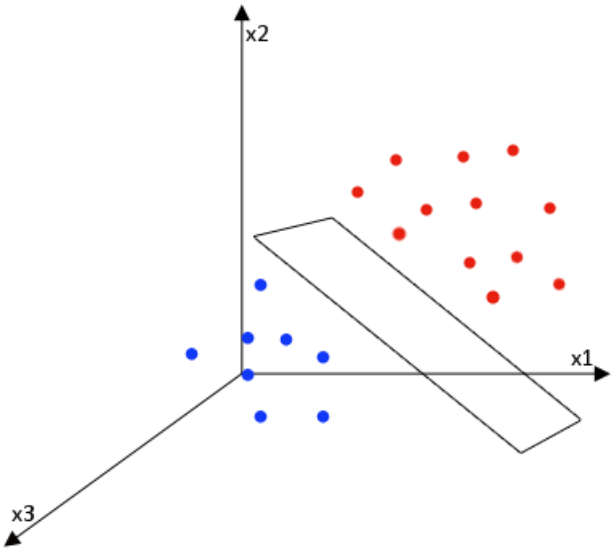
Na kolejnym rysunku 3.7 przedstawiono odpowiednio przetransformowane dane do przestrzeni o większej liczbie wymiarów oraz hiperpłaszczyznę wyznaczającą nowe obszary klasyfikacji.

3.2.4. Zastosowanie SVM dla zagadnień niebinarnych

Przedstawione powyżej rozważania dotyczą binarnego zagadnienia klasyfikacji. Najczęściej klasyfikowane dane należą do większej ilości klas, stąd istnieje konieczność adaptacji algorytmu SVM do rozwiązywania problemów wieloklasowych [12]. Rozwiązaniem tego problemu jest stworzenie wielu klasyfikatorów binarnych i odpowiednie ich wyspecjalizowanie. Przy założeniu, że k jest ilością klas występujących w problemie stosowane podejścia można opisać jako:



Rysunek 3.6: Koncepcja działania metody SVM, dane nieseparowalne liniowo



Rysunek 3.7: Koncepcja działania metody SVM, „Kernel Trick”

- *One Versus One* – dla każdej z istniejących par klas tworzony jest klasyfikator binarny potrafiący rozróżnić, do której z tych dwu klas analizowana próbka danych należy. Przynależność do klasy określana jest na podstawie największej ilości rozpoznań próbki jako reprezentanta danej klasy. Ilość utworzonych klasyfikatorów binarnych wynosi $\binom{k}{2}$.

- *One Versus All* – tworzonych jest k klasyfikatorów. Każdy z nich specjalizuje się w rozpoznawaniu przynależności do jednej z klas. Ostatecznie próbka zostaje sklasyfikowana do klasy, której klasyfikator wyznaczył największą wartość wyjściową (*winner takes all strategy*). Ilość utworzonych klasyfikatorów binarnych dla tego przypadku wynosi k.

3.2.5. Szczegóły implementacyjne oraz parametry metody

Algorytm SVM został zaimplementowany w oparciu o koncepcję Johna C. Platta [9, 10]. Koncepcja ta nosi nazwę *Sequential Minimal Optimization* (SMO). Zaproponowany przez Platta algorytm dotyczy rozwiązania zagadnienia programowania kwadratowego wynikającego z przedstawienia problemu SVM jako minimalizacji funkcji celu (patrz wzór 3.4). Koncepcja ta polega na zastąpieniu pierwotnego zagadnienia szeregiem mniejszych problemów programowania kwadratowego, które mogą zostać rozwiązane bez konieczności wykonywania skomplikowanych i czasochłonnych obliczeń numerycznych. Nie sposób w ramach niniejszej pracy analizować dokładnie tej ciekawej koncepcji, warto natomiast zwrócić uwagę na wyniki algorytmu SVM zaimplementowanego w wersji SMO z innymi podejściami [9, 10], co uzasadnia wybór tej metody.

Ze względu na szerokie zainteresowanie metodą SVM, w niniejszej pracy zostały zaimplementowane trzy odmiany tej metody:

1. Linear SVM – implementacja koncepcji LSVM opisanej we wcześniejszej części tego rozdziału.

2. RBF SVM – implementacja wykorzystująca funkcję gaussowską jako funkcję jądrową metody.

3. Polynomial SVM – implementacja wykorzystująca funkcję wielomianową jako funkcję jądrową metody.

Parametry stosowane w metodzie *Support Vector Machines* przedstawione zostały w tabeli 3.2.

Tablica 3.2: Parametry metody SVM

Nazwa	**Opis**
eps	Dla wszystkich metod. Parametr związany z dokładności obliczeń. Najczęściej jest to liczba rzędu 0.1 – 0.001.
cost	Dla wszystkich metod. Parametr wysokości kosztu za błędne sklasyfikowanie próbki. Jego wpływ wynika z zależności w opisanej funkcji celu(odnośnik).
sigma	Dla metody RBF SVM. Wartość odchylenia standardowego występująca w stosowanej funkcji Gaussa.
d	Dla metody Polynomial SVM. Parametr stopnia stosowanego wielomianu.

3.3. Radial Basis Function Networks

Istnieje wiele odmian sieci neuronowych, które są wykorzystywane w zagadnieniu klasyfikacji. W ramach pracy zaimplementowana została interesująca ich odmiana, a mianowicie *Radial Basis Function Networks* (Sieci Radialnych Funkcji Bazowych, RBFN) [13, 14, 15, 16]. *Radial Basis Function Networks* są sieciami bez sprzężeń zwrotnych (*ang. feedforward*). Ten rodzaj sieci składa się z trzech warstw:

- Warstwa wejściowa – rola neuronów w tej warstwie ogranicza się do przekazywania sygnałów wejściowych do warstwy ukrytej.
- Warstwa ukryta – neurony w tej warstwie wykorzystują radialną funkcję bazową do wygenerowania sygnału wyjściowego.
- Warstwa wyjściowa – sumuję wygenerowane w warstwie ukrytej sygnały wyjściowe (uwzględniając wagę każdego z połączeń), na tej podstawie obliczane są sygnały wyjściowe całej sieci.

Schemat budowy sieci typu RBF został przedstawiony na rysunku 3.8. Występujące na tym rysunku symbole oznaczają kolejno:

x_i – i-ty element wektora wejściowego danych

R_j – j-ty neuron warstwy ukrytej, wartość wyjściowa neuronu R_0 to zawsze 1

y_m – m-te wyjście sieci odpowiadające klasie m

n – wymiar przestrzeni cech

k – ilość neuronów warstwy ukrytej

c – ilość neuronów warstwy wyjściowej (ilość klas występujących w problemie klasyfikacji)

Radialna funkcja jest funkcją spełniająca zależność (przyjęta norma najczęściej oznacza normę euklidesową):

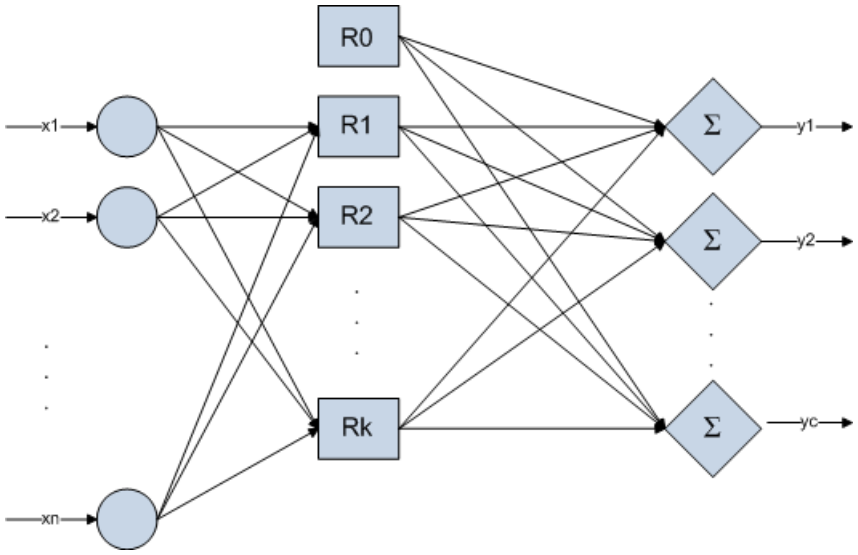
$$\phi(\mathbf{x}, \mathbf{c}) = \phi(\|\mathbf{x} - \mathbf{c}\|)$$

Najczęściej wykorzystywane funkcje radialne w sieciach RBF to:

- funkcja gaussowska $f(\mathbf{x}, \mathbf{c}) = e^{-(\beta\|\mathbf{x}-\mathbf{c}\|^2)}$
- funkcja wielokwadratowa (*ang. multiquadratic*) $f(\mathbf{x}, \mathbf{c}) = \sqrt{\|\mathbf{x} - \mathbf{c}\|^2 + \beta^2}$

Trening *Radial Basis Function Networks* podzielić można na dwa etapy:

- Trening warstwy ukrytej – wybór centrów dla funkcji radialnych oraz innych parametrów funkcji. Ta część treningu odbywa się bez nadzoru (*ang. unsupervised learning*).



Rysunek 3.8: Schemat architektury Radial Basis Function Network

- Trening warstwy wyjściowej – dobranie współczynników wag dla każdego z neuronów warstwy wyjściowej. Dobieranie współczynników jest nazywane nauką z nadzorem (*ang. supervised learning*).

Możliwość podziału treningu tak, aby warstwa ukryta i wyjściowa trenowane były oddzielnie jest istotne z punktu widzenia wydajności. Dzięki takiej konstrukcji sieci, w tym algorytmie nie występuje wymagający obliczeniowo proces wstecznej propagacji błędu (*ang. backpropagation*).

3.3.1. Trening warstwy ukrytej

Pierwszym z zadań związanych z uczeniem warstwy ukrytej sieci RBF jest znalezienie centrów funkcji radialnych. Istnieje kilka podejść do sposobu wyboru centrów neuronów warstwy ukrytej. Do najczęściej stosowanych należą:

- Wybór losowy
- Wybór za pomocą algorytmów analizy skupień
- Wybór za pomocą modeli probabilistycznych

Z zagadnieniem treningu warstwy ukrytej związany jest także dobór pozostałych parametrów wymaganych przez funkcje radialne. Dla obu wymienionych wcześniej funkcji radialnych konieczne jest określenie parametru dyspersji (β), który najczęściej jest wspólny dla każdego z neuronów warstwy ukrytej, a jego wartość to średnia odległość między kolejnymi centrami.

3.3.2. Trening warstwy wyjściowej

W kolejnym etapie treningu sieci RBF konieczne jest określenie współczynników wag dla każdego z połączeń warstwy ukrytej z warstwą wyjściową. Wagi muszą zostać dobrane tak, aby minimalizować wartość błędu będącego różnicą między oczekiwaną wartością wyjścia sieci a wartością obliczoną przez sieć dla każdego z neuronów warstwy wyjściowej. Tak postawiony problem można sprowadzić do minimalizacji funkcji błędu:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(\sum_{k=0}^{K}\mathbf{w_k}\phi_k^n - y^n)^2, \text{ gdzie} \tag{3.7}$$

N - ilość próbek w zbiorze treningowym

K - ilość neuronów warstwy ukrytej

$\mathbf{w_k}$ - waga powiązana z wartością wyjściową k-tego neuronu warstwy ukrytej

ϕ_k^n - wartość funkcji radialnej k-tego neuronu warstwy ukrytej dla n-tego wektora

y^n - oczekiwana wartość wyjściowa dla n-tego wektora zbioru treningowego

3.3.3. Szczegóły implementacyjne oraz parametry metody

Zaimplementowane w ramach pracy sieci RBF pozwalają na wykorzystanie funkcji Gaussa oraz funkcji wielokwadratowej, jako funkcji radialnych występujących w neuronach warstwy ukrytej. Centra dla funkcji radialnych mogą zostać wylosowane ze zbioru treningowego lub znalezione za pomocą algorytmów analizy skupień (*K-mean clustering*).

Parametry związane z metodą RBFN przedstawione zostały w tabeli 3.3.

Tablica 3.3: Parametry metody RBFN

Nazwa	**Opis**
k	Ilość neuronów w warstwie ukrytej. Wielkość ta powinna zostać dostosowana do ilości wzorców w zbiorze treningowym. Przy zbyt dużej ilości neuronów warstwy ukrytej istnieje ryzyko przeuczenia sieci (*ang. overfitting*).

3.4. Probabilistic Neural Networks

Metoda *Probabilistic Neural Networks* (Probabilistyczne Sieci Neuronowe, PNN) [17, 18, 19, 20] jest inspirowana bayesowskimi metodami klasyfikacji i stanowi ich rozwinięcie. Głównym założeniem tej metody jest próba aproksymacji funkcji gęstości prawdopodobieństwa rozkładu wektorów ze zbioru treningowego.

Schemat budowy *Probabilistic Neural Networks* przedstawiony został na rysunku 3.9. Występujące na rysunku symbole oznaczają kolejno:

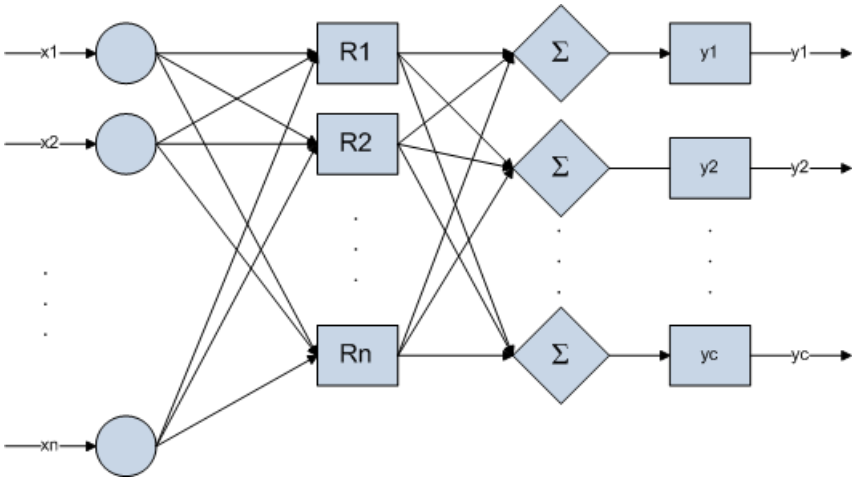
x_i – i-ty element wektora wejściowego danych

P_i – i-ty neuron warstwy ukrytej

y_m – m-te wyjście sieci odpowiadające klasie m

n – wymiar przestrzeni cech, ilość neuronów w warstwie ukrytej

c – ilość neuronów warstwy wyjściowej (ilość klas występujących w problemie klasyfikacji)



Rysunek 3.9: Schemat architektury Probabilistic Neural Network

Sieci PNN zbudowane są z następujących warstw:

- Warstwa wejściowa – składa się z neuronów, których liczba równa jest wymiarowi przestrzeni cech, jej zadaniem jest propagacja sygnałów wejściowych do każdego neuronu warstwy ukrytej.

- Warstwa ukryta – każdy z neuronów w tej warstwie skojarzony jest z jednym wektorem występującym w ciągu uczącym. Porównując działanie tej warstwy z warstwą ukrytą sieci RBF (patrz podrozdział 3.3.1), można powiedzieć, że każdy z neuronów odpowiada centrum zlokalizowanemu w punkcie określonym przez próbkę z ciągu uczącego, z którą jest skojarzony.

- Warstwa wyjściowa – zawiera dokładnie tyle neuronów, ile znanych jest klas w danym problemie klasyfikacji. Neurony w tej warstwie sumują wartości obliczone w warstwie ukrytej dla danej klasy. Prawdopodobieństwo przynależności do klasy reprezentowanej przez neuron warstwy wyjściowej może zostać obliczone po znormalizowaniu wartości wyjściowych do przedziału $(0, 1]$.

3.4.1. Szczegóły implementacyjne oraz parametry metody

Implementacja metody *Probabilistic Neural Networks* zastosowana w niniejszej pracy inspirowana jest opracowaniem tej metody autorstwa Nikolaya Y. Nikolaeva [17]. Zastosowana funkcja jądrowa to funkcja gaussowska.

Parametry stosowane w metodzie PNN przedstawione zostały w tabeli 3.4.

Tablica 3.4: Parametry metody PNN

Nazwa	**Opis**
sigma	Parametr odchylenia standardowego dla funkcji Gaussa stosowanej w neuronach warstwy ukrytej.

3.5. Self Optimizing Neural Networks

Spośród wielu metod klasyfikacji szczególną rolę odgrywają sieci neuronowe. Istnieje wiele rodzajów sieci stosowanych w zagadnieniu klasyfikacji, jak na przykład Perceptron Wielowarstwowy (MLP), czy omawiane w poprzednich rozdziałach Sieci Radialnych Funkcji Bazowych (RBFN) lub Sieci Probabilistyczne (PNN). Modele te charakteryzują się jednak pewnymi ograniczeniami, które często dyskwalifikują te metody w praktycznych zastosowaniach. Do takich ograniczeń należą:

- Potrzeba określenia topologii sieci – konieczność określania liczby warstw oraz liczby neuronów. Konieczność ustalenia architektury sieci a priori lub w sposób doświadczalny sprawia, że proces ten jest bardzo czasochłonny, a najczęściej prowadzi do nieoptymalnego doboru struktury, co wpływa na obniżenie zdolności generalizacyjnych modelu.
- „Przekleństwo wymiarowości" (*ang. curse of dimensionality*) – wraz ze wzrostem liczby wymiarów przestrzeni cech nieliniowo rośnie liczba neuronów (lub warstw) niezbędnych do stworzenia sieci realizującej poprawną klasyfikację danych.

Problemy te były jednym z powodów opracowania koncepcji ontogenicznych sieci neuronowych.

3.5.1. Ontogeniczne Sieci Neuronowe

Sieci ontogeniczne charakteryzują się zdolnością do automatycznej konstrukcji swej topologii (struktury). Sieci tego rodzaju możemy sklasyfikować ze względu na charakterystykę algorytmu konstrukcyjnego:

- Modele powiększające strukturę – to podejście polega na rozpoczęciu procesu uczenia od pewnej minimalnej struktury i stopniowym powiększaniu jej (dodawanie neuronów, warstw i połączeń synaptycznych) tak, aby polepszać zdolności generalizacji sieci.

- Modele zmniejszające strukturę – w tym podejściu proces uczenia rozpoczyna się ustalenia początkowej struktury, która dobrze realizuje zadanie generalizacji na określonym zbiorze (dalsze etapy uczenia nie wpływają na redukcję błędu). Po ustaleniu wstępnej struktury następuje proces redukcji sieci najczęściej poprzez usuwanie połączeń międzyneuronowych oraz neuronów, które mają nieistotny wpływ na realizowane zadanie klasyfikacji.

Analizowana w niniejszej pracy metoda *Self Optimizing Neural Networks* autorstwa dr Adriana Horzyka [21, 22, 23, 24, 25, 26] należy do grupy sieci ontogenicznych. Obok niej, jako przykłady innych podejść, można wymienić takie modele jak: RAN (*Resource Allocation Network*) [27], IncNet [28], upstart, czy algorytm korelacji kaskadowej (*ang. cascade-correlation network*).

3.5.2. Sieci SONN - opis

Celem autora metody SONN było zaprojektowanie uniwersalnego narzędzia klasyfikacji. Metoda ta realizuje postulat onotgoeniczności, czyli automatycznego konstruowania topologii oraz doboru wag połączeń synaptycznych. Model SONN można sklasyfikować jako sieć powiększającą strukturę (główny algorytm konstrukcyjny), przy czym warto zauważyć, że algorytm ten posiada także możliwość dokonywania rewizji struktury sieci i redukowaniu zbędnych neuronów.

Algorytm *Self Optimizing Neural Networks* rozwijany jest od kilku lat w ramach badań [21, 22, 23, 24, 25, 26] prowadzonych przez dr Adriana Horzyka na Akademii Górniczo-Hutniczej w Krakowie. W trakcie badań zaprezentowano kilka wersji algorytmu, których krótka historia zamieszczona jest poniżej.

- SONN-1 (rok 2001) – wersja ta ma początki w opracowywanej w pracy doktorskiej [26] dr Adriana Horzyka metodzie automatycznej konfiguracji sieci neuronowych dla problemów rozpoznawania wzorców binarnych. Do głównych możliwości sieci należy umiejętność automatycznej konstrukcji topologii sieci. Podstawowym ograniczeniem tego modelu było wymaganie stosowania wyłącznie wzorców binarnych w zbiorze treningowym.

- SONN-2 (rok 2005) – poprawa oraz optymalizacja algorytmu konstrukcyjnego (szacowana złożoność obliczeniowa: $O([\text{wymiar przestrzeni cech}] * [\text{ilość wektorów treningowych}]^2)$, zdolność do redukcji wymiaru przestrzeni cech

poprzez automatyczne wartościowanie najbardziej znaczących cech z punktu widzenia dyskryminacji wzorców.

- SONN-3 (rok 2008) – najnowsza wersja, która jest przedmiotem niniejszej pracy. Metoda SONN została wzbogacona o inteligentny sposób tworzenia warstwy wejściowej sieci (algorytm *Automatic Discriminative Lossy Binarization Conversion Algorithm*, ADLBCA), która potrafi klasyfikować dane o wartościach rzeczywistych, całkowitych oraz binarnych. Oprócz tej zmiany dokonano dalszej optymalizacji algorytmu konstrukcyjnego.

- SONN-4 – wersja, która aktualnie jest rozwijana. Planowane zmiany to umożliwienie konstruowania sieci w oparciu o dane symboliczne oraz umiejętność odnajdywania kombinacji dyskryminujących poszczególne klasy już na etapie tworzenia warstwy wejściowej.

Najistotniejsze cechy *Self Optimizing Neural Networks* przedstawione zostały w poniższym zestawieniu.

- Zdolność do automatycznego kreowania struktury sieci (ontogeniczność).

- Automatyczny dobór parametrów w procesie uczenia.

- Umiejętność automatycznej konwersji rzeczywistych danych wejściowych do postaci binarnych w warstwie wejściowej sieci (algorytm ADLBCA).

- Zdolność do klasyfikacji z zerowym błędem wzorców zastosowanych podczas konstruowania sieci (jeżeli wzorce zbioru treningowego nie są sprzeczne).

- Automatyczna redukcja wymiaru przestrzeni cech – podczas konstruowania sieci cechy, które nie mają istotnego znaczenia w procesie klasyfikacji zostają pominięte, co korzystnie wpływa na zdolność uogólniania oraz czas działania algorytmu.

- Dzięki dążeniu do zachowania jak najmniejszej struktury sieci, metoda SONN posiada dobre właściwości generalizacyjne, jest niewrażliwa na przeuczenie (*ang. overfitting*).

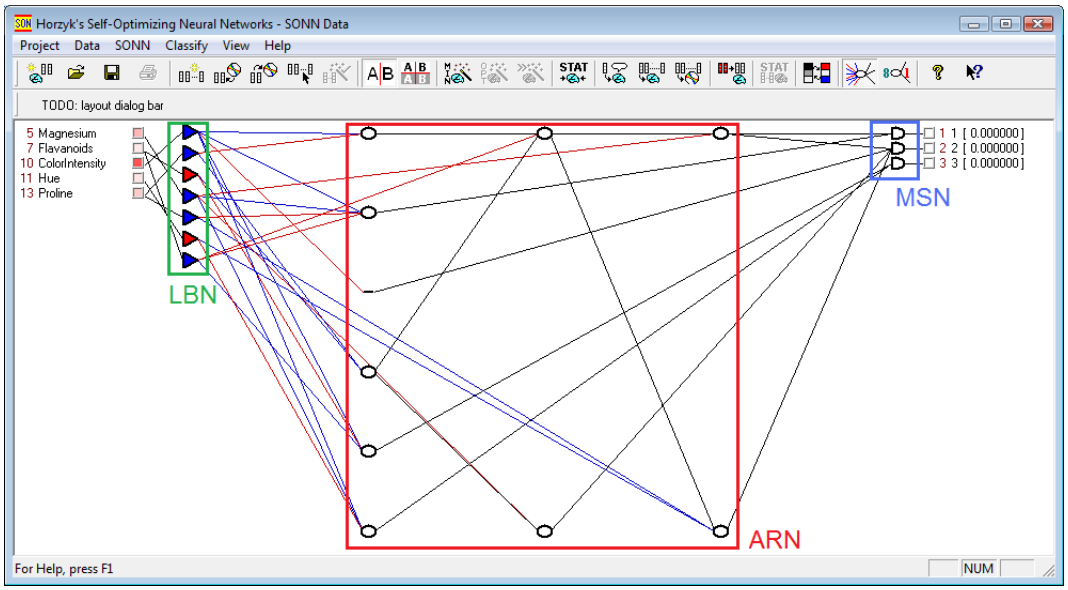
- Algorytm SONN jest niewrażliwy na rozkład klas w zbiorze treningowym.

- Umiejętność rozpoznawania negatywów wzorców danych.

Ciekawym aspektem działania sieci SONN jest stosowanie wartości binarnych jako pobudzeń generowanych przez neurony LBN sieci, co stanowi analogię do sposobu funkcjonowania ludzkiego mózgu.

3.5.3. Architektura

Sieci SONN są sieciami wielowarstwowymi. Ilość warstw i neuronów w każdej warstwie jest determinowana w procesie konstrukcyjnym[2]. Przykładową sieć SONN utworzoną dla zbioru danych Wine przedstawia rysunek 3.10. Na tym rysunku oznaczono miejsca występowania neuronów każdego z typów występujących w sieciach *Self Optimizing Neural Networks*.



Rysunek 3.10: Przykładowa struktura sieci SONN utworzona dla zbioru Wine (program autorstwa dr Adriana Horzyka)

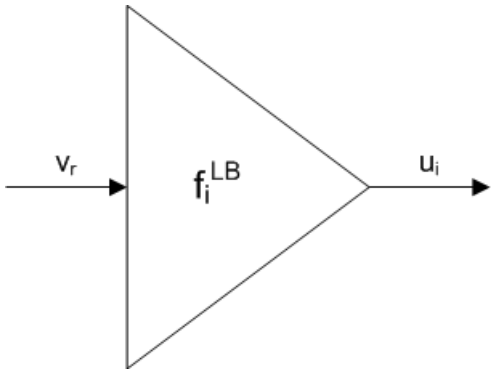
W sieciach SONN wyróżnić można trzy rodzaje neuronów wyspecjalizowanych w pełnieniu określonych funkcji, które występują w kolejnych warstwach sieci opisanych w poniższych sekcjach.

Warstwa wejściowa (Lossy Binarizing Subnetwork)

Warstwa wejściowa sieci SONN składa się wyłącznie z neuronów typu LBN (*Lossy Binarizating Neuron*). Ich zadaniem jest przekształcenie rzeczywistych danych wejściowych na binarne sygnały przesyłane do kolejnych warstw sieci tak, aby już w początkowym etapie tworzenia sieci wyróżnić istotne cechy. Na podstawie tych cech utworzony zostanie model sieci. Schemat neuronu LBN znajduje się na rysunku 3.11. Za tworzenie warstwy wejściowej sieci odpowiedzialny jest algorytmu binaryzacji (np. ADLBCA), którego działanie opisane zostanie w podrozdziale 3.5.4.

Każdy neuron LBN odpowiada za przekształcenie wektora danych rzeczywistych v_r na bipolarny sygnał $u_i = f_i^L B(v_r, R_i)$. Neuron taki związany jest z konkretną cechą k zbioru

[2]Proces konstrukcyjny w sieciach ontogenicznych jest odpowiednikiem procesu uczenia (treningu) dla innych algorytmów klasyfikacji.



Rysunek 3.11: Self Optimizing Neural Network, schemat LBN

uczącego oraz przedziałem wartości tej cechy, który jednoznacznie wskazuje na przynależność do jednej z klas problemu. Funkcja wyznaczająca wartość wyjścia neuronu LBN przedstawiona jest poniżej.

$$f_i^L B(v_r, R_i) = \begin{cases} +1 & \text{dla} \quad v_r^k \in R_i^k, \\ -1 & \text{w przeciwnym przypadku.} \end{cases} \text{, gdzie} \tag{3.8}$$

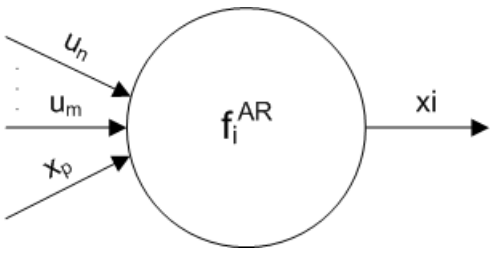
f_i^{LB} - wartość wyjścia i-tego neuronu LBN

v_r^k - wartość k-tej cechy wektora liczb rzeczywistych

R_i^k - przedział reprezentowany przez i-ty neuron LBN dla k-tej cechy

Warstwy ukryte (Aggregation Reinforcement Subnetworks)

Warstwy ukryte powstają w wyniku działania głównego algorytmu konstrukcyjnego sieci. W każdej z warstw ukrytych występują wyłącznie neurony typu ARN (*Aggregation Reinforcement Neuron*), których schemat zilustrowany został na rysunku 3.12. Celem warstw ukrytych jest odszukiwanie i wzmacnianie podobieństw między wzorcami występującymi w zbiorze treningowym.



Rysunek 3.12: Self Optimizing Neural Network, schemat ARN

Neurony ARN na wejściu mogą przyjmować sygnały wygenerowane przez neurony LBN w warstwie wejściowej $(u_m, \ldots, u_n)$ lub sygnały wyjściowe z innego (pojedynczego) neuronu ARN spośród neuronów występujących w warstwach wcześniejszych (x_p). Każde z wejść

związane jest z automatycznie obliczaną wartością wagi. Model matematyczny neuronu ARN przedstawiony zostały we wzorze 3.9.

$$x_i = f_i^{AR}(u_m, \ldots, u_n, x_p) = w_0^{x_p} x_p + \sum_{k \in \{i,\ldots,j\}} w_k^{u_k} u_k \tag{3.9}$$

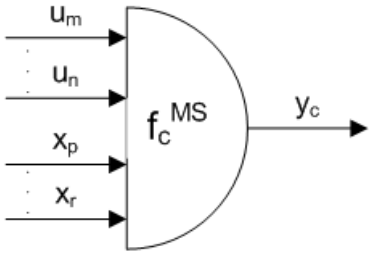
Liczba warstw oraz neuronów wzmacniających agregacje zależna jest od stopnia korelacji danych treningowych. Wyróżnić można dwie reguły określające zależność pomiędzy stopniem korelacji danych a strukturą sieci SONN:

1. Wraz ze wzrostem stopnia korelacji wzorców reprezentujących różne klasy w zbiorze treningowym, struktura sieci staje się coraz bardziej skomplikowana (rośnie liczna warstw i neuronów) i odwrotnie.

2. Wraz ze zmniejszaniem się stopnia korelacji wzorców reprezentujących te same klasy w zbiorze treningowym, struktura sieci staje się coraz bardziej skomplikowana i odwrotnie.

Warstwa wyjściowa (Maximum Selection Subnetwork)

Warstwa wyjściowa sieci SONN jest najprostszym strukturalnie oraz logicznie elementem modelu sieci. W skład tej warstwy wchodzą wyłącznie neurony typu MSN (*Maximum Selection Neuron*). Liczba neuronów zależy ściśle od liczby klas występujących dla danego zbioru danych (dla każdej z klas tworzony jest jeden neuron MSN). Zadaniem neuronów ostatniej warstwy jest wyznaczenie maksimum z otrzymywanych na wejściu sygnałów. Wejściowe sygnały mogą pochodzić bezpośrednio z warstwy wejściowej $(u_m, \ldots, u_n)$ lub z dowolnej liczby neuronów ARN $(x_p, \ldots, x_r)$. Warto zauważyć, że wejścia neuronów MSN nie są związane z żadną wagą. Schemat neuronu MSN przedstawione został na rysunku 3.13, a jego matematyczny opis we wzorze 3.10.



Rysunek 3.13: Self Optimizing Neural Network, schemat MSN

$$y_c = f_m^{MS}(u_m, \ldots, u_n, x_p, \ldots, x_r) = max(u_m, \ldots, u_n, x_p, \ldots, x_r) \tag{3.10}$$

3.5.4. Opis działania algorytmu

Działanie metody jest zdecydowanie bardziej wyszukane od wielu algorytmów opisywanych we wcześniejszych częściach pracy. Metody tej nie sposób opisać prostymi regułami, jest to raczej zbiór algorytmów wyspecjalizowanych w konkretnych zadaniach. Opis działania algorytmu zostanie przedstawiony zgodnie z porządkiem tworzenia kolejnych warstw sieci, począwszy od pojawieniu się na wejściu danych uczących.

Tworzenie warstwy wejściowej – Lossy Binarizing Subnetwork

Za tworzenie warstwy wejściowej sieci SONN odpowiedzialny jest algorytm *Automatic Discriminative Lossy Binarization Conversion Algorithm* (ADLBCA) [25]. Celem tego algorytmu jest utworzenie grupy neuronów odpowiedzialnych za przekształcenie rzeczywistych danych wejściowych do postaci binarnych (bipolarnych) sygnałów. Dokonuje się tego poprzez wyszukanie przedziałów na zbiorze danych dla każdej z cech, które najlepiej determinują przynależność do konkretnej klasy występującej w danym problemie klasyfikacji. Dąży się do minimalizacji ilości neuronów w tej warstwie poprzez wyszukiwanie jak największych przedziałów (algorytm zachłanny). Warto zauważyć, że już na tym etapie dokonuje się selekcja cech, które mają największy wpływ na określenie przynależności do klas.

Dla każdego ze znalezionych przedziałów tworzony jest neuron LBN (patrz podrozdział 3.5.3), który na swym wyjściu da sygnał $+1$, jeżeli podany wzorzec ma wartość cechy w przedziale, który dany neuron reprezentuje lub -1 w przeciwnym przypadku.

Tworzenie warstw ukrytych – Aggregation Reinforcement Subnetwork

Po utworzeniu neuronów LBN następuje główny algorytm konstrukcyjny modelu SONN. Omawianie tego kroku należy rozpocząć od wyjaśnienia pojęcia *Global Discrimination Coefficient* (globalny współczynnik rozróżnienia, GDC).

Globalny współczynnik rozróżnienia wyznaczany jest dla każdego neuronu LBN (każdej zbinaryzowanej cechy używanej przez sieć SONN). Celem wprowadzania tego współczynnika jest konieczność dokładnego określenia zdolności dyskryminacyjnych neuronów doprowadzających sygnały do warstw ukrytych, dzięki temu możliwe jest przeprowadzenie algorytmu konstrukcyjnego i optymalny dobór wag połączeń synaptycznych. W celu wyznaczenia współczynników GDC konieczna jest analiza całego zbioru treningowego. Skutkuje to koniecznością przebudowy sieci po zmianie zbioru uczącego, co generalnie jest cechą charakterystyczną większości klasyfikatorów.

$$d^n_{(k+)} = \begin{cases} \frac{P^m_k}{(M-1)Q^m} \sum\limits_{h=1 \wedge h \neq m}^{M} (1 - \frac{P^h_k}{Q^h}) \text{ gdy } u^n_k \geq 0 \wedge u^n \in C^m, \\ 0 \text{ gdy } u^n_k < 0 \wedge u^n \in C^m. \end{cases} \tag{3.11}$$

$$d^n_{(k-)} = \begin{cases} \frac{N^m_k}{(M-1)Q^m} \sum_{h=1 \wedge h \neq m}^{M} (1 - \frac{N^h_k}{Q^h}) \text{ gdy } u^n_k \leq 0 \wedge u^n \in C^m, \\ 0 \text{ gdy } u^n_k > 0 \wedge u^n \in C^m. \end{cases} \tag{3.12}$$

$$P^m_k = \sum_{u^n_k \in \{u \in U \vee C^m : u^n_k > 0, n \in \{1,\ldots,Q\}\}} u^n_k \tag{3.13}$$

$$N^m_k = \sum_{u^n_k \in \{u \in U \vee C^m : u^n_k < 0, n \in \{1,\ldots,Q\}\}} -u^n_k \tag{3.14}$$

$$Q^m = \| \; u \in U \cap C^m : n \in \{1, \ldots, Q\} \; \| \tag{3.15}$$

Symbole w powyższych wzorach oznaczają:

M - liczba klas dla konkretneog problemu klasyfikacji

Q - liczba wzorców w zbiorze treningowym

C^m - identyfikato m-tej klasy

u^n_k - wartość binarnego wyjścia n-tego neuronu związaengo z k-tą cechą

Po wyznaczeniu globalnych współczynników dyskryminacji konieczne jest określenie, które z nich posłużą w procesie konstrukcyjnym tak, aby wszystkie wzorce treningowe mogły zostać sklasyfikowane poprawnie. Takie współczynniki noszą nazwę obligatoryjnych (*Obligatory*) i uwzględnienie ich jest konieczne. Rozróżnia się jeszcze współczynniki opcjonalne, których uwzględnienie może nie wpłynąć na rozmiar sieci. Algorytm wyznaczania obligatoryjnych i opcjonalnych współczynników dyskryminacji przebiega w następujących krokach:

1. Dla każdego ze współczynników, które nie zostały uznane za obligatoryjne, należy oszacować ilość nierozróżnionych wcześniej wzorców, które mogą zostać rozróżnione przez ten współczynnik. Obliczenie iloczynu wartości GDC i szacowanej ilości wzorców.

2. Wybór największego z iloczynów obliczonych w poprzednim kroku.

3. Oznaczenie wzorców, które mogą zostać rozróżnione przez współczynnik wybrany w kroku 2.

4. Jeżeli pozostały niedyskryminowane wzorce, to powrót do kroku 2. W przeciwnym wypadku - koniec.

Zastosowanie powyższego algorytmu gwarantuje zdolność sieci SONN do bezbłędnej klasyfikacji wzorców uczących. Z drugiej strony, poszukiwanie cech o największych zdolnościach określających przynależność do klas korzystnie wpływa na właściwości generalizacji w ten sposób zbudowanego modelu.

Neurony ARN w podsieci ARS są łączone z dowolną liczbą neuronów LBN oraz pojedynczym - jeżeli taki istnieje - neuronem ARN z poprzedniej warstwy. Celem tworzenia

połączeń jest wzmacnianie (*ang. reinforcement*) zdolności do określania przynależności do konkretnej klasy.

Tworzenie warstwy wyjściowej – Maximum Selection Subnetwork

Liczba neuronów warstwy wyjściowej bezpośrednio zależy od ilości klas występujących w danym problemie klasyfikacji. Neurony w warstwie wyjściowej mają za zadanie wybrać maksymalną wartość uzyskaną od połączonych z nimi neuronami warstw wcześniejszych. Wartość wyjściowa każdego z neuronów MSN może być interpretowana jako prawdopodobieństwo, z jakim sieć zakwalifikowała klasyfikowany wzorzec do klasy związanej z danym neuronem MSN. Warto zwrócić uwagę, że wejścia neuronów warstwy wejściowej nie są powiązane z żadną wagą, wszystkie niezbędne obliczenia dokonują się w warstwach ukrytych i warstwie wejściowej. Najwyższa wygenerowana wartość wyjściowa spośród neuronów MSN jest traktowana jako wartość wyjściowa sieci SONN, co obrazuje równanie 3.10.

3.5.5. Szczegóły implementacyjne

Implementacja algorytmu SONN, która jest elementem biblioteki algorytmów, bazuje na doświadczalnym kodzie algorytmu stworzonym przez dr Adriana Horzyka. Kod został przepisany na platformę .NET (C#) i zawiera kilka usprawnień w stosunku do wersji oryginalnej.

Metoda SONN nie jest parametryzowana.

4. Metodologia porównań metod klasyfikacji

W niniejszym rozdziale omówiona zostanie metodologia dokonywania porównań algorytmów klasyfikacji. Przedstawiony zostanie proces badawczy prowadzący do otrzymania wyników oraz zostaną omówione zagadnienia związane z tym procesem. Zaprezentowane zostaną wyselekcjonowane zbiory danych, dla których sprawdzone zostanie działanie klasyfikatorów. Szczególny nacisk położony zostanie na kryteria porównawcze, wskaźniki stosowane do porównań oraz mechanizmy walidacji. Zostaną także omówione sposoby wstępnego przetwarzania danych, stosowane nie tylko przy porównywaniu algorytmów, ale też w innych dziedzinach.

Motywacją do napisania tego rozdziału w obecnym kształcie był fakt istnienia znacznej ilości publikacji poruszających zagadnienie klasyfikacji i oferujących porównanie poszczególnych metod, przy braku opisu metodologii zastosowanej do przeprowadzenia tych porównań.

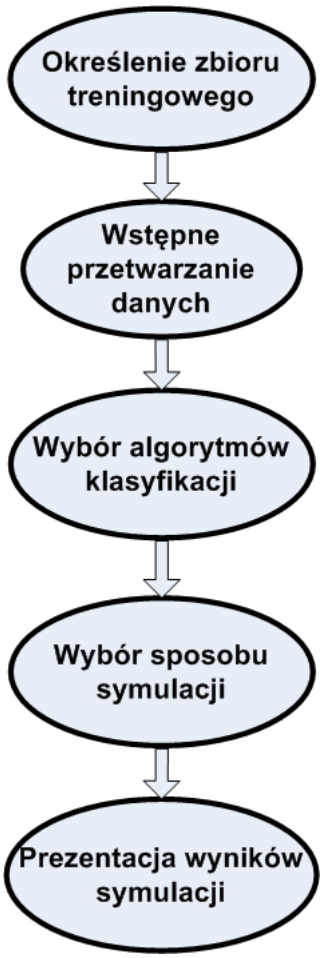
Wreszcie rozdział ten może stanowić podsumowanie istniejącego i stosowanego w praktyce zestawu koncepcji i narzędzi związanego ze stosowaniem i porównywaniem metod klasyfikacji.

4.1. Proces porównywania

Dokonując porównań metod klasyfikacji użytkownicy muszą wykonać szereg czynności, które wymagane są do uzyskania finalnych wyników. Charakterystyka algorytmów klasyfikacji sprawiła, że opisany poniżej proces (patrz rysunek 4.1) jest standardowy i występuje w takim lub podobnym kształcie w wielu aplikacjach związanych z eksploracją danych i metodami klasyfikacji [29, 30]. Kolejne etapy tego procesu to:

1. Określenie zbioru treningowego – celem dokonywania porównań jest wybranie najlepszego klasyfikatora, który zgodnie z przyjętymi kryteriami jakości, pozwala na realizację zagadnienia klasyfikacji dla konkretnego problemu. Przez problem rozumiemy zestaw danych (zbiór danych) podlegających klasyfikacji. Zbiory danych, dla których przeprowadzone zostaną badania opisane będą w podrozdziale 4.2.

2. Wstępne przetwarzanie danych – wybrany zbiór danych może pochodzić z różnych źródeł. Często przed przystąpieniem do treningu klasyfikatora przeprowadzane są czyn-



Rysunek 4.1: Schemat procesu porównywania metod klasyfikacji

ności pozwalające na transformację zbioru danych do innej postaci, co często pozwala na uzyskanie lepszych wyników klasyfikacji. Metody przetwarzania danych zaimplementowane w ramach pracy zostaną zaprezentowane w podrozdziale 4.3.

3. Wybór algorytmów klasyfikacji – w tym etapie wybierane są algorytmy, które mają zostać sprawdzone na wybranym i przetworzonym wcześniej zbiorze danych. Istotna jest możliwość sprawdzenia konkretnego algorytmu w różnych wariantach, jeżeli takie występują oraz przy zastosowaniu różnej konfiguracji parametrów występujących w algorytmie. Algorytmy klasyfikacji, które znalazły się w bibliotece zaimplementowanej w ramach pracy opisane zostały w rozdziale 3.

4. Wybór sposobu symulacji – z porównywaniem metod klasyfikacji związane jest zagadnienie walidacji, dzięki któremu określić można, jakie wyniki osiągane są na znanym zbiorze treningowym, a jakich można spodziewać się po wprowadzeniu danych nieznanych podczas uczenia. Stosowane sposoby walidacji opisane zostały w podrozdziale 4.4.

5. Prezentacja wyników – ostatnim krokiem omawianego procesu jest prezentacja wyników otrzymanych po zakończeniu symulacji. Poprzez wyniki rozumiane są statystyki algorytmów oraz kryteria jakości, które opisane zostały w podrozdziale 4.5.

4.2. Zbiory danych

Kluczowym elementem przy dokonywaniu ogólnego porównania metod klasyfikacji jest wybór zbiorów danych. Wybierając zbiory danych do niniejszej pracy brane były pod uwagę następujące kryteria:

1. Rozpoznawalność – istnieje wiele publikacji przedstawiających porównanie metod klasyfikacji. Bardzo często takie porównania przeprowadza się na tych samych zbiorach, co czyni je rozpoznawalnymi i dzięki temu możliwa jest konfrontacja konkretnego algorytmu z innymi, opisanymi w innych pracach. Do takich często stosowanych zbiorów niewątpliwie należą Wine i Iris, które wykorzystane zostaną również w niniejszej pracy.

2. Zróżnicowanie – w celu uzyskania rzetelnych wyników koniecznym jest zróżnicowanie zestawu stosowanych zbiorów. W związku z tym wybrane zostały zbiory charakteryzujące się:

   (a) Różną wielkością – wielkość zbioru wpływa na czas treningu i klasyfikacji. Wśród stosowanych zbiorów znajdują się zarówno zbiory małe (Iris, wielkość wolumenu = 600), jak i zbiory stosunkowo duże (Arrhythmia, wielkość wolumenu = 126108).

   (b) Różną liczbą cech – liczba cech dla wielu algorytmów jest istotnym parametrem wpływającym na strukturę klasyfikatora, dlatego wybrane zostały zbiory opisane przez kilka (Iris, liczba cech = 4) oraz kilkaset cech (Arrhythmia, liczba cech = 279).

   (c) Różnym rozkładem ilości reprezentantów każdej z klas – wśród stosowanych zbiorów znajdują się takie, które zawierają równomierny podział reprezentantów każdej z klas (Iris), jak i te, dla których rozkład ten jest nierównomierny (np. Yeast).

   (d) Rodzajem danych – wśród wybranych zbiorów większość opisana jest przez dane rzeczywiste, ale znajdują się też dane w formacie binarnym bipolarnym[1] (Congressional Voting Record).

   (e) Dziedziną pochodzenia – w celu zaprezentowania obszarów wykorzystywania metod klasyfikacji wybrane zostały zbiory związane z różnymi dziedzinami. Wśród

[1] W rzeczywistości są to dane trójwartościowe, w których występuję także wartość nieokreślona (1 - głos na tak, -1 – głos na nie oraz 0 - wstrzymanie się od głosu)

tych dziedzin znajduje się medycyna (Arrhythmia), biochemia (Wine, Yeast), botanika (Iris), technika (Ionosphere), a nawet polityka (Congressional Voting Records).

3. Powszechność – realizowane porównania algorytmów mogą być zestawiane z podobnymi próbami. Aby było to możliwe konieczne jest stosowanie publicznie dostępnych zestawów danych. Zbiory danych stosowane w niniejszej pracy pochodzą z popularnego i ogólnodostępnego repozytorium danych [31].

Wybrane zbiory danych zostały zaprezentowane wraz z informacjami interesującymi z punktu widzenia klasyfikacji (liczba cech, liczba wektorów). Do opisu dodany został też wskaźnik „wielkość wolumenu" obrazujący wielkość zbioru danych (iloczyn liczby cech i liczby wektorów).

Tablica 4.1: Charakterystyka zbioru danych Iris

Nazwa	Iris
Liczba cech	4
Liczba wektorów treningowych	150
Wielkość wolumenu	600
Rozkład danych wg. klas	Iris-Setos = 50, Iris-Virginica = 50, Iris-Versicolor = 50
Źródło	`http://archive.ics.uci.edu/ml/datasets/Iris`
Opis	Zbiór ten zawiera informacje o trzech gatunkach kwiatów z rodzaju kosaćców (irysów). Popularny zbiór danych występujący w wielu publikacjach dotyczących zagadnienia klasyfikacji. Cechuje się niewielką liczbą danych i równomiernym rozkładem klas.

Tablica 4.2: Charakterystyka zbioru danych Wine

Nazwa	Wine
Liczba cech	13
Liczba wektorów treningowych	178
Wielkość wolumenu	2314
Rozkład danych wg. klas	1 = 59, 2 = 71, 3 = 48
Źródło	`http://archive.ics.uci.edu/ml/datasets/Wine`
Opis	Zbiór danych zawierający wybranych 13 parametrów zebranych na podstawie analizy chemicznej trzech 3 rodzajów win produkowanych w pewnym regionie Włoch. Jest to popularny, często występujący w publikacjach zbiór danych. Jest uznawany za nieskomplikowany.

Tablica 4.3: Charakterystyka zbioru danych Yeast

Nazwa	Yeast
Liczba cech	8
Liczba wektorów treningowych	1484
Wielkość wolumenu	11872
Rozkład danych wg. klas	CYT = 463, NUC = 429, MIT = 244, ME3 = 163, ME2 = 51, ME1 = 44, EXC = 37, VAC = 30, POX = 20, ERL = 5
Źródło	`http://archive.ics.uci.edu/ml/datasets/Yeast`
Opis	Zbiór zawiera informacje o lokalizacji protein w analizowanych komórkach drożdży. Zbiór ten został wybrany ze względu na nierównomierny rozkład klas w zbiorze danych.

Tablica 4.4: Charakterystyka zbioru danych Congressional Voting Records

Nazwa	Congressional Voting Records
Liczba cech	16
Liczba wektorów treningowych	435
Wielkość wolumenu	6960
Rozkład danych wg. klas	democrate = 267, republican = 168
Źródło	`http://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records`
Opis	Zbiór danych zawierający wyniki 16 głosowań kongresu Stanów Zjednoczonych Ameryki Północnej z roku 1984. Zbiór tez został wybrany, ponieważ składa wyłącznie z danych o wartościach: +1, 0, -1.

Tablica 4.5: Charakterystyka zbioru danych Car Evaluation

Nazwa	Car Evaluation
Liczba cech	6
Liczba wektorów treningowych	1728
Wielkość wolumenu	10368
Rozkład danych wg. klas	unacc = 1210, acc = 384, good = 69, vgood = 65
Źródło	`http://archive.ics.uci.edu/ml/datasets/Car+Evaluation`
Opis	Zbiór zawierający dane charakteryzujące przykładowe samochody. Klasa w tym zbiorze określa jeden z czterech stopni akceptowalności danego modelu.
Uwagi	Tekstowe dane opisowe zostały zamienione na postać liczbową.

Tablica 4.6: Charakterystyka zbioru danych Ionoshpere

Nazwa	Ionoshpere
Liczba cech	34
Liczba wektorów treningowych	351
Wielkość wolumenu	11934
Rozkład danych wg. klas	g = 225, b = 126
Źródło	`http://archive.ics.uci.edu/ml/datasets/Ionosphere`
Opis	Zbiór danych zawierający wyniki testowych pomiarów jakości działania radarów. Każdy z pomiarów może zostać sklasyfikowany do jednej z dwóch klas (good, bad). Zbiór ten jest przydatny do testowania szczególnego przypadku – binarnego zagadnienia klasyfikacji.

Tablica 4.7: Charakterystyka zbioru danych Arrhythmia

Nazwa	Arrhythmia
Liczba cech	279
Liczba wektorów treningowych	452
Wielkość wolumenu	126108
Rozkład danych wg. klas	01 = 245, 02 = 44, 03 = 15, 04 = 15, 05 = 13, 06 = 25, 07 = 3, 08 = 2, 09 = 9, 10 = 50, 14 = 4, 15 = 5, 16 = 22
Źródło	`http://archive.ics.uci.edu/ml/datasets/Arrhythmia`
Opis	Zbiór zawiera dane osobnicze pacjentów (wiek, płeć, wzrost, waga) oraz dane pobrane z aparatury medycznej. Klasami w tym zbiorze są grupy określające występowanie bądź brak arytmii w pracy serca. Zbiór został wybrany ze względu na dużą liczbę cech, z których większość może być zredukowana, co stanowi dobre sprawdzenie zdolności do redukcji zbędnych wymiarów przestrzeni cech.
Uwagi	Oryginalny zbiór zawiera brakujące wartości dla niektórych cech, które zostały uzupełnione wartościami średnimi dla danej cechy.

4.3. Wstępne przetwarzanie danych

Przetwarzanie danych jest niezmiernie istotnym zagadnieniem towarzyszącym procesowi klasyfikacji, jak również innym metodom eksploracji danych. Celem stosowania wstępnego przetwarzania danych w zagadnieniu klasyfikacji jest taka transformacja zbioru danych, której zastosowanie utworzy nowy zbiór, dla którego algorytmy klasyfikacji rozwiążą problem w krótszym czasie lub z mniejszym błędem. Spośród wielu sposobów transformacji danych omówione zostaną te, które zostały zaimplementowane w ramach pracy.

4.3.1. Normalizacja

Jest jedną z najczęściej stosowanych metod wstępnego przetwarzania danych. Działanie tej metody polega na takim przekształceniu wartości wybranej cechy, aby wartości po transformacji zawierały się w przedziale $[min, max]$, gdzie najczęściej stosowanymi wartościami są $min = 0$ oraz $max = 1$. Zależność między danymi przed i po normalizacji określona jest wzorem 4.1.

$$x_i^{'} = min + \frac{(max - min)(x_i - x_{min})}{x_{max} - x_{min}}, \text{ gdzie} \tag{4.1}$$

$x_i^{'}$ - wartość cechy dla i-tego wektora po normalizacji
x_i - wartość cechy dla i-tego wektora przed normalizacją
x_{min} - minimalna wartość cechy
x_{max} - maksymalna wartość cechy
max - oczekiwana wartość maksymalna po normalizacji
min - oczekiwana wartość minimalna po normalizacji

4.3.2. Standaryzacja

Kolejnym sposobem transformacji danych zaimplementowanym w ramach pracy jest standaryzacja. Standaryzacja pozwala nam na uzyskanie zbioru danych, w którym wartości dla danej cechy mają oczekiwaną wartość średnią równą zero oraz odchylenie standardowe równe jedności. W ramach pracy zaimplementowana została metoda standaryzacji Z, w której zależność między danymi przed i po przetworzeniu wyraża się wzorem 4.2.

$$x_i^{'} = \frac{x_i - \mu}{\sigma} \tag{4.2}$$

$x_i^{'}$ - wartość cechy dla i-tego wektora po normalizacji
x_i - wartość cechy dla i-tego wektora przed normalizacją
μ - średnia wartość cechy przed standaryzacją
σ - odchylenie standardowe cechy przed standaryzacją

4.3.3. Principal Component Analysis

Zastosowanie wyżej wymienionych metod wstępnego przetwarzania danych może mieć wpływ na jakość uogólnienia klasyfikatorów. Metodę *Principal Component Analysis* (PCA, Analiza głównych składowych, *Proper Orthogonal Decomposition*, czasem nazywana także dyskretną transformacją Karhunen-Loéve)[32] można z kolei stosować w celu skrócenia czasu treningu i klasyfikacji algorytmu poprzez redukcję wymiaru przestrzeni cech, przy zachowaniu najistotniejszych informacji zawartych w zbiorze danych.

Algorytm *Principal Component Analysis* należy do grupy metod analizy statystycznej. Ogólne zadanie metody PCA można sformułować jako wyznaczenie obrotu układu współrzędnych tak, aby pierwsze jego współrzędne związane były ze zmiennymi, które niosą najwięcej informacji o danych w zbiorze. Dzięki tej operacji można wybrać składowe główne (zmienne niosące najwięcej informacji) i usunąć pozostałe, dzięki czemu dokonuje się redukcja wymiaru problemu i skraca czas obliczeń.

Algorytm PCA składa się z następujących kroków:

1. Transformacja wstępna. Taka transformacja polega na odjęciu średniej wartości każdej cechy od każdej pozycji dla danej cechy. Dzięki tej operacji uzyskujemy zbiór danych, którego średnia wynosi zero.

2. Wyznaczenie macierzy kowariancji dla zbioru uzyskanego po wstępnej transformacji.

3. Obliczenie wartości własnych macierzy kowariancji oraz ich posortowanie w porządku malejącym.

4. Wybór składowych głównych. Spośród wartości własnych wybierane są te, które uznane zostały za składowe główne.

5. Utworzenie macierzy składającej się z wektorów własnych macierzy kowariancji dla wybranych wartości własnych.

6. Transformacja względem składowych głównych. Mając wybrane wektory własne, odpowiadające składowym niosącym najwięcej informacji, możliwa jest transformacja oryginalnego zbioru danych do postaci, w której pierwszymi składowymi są składowe dla największych znalezionych wartości własnych macierzy kowariancji.

4.4. Metody walidacji

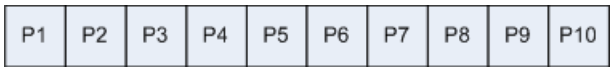
Klasyfikatory są tworzone do rozwiązywania realnych problemów. Celem osób stosujących metody klasyfikacji w praktyce jest uzyskanie jak najlepszego mechanizmu dla danego pro-

blemu, w związku z tym pojawia się konieczność walidowania algorytmów w odniesieniu do konkretnego zbioru danych.

Naturalną sytuacją jest, że badacze próbujący zbudować mechanizm klasyfikacji, dysponują zbiorem danych, który reprezentuje część możliwych przypadków występujących w obrębie badanej dziedziny. Dla ustalenia uwagi przytoczę hipotetyczną sytuację, w której grupa lekarzy współpracująca ze specjalistami z dziedziny eksploracji danych próbuje zbudować mechanizm automatyzujący diagnozę raka płuc. Lekarze dysponują zbiorem przebadanych dotychczas przypadków, a celem tworzonego mechanizmu jest jak najdokładniejsze diagnozowanie przypadków jeszcze nieznanych. Pojawia się problem „jak na podstawie istniejących danych określić, który klasyfikator będzie najlepszy". W praktyce stosuje się kilka prostych metod, które zostaną opisane poniżej. Większość z nich wywodzi się ze statystycznej metody sprawdzianu krzyżowego (walidacji krzyżowej). Algorytmy te zostaną udostępnione w ramach aplikacji wspierającej porównywanie sieci SONN z wybranymi metodami klasyfikacji.

Tworząc podzbiory treningowe warto zwrócić uwagę na rozkład wzorców według klas w zbiorze oryginalnym i podzbiorze testowym. Najczęściej dąży się do zachowania proporcji między przedstawicielami kolejnych klas. Takie podejście nosi nazwę walidacji stratyfikowanej (*ang. stratified*) i zostało uwzględnione podczas implementacji aplikacji wspierającej proces porównania metod klasyfikacji.

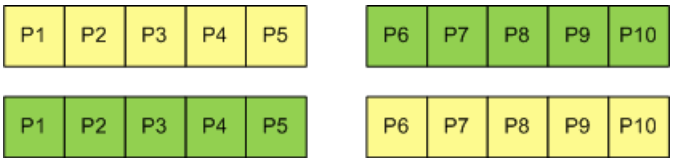
Opisując metody walidacji podawane przykłady dotyczyć będą dziesięcioelementowego zbioru przedstawionego na rysunku 4.2. W przykładach kolorem żółtym oznaczone zostaną wzorce ze zbioru uczącego, natomiast próbki ze zbioru walidacyjnego oznaczone są kolorem zielonym.



Rysunek 4.2: Metody walidacji, zbiór danych

4.4.1. k-fold cross-validation

Metoda *k-fold cross-validation* jest bardzo często stosowaną metodą walidacji, pozwalającą na uzyskanie informacji o możliwościach uogólnienia modelu, przy niskim nakładzie obliczeniowym. W metodzie *k-fold cross-validation* zbiór treningowy dzielony jest na k podzbiorów tak, że do walidacji przeznacza się każdą z próbek tylko raz. Na rysunkach 4.3 oraz 4.4 przedstawiono uzyskane podziały dzięki tej metodzie dla różnych parametrów.



Rysunek 4.3: Metody walidacji, k-fold cross validation (k=2)

Rysunek 4.4: Metody walidacji, k-fold cross validation (k=5)

4.4.2. Leave-one-out cross-validation

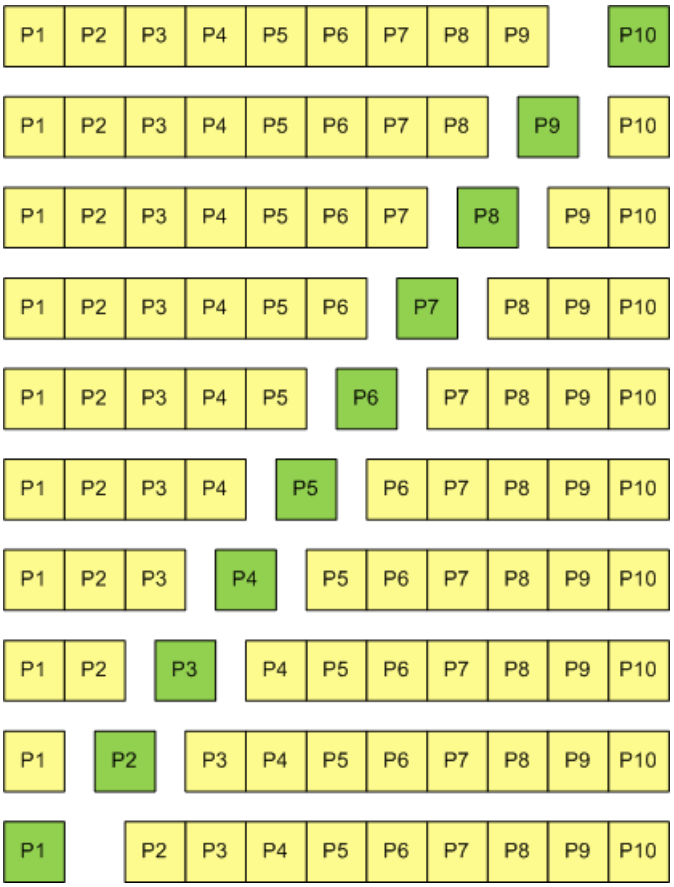
Szczególnym przypadkiem *k-fold cross-validation*, który często występuje jako osobna metoda walidacji jest *leave-one-out cross-validation* (LOOCV). Metodę tą można opisać jako n-krotną (gdzie n to ilość próbek danych w zbiorze uczącym) walidację dla każdej próbki, gdzie do uczenia modelu zostały wybrane pozostałe wzorce. Zatem jest to przypadek *k-fold cross-validation*, dla parametru k równego ilości próbek w zbiorze. Przykład podziału zbioru uczącego na podzbiory przedstawiony został na rysunku 4.5.

4.4.3. Podział procentowy

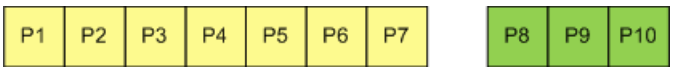
Często stosowaną metodą walidacji jest podział zbioru treningowego na dwie części: treningową i walidacyjną, przy czym część treningowa zawiera najczęściej więcej elementów (popularne proporcje to: $90:10$, $75:25$). Można porównać tą metodę do wykorzystania jednego z podziałów uzyskanych metodą *k-fold cross-validation*. Na poniższym przykładzie (patrz rysunek 4.6) przedstawiony został podział $70:30$.

4.4.4. Test na danych treningowych

Kolejnym testem jest sprawdzenie zachowania klasyfikatora na wszystkich dostępnych próbkach ze zbioru treningowego. Dzięki tej metodzie można stwierdzić, czy klasyfikator potrafi nauczyć się poprawnie rozpoznawać wszystkie znane przypadki treningowe.



Rysunek 4.5: Metody walidacji, LOOCV



Rysunek 4.6: Metody walidacji, Podział procentowy (70:30)

4.5. Kryteria porównania

Porównując metody klasyfikacji konieczne jest określenie, jakie kryteria zostaną zastosowane przy dokonywaniu porównań. Tematem pracy jest porównanie jakości uogólnienia i efektywności działania, dlatego te kryteria zostaną omówione szczegółowo. Konieczne jest analizowanie tych dwóch kryteriów jednocześnie, ponieważ można sobie wyobrazić algorytmy, które osiągają doskonałe wyniki przy ogromnym zużyciu zasobów lub zużywają minimalne ilości zasobów, ale fatalnie radzą sobie z zadaniem klasyfikacji. Opisane zostaną także inne kryteria, które mogą mieć wpływ na wybór metody klasyfikacji.

W praktyce, podczas porównań interesujący jest klasyfikator osiągający najlepsze rezultaty w kontekście stosowanej miary. Dla celów badawczych w tworzonej aplikacji oraz niniejszej pracy podawane będą także wartości średnie oraz maksymalne i minimalne wartości miar uzyskane dla danego algorytmu oraz różnych sposobów walidacji.

4.5.1. Jakość uogólnienia

Poprzez jakość uogólnienia (generalizacji) rozumiana jest skuteczność metody podczas rozpoznawania wzorców wybranych zgodnie z określoną metodą walidacji (patrz podrozdział 4.4). O uogólnieniu możemy mówić, gdy dana metoda nauczy się rozpoznawać klasę wzorca wykorzystując pewne podobieństwa do innych wzorców, które były obecne w procesie uczenia, przy jak najmniejszym stopniu bezpośredniego porównywania wzorców (przeuczenie, *ang. overfitting*).

Aby zobrazować tę różnicę można posłużyć się popularną analogią do nauki ze zrozumieniem, która pozwala na radzenie sobie z problemami wcześniej nieznanymi oraz nauki „na pamięć", która może sprawdzać się w problemach znanych, ale najczęściej jest zupełnie nieskuteczna w konfrontacji z nowym zagadnieniem. Istnieje wiele miar stosowanych w procesie walidacji, których wybór zależy od obszaru zastosowania klasyfikatora. Opisywane w tej pracy miary jakości uogólnienia podzielić można na dwie kategorie:

- Miary bazujące na poprawności określenia przynależności do klasy – współczynnik jakości określany jest wyłącznie na podstawie tego, czy klasyfikator poprawnie określił przynależność do klasy. Do takich miar należą Q_COR (współczynnik poprawności), Q_ERR (współczynnik błędu).

- Miary bazujące na pewności[2] określenia przynależności do klasy – miary te analizują współczynnik prawdopodobieństwa, z jakim klasyfikator określił przynależność do konkretnej klasy. Do takich miar należą: Q_SSE (*sum squared error*, sumaryczny błąd kwadratowy), Q_MSE (*mean squared error*, średni błąd kwadratowy), Q_RMSE (*root mean squared error*, pierwiastek średniego błędu kwadratowego), Q_ARV (*average relative variance*, średnia wariancja względna).

Miary należące do grupy miar jakości uogólnienia zostaną oznaczone przedrostkiem „Q_", w celu łatwiejszego rozróżnienia podczas analizy wyników porównań.

Współczynnik poprawności, Q_COR

$$Q\_COR = \frac{\text{ilość wzorców sklasyfikowanych poprawnie}}{\text{ilość wzorców w zbiorze danych}}$$

lub wartość procentową:

$$Q\_COR = \frac{\text{ilość wzorców sklasyfikowanych poprawnie}}{\text{ilość wzorców w zbiorze danych}} * 100\%$$

[2]Pewność określenia przynależności czasem bywa nazywana prawdopodobieństwem. Warto zwrócić uwagę, że to prawdopodobieństwo może nie spełniać aksjomatycznej definicji Kołmogorowa, a jedynie oznaczać potocznie rozumianą szansę tego, że badany wzorzec należy do pewnej klasy.

Współczynnik błędu, Q_ERR

$$Q\_ERR = \frac{\text{ilość wzorców sklasyfikowanych błędnie}}{\text{ilość wzorców w zbiorze danych}} * 100\%$$

Sumaryczny błąd kwadratowy, Q_SSE

$$Q\_SSE = \sum_{i=1}^{n}(f(x_i) - y_i)^2$$

Średni błąd kwadratowy, Q_MSE

$$Q\_MSE = \frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2 = \frac{1}{n}Q\_SSE$$

Pierwiastek średniego błędu kwadratowego, Q_RMSE

$$Q\_RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2} = \sqrt{Q\_MSE}$$

Średnia wariancja względna, Q_ARV

$$Q\_ARV = \frac{\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2}{\frac{1}{n}\sum_{i=1}^{n}(\frac{1}{n}\sum_{i=1}^{n}y_i - y_i)^2}$$

Macierz niepewności

Macierz niepewności (*ang. confusion matrix*) jest sposobem wizualizacji wyników działania klasyfikatora (lub innych metod eksploracji danych). Dzięki tej metodzie można uzyskać przejrzystą interpretację ile wzorców zostało zakwalifikowanych do poszczególnych klas oraz do jakiej klasy powinny były zostać zaklasyfikowane.

Na rysunku 4.7 przedstawiona została macierz niepewności wygenerowana w aplikacji zrealizowanej w części praktycznej (odnośnik). Ten konkretny przykład dotyczy jednego etapu walidacji krzyżowej algorytmu kNN na zbiorze danych Wine. Kolumny macierzy reprezentują wynik klasyfikacji (określenie przynależności do klasy), natomiast wiersze macierzy reprezentują oczekiwany wynik klasyfikacji (zakładając pełną poprawność). *Confusion matrix* dostarcza wygodnej wizualizacji, która dla tego przykładu pozwala łatwo stwierdzić, że dwa wzorce, które powinny zostać skojarzone z klasą „2" zostały niepoprawnie sklasyfikowane. Przy całkowitej skuteczności algorytmu macierz niepewności jest macierzą diagonalną.

Oczekiwana klasa \ Sklasyfikowano jako	1	2	3
1	6	0	0
2	1	5	1
3	0	0	5

Rysunek 4.7: Macierz niepewności dla jednego etapu walidacji krzyżowej algorytmu kNN na zbiorze Wine

4.5.2. Efektywność działania

Efektywność działania to ogólna nazwa grupy kryteriów, które związane są z czasem i wykorzystaniem zasobów przez mechanizmy klasyfikacji. Kryteria stosowane w porównaniach w ramach pracy zostaną wyjaśnione w kolejnych sekcjach.

Miary należące do grupy miar efektywności działania zostaną oznaczone przedrostkiem „E_”, w celu łatwiejszej analizy wyników porównań.

Czas treningu, E_TTR

Kryterium efektywności działania najczęściej utożsamiane jest z czasem treningu klasyfikatora. Czas treningu określa jak szybko algorytm potrafi odnaleźć zależności w zbiorze danych niezbędne do przeprowadzenia klasyfikacji. Dąży się do minimalizacji tego wskaźnika (w pracy został oznaczony symbolem E_TTR).

Podczas doświadczeń przeprowadzanych w ramach pracy zauważono, że takie zdefiniowanie kryterium efektywności działania bywa niewystarczające.

Czas klasyfikacji, E_TCL

Mając na uwadze dwuetapowość zadania klasyfikacji (rysunek 2.1) konieczna jest również analiza czasu klasyfikacji (wskaźnik oznaczony symbolem E_TCL). Okazuje się, że dla niektórych algorytmów (np. kNN, co można zaobserwować analizując wyniki porównań metod klasyfikacji) etap treningu jest mniej skomplikowany obliczeniowo od części klasyfikacyjnej. Z drugiej strony należy pamiętać o najczęstszym sposobie wykorzystywania metod klasyfikacji, które polega na jednorazowym wytrenowaniu algorytmu (często w warunkach, gdzie czas treningu nie ma dużego znaczenia). Mając wytrenowany klasyfikator jest on wielokrotnie używany do klasyfikacji wzorców, co wskazuje na bardziej istotną wartość czasu klasyfikacji (taki przypadek będzie miał miejsce w systemach czasu rzeczywistego realizujących wiele operacji klasyfikacji dla wytrenowanego klasyfikatora).

Chcąc obiektywnie porównać metody klasyfikacji, należy brać pod uwagę zarówno czas klasyfikacji, jak i treningu. Dodatkowo wspomnianą zaletą takiego podejścia jest możliwość skonfrontowania dowolnego z czasów z realnymi potrzebami uwarunkowanymi przez obszar zastosowań.

Naturalnie najlepszym klasyfikatorem z punktu widzenia efektywności działania będzie ten, który osiągnie najmniejsze czasy treningu i klasyfikacji. Przeprowadzając wiele symulacji warto także analizować średnie czasy dla każdego problemu i klasyfikatora, co zostało również uwzględnione w implementowanej w ramach pracy aplikacji.

Czas całkowity operacji, E_TOT

Miarą agregującą wartości czasu treningu i czasów klasyfikacji jest całkowity czas operacji (E_TOT), pokazujący jak dużo czasu zajęło danej metodzie utworzenie klasyfikatora i wykonanie wszystkich przewidzianych operacji klasyfikacji. Sposób jego obliczania obrazuje wzór 4.3.

$$E\_TOT = E\_TTR + \sum_i E\_TCL_i \tag{4.3}$$

Wykorzystanie pamięci

Często pomijanym kryterium, które ma szczególne znaczenia w systemach wbudowanych (*ang. embeded*) jest ilość pamięci wykorzystywanej przez klasyfikator. Możemy rozróżnić dwie miary dla tego kryterium:

- Złożoność pamięciowa – zależność wymaganej pamięci RAM używanej w procesie treningu klasyfikatora od wielkości zbioru treningowego. Ze względu na zastosowanie technologii wykorzystującej automatyczne zwalnianie pamięci[3] (*ang. garbage collecting*) niemożliwy był pomiar tej wielkości, przez co nie znajdzie się ona w wynikach porównań.

- Objętość pamięciowa wytrenowanego klasyfikatora – realna wielkość zajmowanego miejsca w pamięci dla wytrenowanego klasyfikatora. Oznacza ona jak wiele miejsca w pamięci potrzeba do przechowywania obiektu utworzonego klasyfikatora. Pomiar tego wskaźnika może być przeprowadzony poprzez serializację[4] do formatu binarnego obiektu klasyfikatora. Taka wielkość jest udostępniana w symulacjach oraz występuję w porównaniach, jest ona oznaczona symbolem E_MEM.

[3]Więcej informacji o mechanizmie garbate collecting w środowisku .NET znaleźć można na stronach MSDN (`http://msdn.microsoft.com/en-us/magazine/bb985010.aspx`, `http://msdn.microsoft.com/en-us/library/0xy59wtx.aspx`)

[4]Więcej informacji o serializacji w środowisku .NET znaleźć można na stronach MSDN (`http://msdn.microsoft.com/en-us/library/system.runtime.serialization.aspx`)

4.5.3. Skalowalność

Kryterium skalowalności (*ang. scalability*) dla metod klasyfikacji określa czy konkretny algorytm może zostać zastosowany dla dowolnie dużego zbioru danych. Jeżeli algorytm można zastosować dla dużego zbioru danych i osiąga on dobre wyniki, to o takim algorytmie mówimy, że jest dobrze skalowalny lub, że dobrze się skaluję. W praktyce, ciężko określić miarę, która w sposób obiektywny potrafiła porządkować klasyfikatory pod kątem skalowalności. Z tego powodu skalowalność jest rzadko wymienianym kryterium podczas porównywania metod klasyfikacji.

W rzeczywistości skalowalność ściśle związana jest z kryteriami efektywności działania (rozdział 4.5.2) oraz jakości uogólnienia (rozdział 4.5.1). Ponieważ celem niniejszej pracy jest wykonanie porównań uwzględniających zbiory o różnej wielkości, w rozdziale 6.9 przedstawione zostaną zależności czasu operacji oraz potrzebnej pamięci, w funkcji stopnia trudności zbioru danych. Należy zwrócić uwagę, że taka analiza może okazać się niemiarodajna, ponieważ nie istnieje ogólny sposób wyznaczania stopnia trudności zbioru danych. Ilość elementów w zbiorze danych nie jest jedynym czynnikiem wpływającym na efektywność działania oraz jakość uogólnienia. Bardzo często kluczową rolę odgrywają zależności pomiędzy elementami zbioru danych, stopień ich skorelowania, ilość oraz rozkład klas, a także wiele innych czynników. Należy o tym pamiętać analizując takie zestawienie.

4.5.4. Interpretowalność

Interpretowalność jest kryterium, które bardziej związane jest z podejściem do sposobu klasyfikacji, niż z konkretnym klasyfikatorem. To kryterium porządkuje klasyfikatory według możliwości zrozumienia sposobu klasyfikacji na podstawie struktury zbudowanego klasyfikatora. Interpretowalność jest kryterium subiektywnym i niemierzalnym, natomiast istnieją przypadki, kiedy jest istotne w procesie podejmowania decyzji o wyborze klasyfikatora. Interpretowalność związana jest z zaufaniem do klasyfikatora (*ang. trust*).

Wyobraźmy sobie sytuację, w której konieczne jest zweryfikowanie zbudowanego modelu klasyfikacji przez grupę specjalistów z konkretnej dziedziny. Prawdopodobnie okaże się, że algorytm drzew decyzyjnych, w których łatwo można prześledzić proces podejmowania decyzji o przynależności do klasy oraz sporządzić graficzną wizualizację tego procesu, okaże się łatwiejszym do zinterpretowania od metody *Radial Basis Function Networks*, w której struktura sieci niesie wiele mniej informacji o procesie decyzyjnym.

4.5.5. Inne kryteria

Oprócz wymienionych kryteriów porównania istnieć mogą inne. Bardzo często w praktyce spotyka się porównania algorytmów w kontekście konkretnej dziedziny (np. medycyna, genetyka, mechanika).

5. Część praktyczna

W tym rozdziale przedstawione zostaną istotne szczegóły związane z realizacją praktycznej części pracy. Omówione zostaną założenia, które uwarunkowały ostateczny kształt stworzonych rozwiązań. W dalszej części zaprezentowane będą wybrane metody inżynierii oprogramowania, dzięki którym udało się zakończyć projekt sukcesem oraz wymienione zostaną kluczowe technologie użyte podczas realizacji części praktycznej. W podrozdziale 5.4. szczegółowo omówione zostaną istotne aspekty implementacyjne, które mogą posłużyć podczas dalszego rozwoju aplikacji. W ostatniej części rozdziału przedstawione i omówione zostaną najważniejsze elementy aplikacji wspierającej porównania.

5.1. Założenia

Nadrzędnym celem realizacji części praktycznej jest wykonanie narzędzi pozwalających na dokonanie porównań rozmaitych metod klasyfikacji z sieciami SONN. Z uwagi na ogrom istniejących algorytmów nie sposób było zaimplementować wszystkich z nich, dlatego duży nacisk położony został na możliwość rozszerzenia biblioteki o nowe algorytmy. Kolejnym istotnym aspektem, który został uwzględniony podczas implementacji jest całkowite oddzielenie kodu samych algorytmów od aplikacji wspierającej porównania. Dzięki temu można łatwo użyć zestawu algorytmów w dowolnej aplikacji. Mając na uwadze konieczność podziału całego projektu na dwa moduły (bibliotek algorytmów, aplikacja wspierająca porównania), można określić dokładniej wymagania w stosunku do każdego z nich.

Biblioteka algorytmów:

- Spójność – wszystkie algorytmy powinny implementować wspólny interfejs (*IAlgorithm*), który będzie reprezentował algorytm klasyfikacji. Dzięki temu łatwiejsze będzie rozszerzanie kodu o nowe algorytmy. Ponadto, taka konwencja sprzyja poprawie jakości oraz czytelności kodu.

- Jakość – biblioteka algorytmów może być wykorzystana w innych projektach (w szczególności w aplikacji wspierającej porównania). Z tego powodu konieczne jest

napisanie zestawu testów, które oprócz wymienionego powodu powinny być przydatne podczas dalszego rozwoju biblioteki (np. SONN-4).

- Import danych z różnych źródeł – obok zestawu algorytmów konieczne jest stworzenie mechanizmów pozwalających na importowanie z rozmaitych źródeł. W rzeczywistości dane mogą pochodzić z dowolnego źródła i występować w dowolnym formacie. Konieczne jest zaimplementowanie podstawowych mechanizmów (.csv, ADO.NET), przy jednoczesnym zachowaniu możliwości rozszerzenia tej funkcjonalności (interfejs *IDataProvider*).

Aplikacja wspierająca porównania:

- Dostępność – jak wielokrotnie wspominano w pracy istnieje wiele porównań algorytmów klasyfikacji. Za cel postawiono udostępnienie realizowanej pracy tak, aby w przyszłości osoby realizujące podobne zadania mogły wykorzystać stworzoną aplikację lub skonfrontować swoje wyniki z wynikami, które można w niej uzyskać. Mając to na uwadze warto wykorzystać zalety popularnego medium jakim jest Internet i dlatego realizowana aplikacja została wykonana jako aplikacja internetowa.

- Prostota – istnieje kilka dojrzałych narzędzi wspomagających eksplorację danych [29, 30], które zawierają wiele algorytmów, metod przetwarzania danych, mechanizmów walidacji oraz innych opcji konfiguracyjnych. Bardzo często konsekwencją tego jest nadmierny stopień skomplikowania czynności tworzenia klasyfikatora i porównania jego wyników z wynikami innych klasyfikatorów. Dlatego celem stawianym przed aplikacją realizowaną w ramach pracy jest zapewnienie prostoty oraz intuicyjności wykonywanych czynności podczas pracy z aplikacją.

- Funkcjonalność – proces porównywania metod klasyfikacji (opisany w rozdziale 4.) składa się z kilku niezbędnych etapów, z których każdy powinien zostać zaimplementowany w aplikacji.

5.2. Wybrane aspekty inżynierii oprogramowania

Realizując projekt informatyczny, który ma spełniać pewne wymogi odnośnie jakości, konieczne jest stosowanie sprawdzonych metod oraz praktyk, które związane są z dziedziną inżynierii oprogramowania. Poniżej wymienionych zostało kilka aspektów związanych z inżynierą oprogramowania, które odegrały szczególną rolę w realizowanym projekcie. Wskazane zostały także narzędzia, które zostały wykorzystane podczas prac.

- Zarządzanie projektem – realizując rozmaite projekty (nie tylko informatyczne) warto dzielić pracę na mniejsze etapy, kontrolować postęp w ich realizacji, śledzić błędy i wykonywać inne czynności w ramach zarządzania zadaniami, dzięki którym całość może zakończyć się z oczekiwanym rezultatem. Narzędzia które zostało wykorzystane, aby wspierać zarządzanie projektem to pakiet usługi dostępnych w ramach portalu internetowego assembla.com.

- Wersjonowanie kodu – chcąc zabezpieczyć się przed utratą kodu aplikacji (na skutek awarii stacji roboczej lub wprowadzeniu trudnego do wykrycia błędu) warto korzystać z narzędzi umożliwiających wersjonowanie kodu. Podczas prac nad projektem stosowane było rozwiązanie Subversion (SVN).

- Testy aplikacji – jednym ze sposobów zwiększenia jakości tworzonego kodu jest utworzenie specjalnych testów sprawdzających jego poprawność. Istnieje wiele rodzajów testów aplikacji. Te, które odegrały szczególną rolę podczas prac nad projektem, to testy jednostkowe (pisane przy wsparciu zestawu bibliotek NUnit). Dzięki testom możliwe było sprawdzenie poprawności działania algorytmów, co okazało się szczególnie istotne podczas implementacji algorytmu SONN.

- Ciągła integracja (*ang. continuous integration*) – automatyzacja procesu budowania i testowania aplikacji pomaga wychwycić błąd niezwłocznie po wprowadzeniu go do repozytorium kodu. W ramach projektu stosowano popularne narzędzie Cruise Control, które umożliwiło ciągłą integrację projektu.

- Wzorce projektowe – stosowanie sprawdzonych wzorców projektowych oraz dobrych praktyk programistycznych jest niezbędną składową projektów, gdzie określone są pewne wymagania co do jakości. Kilka z zastosowanych rozwiązań zostanie przedstawionych w dalszej części tego rozdziału.

5.3. Stosowane technologie

Do najważniejszych technologii wykorzystanych w ramach projektu należą:

- .NET Framework 3.5 – platforma programistyczna. Jako język programowania został wybrany nowoczesny C# 3.0 wraz z mechanizmem LINQ. Do realizacji warstwy prezentacji zastosowano ASP.NET WebForms. W oparciu o ADO.NET zrealizowano jeden z providerów danych wykorzystywanych w projekcie.

- SQLServer 2005 –silnik bazy danych służącej do przechowywania stosowanych zbiorów danych.

- jQuery, AjaxControlToolkit – biblioteki wykorzystane przy programowaniu internetowego interfejsu użytkownika.
- ZedGraph – biblioteka pozwalająca na tworzenie wykresów.
- NUnit – framework wspierający tworzenie testów jednostkowych aplikacji.
- Subversion – narzędzie wspierające wersjonowanie kodu.

5.4. Szczegóły implementacyjne

W tym podrozdziale omówione zostaną aspekty techniczne zrealizowanego rozwiązania, dzięki którym możliwa jest rozbudowa projektu oraz integracja z innymi aplikacjami. Przedstawiona zostanie struktura projektu oraz podstawowe interfejsy.

5.4.1. Struktura projektu

Zrealizowane w ramach pracy rozwiązanie składa się z 9 modułów enkapsulujących podstawowe funkcje systemu.

W ramach biblioteki z algorytmami wyróżnić można następujące moduły:

- *Algorithms* - zbiór klas reprezentujących algorytmy klasyfikacji oraz inne algorytmy wykorzystywane podczas treningu klasyfikatorów (np. algorytm analizy skupień *k-mean*). W tym module znajdują się podstawowe interfejsy, które zostały opisane w dalszej części pracy.
- *Algorithms.UnitTests* - zestaw testów jednostkowych dla klas modułu *Algorithms*.
- *Data* - zbiór klas związanych z mechanizmami dostarczania (*Data.Providers*) i przetwarzania (*Data.Preprocessors*) danych używanych przez algorytmy klasyfikacji.
- *Data.UnitTests* - zestaw testów jednostkowych dla klas modułu *Data*.

Kolejne trzy moduły zawierają logikę oraz elementy warstwy prezentacji aplikacji wspierającej porównania metod klasyfikacji:

- *Web* - główny moduł aplikacji internetowej. Zawiera logikę stron, elementy interfejsu graficznego oraz ekrany systemu pomocy.
- *Web.Controls* - moduł zawierający kod niestandardowych kontrolek ASP.NET (*ang. control*) używanych w aplikacji.

- *DAL* - moduł *DAL* (*Data Access Layer*) realizuje warstwę dostępu do danych. Jako dane rozumiane są zestawy algorytmów, metod przetwarzania oraz ich parametry, które używane są w aplikacji. Logika zawarta w tej części zezwala na rozszerzanie aplikacji wspierającej porównania o nowe algorytmy i metody przetwarzania danych.

Oprócz wymienionych modułów wyróżnić można także moduły realizujące funkcje pomocnicze:

- *Utils* - zbiór często wykorzystywanych w aplikacji funkcji pomocniczych (np. operacje matematyczne).
- *Utils.UnitTests* - zestaw testów jednostkowych dla klas modułu *Utils*.

5.4.2. Podstawowe interfejsy

Interfejs *IAlgorithm* jest kluczowym interfejsem określonym w systemie. Dzięki niemu możliwe jest korzystanie ze skomplikowanych klas algorytmów w spójny i intuicyjny sposób, nawiązujący do dwuetapowego zadania klasyfikacji (patrz rysunek 2.1).

```
public interface IAlgorithm
{
    void Train(ITrainingData trainingData);
    IClassificationResults Classify(double[] testData);
}
```

Każdy z algorytmów klasyfikacji musi realizować dwie operacje, które określone są w ramach interfejsu *IAlgorithm*:

- *Train* – trening klasyfikatora w oparciu o znany zbiór treningowy. Ta operacja powinna wywołać logikę charakterystyczną dla danego algorytmu klasyfikacji oraz przygotować tak ten obiekt, aby możliwe było wielokrotne wywołanie metody pozwalającej na sklasyfikowanie nieznanego wektora danych.
- *Classify* – operacja klasyfikacji. Dla podanego wektora liczb rzeczywistych metoda ta powinna zwrócić obiekt klasy implementującej interfejs *IClassificationResults*, który zawiera informacje o przebiegu klasyfikacji podanego wektora wartości cech.

```
public interface IAlgorithmInfoProvider
{
    Dictionary<string, string> SpecificInformation { get; }
}
```

Interfejs *IAlgorithmInfoProvider* wymaga od klas, które go implementują definicji jednej właściwości (*ang. property*), dzięki której możliwe jest uzyskanie słownika zawierającego szczegółowe informacje o obiekcie algorytmu.

Wprowadzenie powyższych interfejsów spełnia założenia wzorca projekowego fasada (*ang. facade*)[33, 34], dzięki któremu możliwe jest ukrycie skomplikowanych szczegółów biblioteki algorytmów, pozostawiając proste interfejsy, które mogą zostać użyte w zależności od wymaganej sytuacji.

Poza opisanymi interfejsami występującymi w module *Algorithms* na uwagę zasługują także inne, których znajomość może okazać się istotna podczas rozszerzania lub wykorzystywania opisywanego kodu:

- *Algorithms.IClassificationResults* - dzięki wprowadzeniu tego interfejsu możliwe jest spójne wykorzystywanie wyników klasyfikacji zwracanych przez algorytmy.
- *Data.IDataProvider* - interfejs określający sygnatury metod, które realizują importowanie danych z rozmaitych źródeł.
- *Data.IDataPreprocessor* - podobnie jak algorytmy klasyfikacji, które muszą implementować interfejs *IAlgorithm* każda z metod wstępnego przetwarzania danych musi realizować metody wymagane przez interfejs *IDataPreprocessor*.

Naturalnie, oprócz powyższych interfejsów, biblioteka zawiera inne składowe, głównie klasy realizujące często skomplikowane algorytmy (np. SONN). Nie sposób ich jednak tutaj wszystkich opisać, dlatego przedstawione zostały tylko podstawowe interfejsy, dzięki którym możliwe jest korzystanie z biblioteki w rozmaitych aplikacjach.

5.5. Prezentacja aplikacji

Zgodnie z założeniami, aplikacja wspierająca porównania ma być narzędziem umożliwiającym praktyczną realizację procesu przedstawionego w rozdziale 4. Użytkownik pragnący porównać wybrane metody klasyfikacji musi wykonać pięć kroków zaprezentowanych i opisanych poniżej.

Warto zwrócić uwagę na system pomocy, który dostępny jest podczas pracy z aplikacją. Można w nim znaleźć wiele cennych informacji dotyczących kolejnych stron witryny, czy zapoznać się ze specyfiką dostępnych algorytmów klasyfikacji.

5.5.1. Wybór treningowego zbioru danych

Pierwszym etapem pracy z aplikacją jest określenie zbioru danych, dla którego przeprowadzone zostaną porównania. Użytkownik ma następujące możliwości wprowadzenia danych:

- Import z pliku .csv - popularna metoda przechowywania zbiorów danych. Wiele programów (np. arkusze kalkulacyjne) zawierają możliwość eksportu do formatu .csv (*ang.*

comma separated value), dzięki czemu można przeprowadzać klasyfikację dla danych pochodzących z takich źródeł.

- Import ze źródeł danych zgodnych z ADO.NET - możliwe jest importowanie danych ze źródeł danych wspieranych przez ADO.NET[35]. Aby było to możliwe konieczne jest podanie następujących parametrów (szczegóły w systemie pomocy aplikacji):

  1. Provider - nazwa dostawcy danych (np. *System.Data.SqlClient*).
  2. ConnectionString - łańcuch połączenia (*ang. connection string*) pozwalający na połączenie ze źródłem danych.
  3. Polecenie SELECT - kwerenda zgodna ze źródłem danych, dzięki której możliwe jest pobranie danych.

- Wybór predefiniowanego zbioru - wszystkie zbiory używane podczas porównań (patrz rozdział 4.2) mogą zostać wczytane bez konieczności podawania pliku .csv lub określania specyfikacji źródła danych ADO.NET.

Po prawidłowym wczytaniu danych użytkownik zostanie powiadomiony stosownym komunikatem. Istnieje także możliwość wyświetlenia zaimportowanego zbioru danych. W tym celu należy odznaczyć opcję „Wyświetl zbiór treningowy".

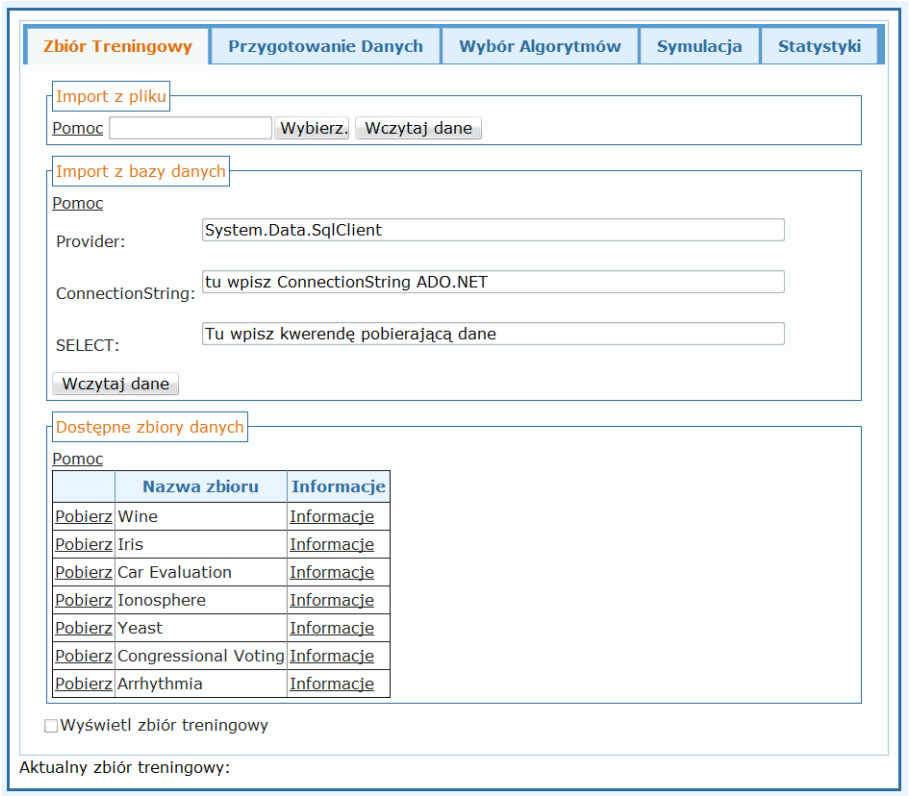
Ekran aplikacji umożliwiający wykonanie powyższych czynności przedstawiony został na rysunku 5.1.

5.5.2. Wybór metod wstępnego przetwarzania danych

Po wczytaniu danych istnieje możliwość wybrania oraz określenia parametrów metod wstępnego przetwarzania danych (patrz rysunek 5.2). Dostępne możliwości to:

- Normalizacja (opisano w rozdziale 4.3.1).
- Standaryzacja (opisano w rozdziale 4.3.2).
- Principal Component Analysis (opisano w rozdziale 4.3.3).
- Mix - zmiana porządku wektorów treningowych w zbiorze.

Podobnie jak w poprzednim kroku, możliwy jest podgląd wstępnie przetworzonych danych.



Rysunek 5.1: Aplikacja - wybór treningowego zbioru danych

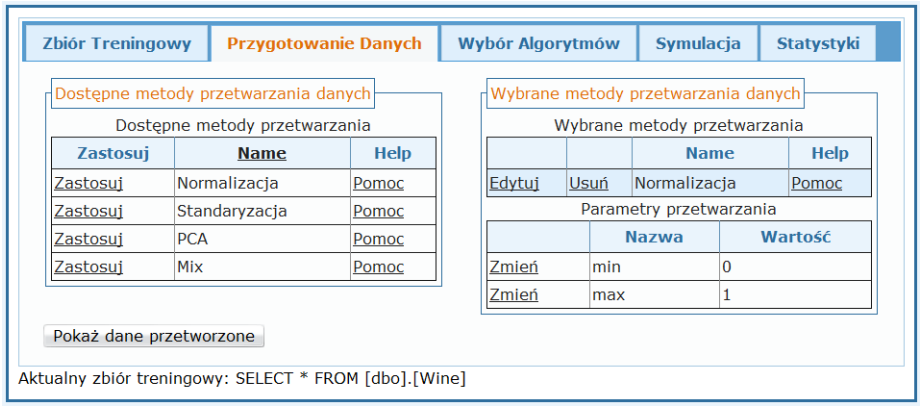
5.5.3. Wybór oraz parametryzacja algorytmów

Kolejnym krokiem omawianego procesu jest wybór oraz odpowiednia parametryzacja algorytmów, które mają zostać ze sobą porównane. Dostępne są wszystkie algorytmy szczegółowo omówione w rozdziale 3. Możliwy jest wybór wielu istancji metody konkretnego typu i określenie różnej listy parametrów dla każdej z nich, dzięki temu możliwe jest porównywanie algorytmów tego samego rodzaju. Zrzut ekranu przedstawiający omawiany widok został pokazany na rysunku 5.3.

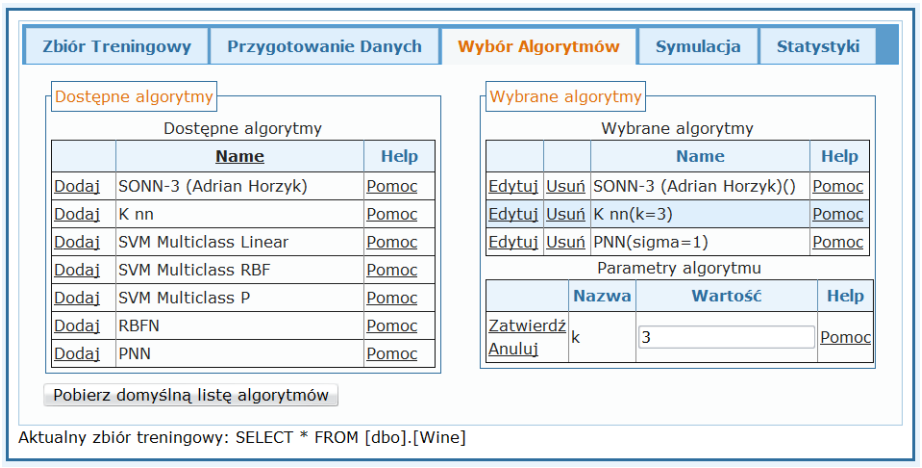
5.5.4. Konfiguracja symulacji

Ostatnim krokiem konfiguracyjnym przed rozpoczęciem symulacji jest wybór metod walidacji. Podobnie jak w przypadku algorytmów, wszystkie dostępne metody zostały omówione we wcześniejszej części pracy (patrz rozdział 4.4. Niektóre z nich (np. podział procentowy, *k-fold cross-validation*) mogą zostać odpowiednio sparametryzowane.

Oprócz wyboru metody walidacji możliwe jest określenie poziomu szczegółowości statystyk, które są przygotowywane w procesie symulacji. Jest to bardzo istotne podczas porównywania wielu algorytmów lub podczas korzystania z wielu metod walidacji. Ograniczenie szczegółowości statystyk przyspiesza działanie aplikacji w trakcie przeglądania wyników.



Rysunek 5.2: Aplikacja - wybór metod przetwarzania danych



Rysunek 5.3: Aplikacja - wybór algorytmów

Kolejne poziomy szczegółowości statystyk oznaczają:

1. Poziom algorytmu - zawiera statystyki przekrojowe dla wszystkich stosowanych metod walidacji.

2. Poziom metody walidacji - dla każdej z wybranych metod walidacji dostępne są statystyki agregujące walidację na zbiorze treningowym i testowym.

3. Poziom walidacji na zbiorze treningowym/testowym - dla każdej metody walidacji przeprowadzana jest walidacja na podzbiorach trenigowym i testowym. Poziom 3. pozwala na analizę statystyk dla walidacji przeprowadzonej na wymienionych podzbiorach.

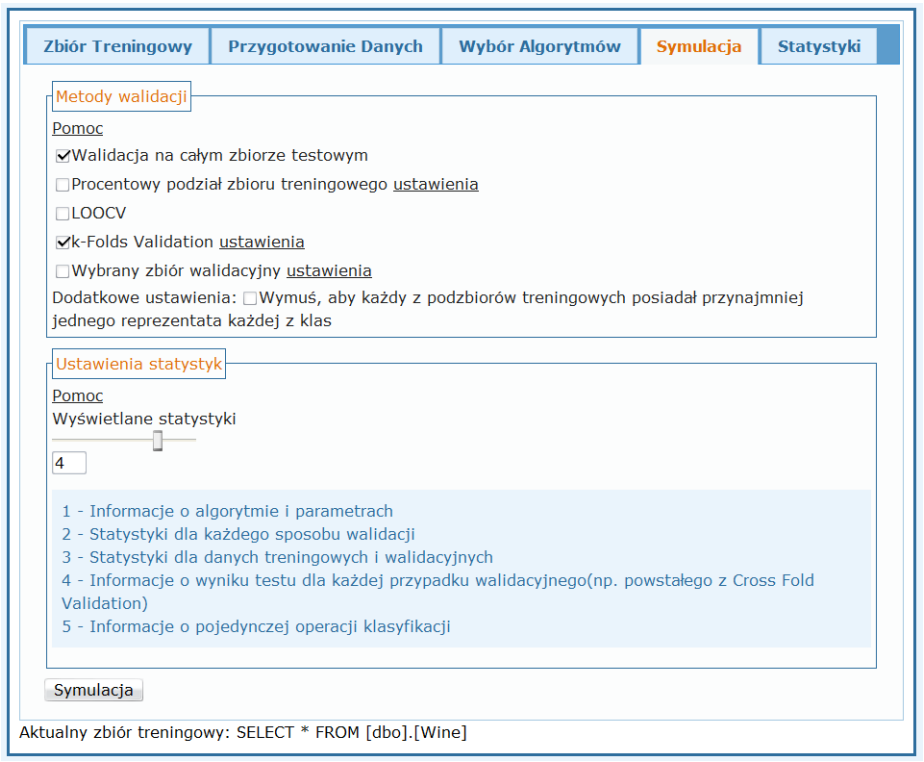
4. Poziom konkretnego klasyfikatora - na tym poziomie (zależnie od metody walidacji) dostępnych jest jeden lub więcej wpisów, związanych z konkretną operacją tworzenia klasyfikatora. Jest to najbardziej istotny poziom statystyk przedstawiający takie dane jak:

informacje o klasyfikatorze, macierz niepewności, wskaźniki jakości i efektywności działania dla klasyfikatora i inne.

5. Poziom operacji klasyfikacji - najbardziej atomowy poziom statysyk zawierający informacje o pojedynczej operacji klasyfikacji wykonanej w procesie walidacji (zaleca się pominięcie tego poziomu szczegółowości w celu przyspieszenia działania aplikacji).

Po zakończeniu konfiguracji należy wcisnąć przycisk „Symulacja", który rozpocznie proces treningu i klasyfikacji, dla każdego z wybranych algorytmów. Proces ten może trwać bardzo długo - zależnie od wybranego zbioru danych, zestawu algorytmów i metod walidacji. Po zakończeniu symulacji aplikacja przedstawi wyniki przeprowadzonych działań.

Omawiane możliwości konfiguracji dostępne są na ekranie aplikacji przedstawionym na rysunku 5.4.



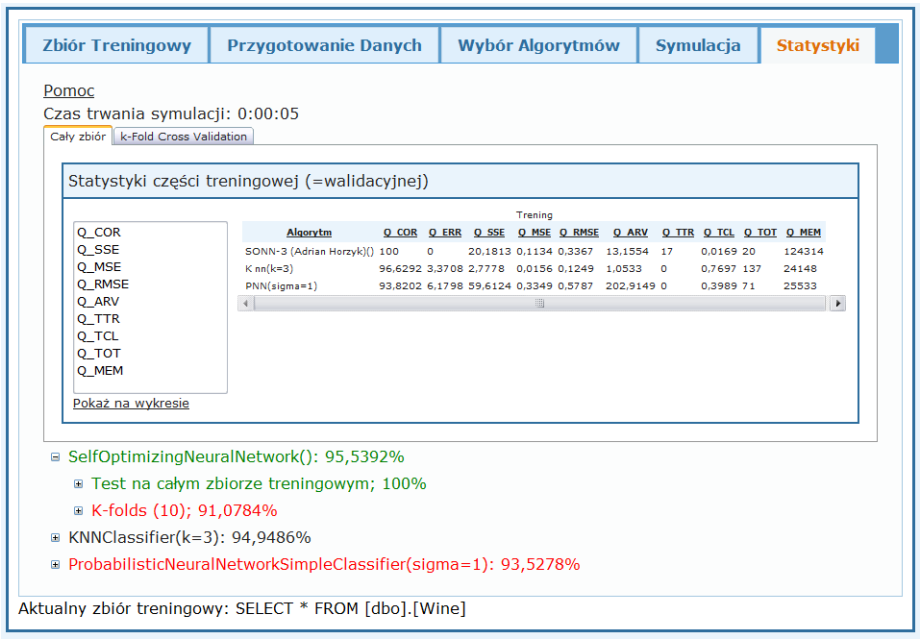
Rysunek 5.4: Aplikacja - konfiguracja symulacji

5.5.5. Przeglądanie wyników

Po wykonaniu symulacji aplikacja przedstawi wyniki przeprowadzonych działań. Wygląd ekranu zawierającego wyniki przedstawia rysunek 5.5. Na tym ekranie użytkownik może przeanalizować szczegółowo cały przeprowadzony przez program proces. Oprócz statystyk omówionych w poprzednim podrozdziale dostępne są także zbiorcze statystyki jakości uogólnienia i efektywności działania, zgrupowane według każdej z wybranych metod walidacji.

Ciekawą możliwością oferowaną na tym etapie pracy z aplikacją jest możliwość zapisania wybranego klasyfikatora do postaci binarnej (serializacja). Dzięki temu osoba starająca się wybrać najlepszy algorytm do praktycznych zastosowań może, po przeprowadzeniu odpowiednich porównań, wybrać i zapisać najlepszy z klasyfikatorów dla danego problemu. Tak zapisany obiekt algorytmu można odtworzyć w dowolnej aplikacji wykorzystującej biblioteki projektu wywołując statyczną metodę klasy *Algorithms.AlgorithmHelper*:

```
public static IAlgorithm Deserialize(Stream stream)
```



Rysunek 5.5: Aplikacja - wyniki

6. Wyniki porównań metod klasyfikacji

W niniejszym rozdziale przedstawione zostaną wyniki porównań wybranych algorytmów wykonane za pomocą zaimplementowanej aplikacji opisanej w poprzednim rozdziale. W kolejnych podrozdziałach zaprezentowane zostaną wyniki algorytmów dla każdego z wybranych zbiorów danych. Dodatkowo dla zbioru Wine porównane zostaną wyniki algorytmów po zastosowaniu popularnych metod wstępnego przetwarzania danych.

Jako metody walidacji wybrane zostały:

- test na całym zbiorze treningowym - wskaźniki jakości tak jak w rozdziale 4.5.

- dziesięcioelementowa walidacja krzyżowa (*10-fold cross-validation*) - wskaźniki opisane w rozdziale 4.5 dotyczą wartości średniej (*avg*) dla wszystkich podzbiorów powstałych w wyniku testu krzyżowego oraz wartości minimalnej (*min*) i maksymalnej (*max*), które zostały osiągnięte dla konkretnych podzbiorów. Przedstawione zostaną wyniki osiągnięte zarówno dla danych treningowych, jak i podzbioru walidacyjnego.

W podrozdziale 6.9 przeanalizowane zostaną także wskaźniki, które mogą zobrazować zachowanie konkretnych algorytmów w zależności od wielkości i stopnia skomplikowania zbioru danych.

6.1. Stanowisko testowe i uwagi o procesie porównywania

Wszystkie prezentowane porównania wykonane zostały na komputerze o parametrach przedstawionych w tabeli 6.1. Szczegóły techniczne aplikacji wykorzystanej do porównań opisane zostały w rozdziale 5.

W tabelach przedstawiających wskaźniki jakości uogólnienia i efektywności działania oprócz wartości liczbowych można znaleźć następujące informacje:

- NaN (*ang. not a number*) - wartość, która powstała w wyniku nieprawidłowej operacji numerycznej (np. dzielenie przez wartość bliską zero).

Tablica 6.1: Parametry techniczne komputera, na którym zrealizowano porównania

Parametr	Wartość
Procesor	Intel Core 2 Duo CPU T8300 2.40GHz
Pamięć RAM	4096 MB
System operacyjny	Windows Vista 32-bit
.NET Framework	3.5

- nieakceptowalny czas treningu - dla kilku algorytmów czas tworzenia klasyfikatora przekroczył dopuszczalną wartość (wielkość zależna od zbioru danych, najczęściej kilka godzin).
- nie udało się przeprowadzić treningu - brak możliwości przeprowadzenia treningu z powodu konieczności wykonania niedozwolonej operacji matematycznej (np. próba odwrócenia macierzy osobliwej).

Z powodu dużej ilości prezentowanych wskaźników jakości konieczne było zastosowanie skrótowych nazw algorytmów. Legenda wyjaśniające zastosowane oznaczenia została zamieszczona w tabeli 6.2.

6.2. Porównania dla zbioru Wine

Jednym z najbardziej popularnych zbiorów danych, na których testowane są metody klasyfikacji, jest zbiór Wine. Dla tego zbioru przeanalizowany zostanie wpływ zastosowania różnych metod wstępnego przetwarzania danych oraz przeprowadzone zostaną porównania części treningowej i walidacyjnej powstałych w wyniku zastosowania dziesięcioelementowego testu krzyżowego.

Tabele 6.4, 6.5, 6.6 oraz 6.7 przedstawiają wyniki wybranych algorytmów klasyfikacji dla zbioru Wine poddanego zaimplementowanym metodom wstępnego przetwarzania danych. Na podstawie takiego porównania można zauważyć, że:

- Dla danych znormalizowanych (Tablica 6.5 większość algorytmów uzyskała najlepszą skuteczność (Q_COR) oraz potrafiła utworzyć klasyfikator i przeprowadzić proces klasyfikacji w stosunkowo krótkim czasie (E_TOT). W dalszych porównaniach wykonanych w tej pracy będzie to jedyna stosowana metoda przetwarzania danych.
- Zastosowanie algorytmu PCA do redukcji wymiaru przestrzeni cech dla wielu algorytmów istotnie skraca czas obliczeń przy jednoczesnym obniżeniu zdolności klasyfikacyjnych.

Tablica 6.2: Legenda dla algorytmów i parametrów występujących w wynikach porównań

Symbol	Algorytm	Opis parametrów
SONN-3	Self Optimizing Neural Networks	
KNN(k)	k Nearest Neighbours	(patrz tabela 3.1)
SVML(c;e)	Support Vector Machines	klasyfikator liniowy, c - cost, e - eps, (patrz tabela 3.2)
SVMR(c;s;e)	Support Vector Machines	klasyfiaktor nieliniowy, metoda jądrowa - f. gaussowska, s - sigma, pozostałe parametry jak dla SVML (patrz tabela 3.2)
SVMP(c;d;e)	Support Vector Machines	klasyfiaktor nieliniowy, metoda jądrowa: f. wielomianowa, (patrz tabela 3.2)
RBFN(k;f;c)	Radial Basis Function Networks	k - ilość neuronów warstwy ukrytej, f - funkcja radialna (1 - f. gaussowska, 2 - f. wielokwadratowa), c - metoda wyboru centrów (1 - wybór losowy, 2 - algorytm analizy skupień K-means), (patrz tabela 3.3)
PNN(sigma)	Probabilistic Neural Networks	(patrz tabela 3.4)

- Dla sieci SONN zastosowanie algorytmu PCA poskutkowało nieznacznym zwiększeniem czasu konstrukcji, znaczącym zwiekszeniem używanej pamięci (Q_MEM) oraz struktury sieci (porównanie struktury sieci SONN przy zastosowaniu normalizacji i algorytmu PCA znajduje się w tabeli 6.3). Można przypuszczać, że algorytm dyskryminacja ADL-BCA potrafi sprawniej odszukać informacje pozwalające na utworzenie klasyfikatora od algorytmu PCA.
- Sieci SONN, niezależnie od zastosowanej metody przetwarzania, potrafią bezbłędnie rozpoznawać wzorce zastosowane w procesie konstrukcji sieci.

Tablica 6.3: Struktura sieci SONN dla wybranych metod przetwarzania danych, Wine

Element struktury	**Normalizacja**	**PCA**
Liczba neuronów LBN	7	41
Liczba neuronów ARN	6	8
Liczba neuronów MSN	3	3
Liczba warstw sieci	5	5
Liczba połączeń międzyneuronowych	38	200

Tablica 6.4: Wine, brak wstępnego przetwarzania, test na całym zbiorze treningowym

Algorytm	**Q_COR**	**Q_ERR**	**Q_SSE**	**Q_MSE**	**Q_RMSE**	**Q_ARV**	**E_TTR**	**E_TCL**	**E_TOT**	**E_MEM**
SONN-3	100,00	0,00	20,18	0,11	0,34	13,16	55	0,06	65	124314
KNN(k=1)	100,00	0,00	0,00	0,00	0,00	NaN	1	0,56	100	24148
KNN(k=3)	67,98	32,02	16,67	0,09	0,31	1,60	0	0,57	101	24148
KNN(k=5)	65,17	34,83	23,88	0,13	0,37	1,94	0	0,72	129	24148
KNN(k=10)	71,35	28,65	27,96	0,16	0,40	2,19	0	1,02	181	24148
KNN(k=15)	61,80	38,20	31,68	0,18	0,42	2,27	0	1,30	231	24148
KNN(k=30)	65,73	34,27	32,42	0,18	0,43	2,68	0	2,38	423	24148
SVML(c=1;e=0,1)	33,15	66,85	178,00	1,00	1,00	∞	126	0,03	132	38246
SVML(c=10;e=0,1)	33,15	66,85	178,00	1,00	1,00	∞	123	0,00	123	38246
SVML(c=50;e=0,1)	33,15	66,85	178,00	1,00	1,00	∞	120	0,01	121	38246
SVML(c=100;e=0,1)	33,15	66,85	178,00	1,00	1,00	∞	121	0,01	122	38246
SVML(c=20;e=0,1)	33,15	66,85	178,00	1,00	1,00	∞	142	0,01	144	38246
SVML(c=20;e=0,01)	94,38	5,62	16,52	0,09	0,30	4,49	6481	0,01	6482	38246
SVML(c=20;e=0,001)	nieakceptowalny czas treningu									
SVMR(c=10;s=0,5;e=0,1)	100,00	0,00	0,24	0,00	0,04	1,82	507	0,33	565	37980
SVMR(c=10;s=1;e=0,1)	100,00	0,00	0,34	0,00	0,04	2,55	584	0,31	640	37980
SVMR(c=10;s=5;e=0,1)	100,00	0,00	0,20	0,00	0,03	2,73	609	0,30	663	37980
SVMR(c=10;s=10;e=0,1)	100,00	0,00	0,75	0,00	0,06	1,90	1912	0,25	1956	37980
SVMP(c=10;d=3;e=0,01)	33,15	66,85	178,00	1,00	1,00	∞	2317	0,06	2327	38326
SVMP(c=10;d=4;e=0,01)	33,15	66,85	178,00	1,00	1,00	∞	2322	0,06	2332	38326
SVMP(c=10;d=6;e=0,01)	33,15	66,85	178,00	1,00	1,00	∞	2325	0,06	2335	38326
RBFN(k=5;f=2;c=1)	68,54	31,46	36,24	0,20	0,45	4,22	15	0,01	17	23540
RBFN(k=10;f=2;c=1)	68,54	31,46	42,68	0,24	0,49	3,09	1	0,02	4	23685
RBFN(k=15;f=2;c=1)	24,16	75,84	100,50	0,56	0,75	7,17	2	0,02	6	23830
RBFN(k=20;f=2;c=1)	42,70	57,30	74,36	0,42	0,65	5,06	3	0,02	7	23975
RBFN(k=5;f=1;c=2)	84,83	15,17	22,23	0,12	0,35	2,91	4	0,01	6	24126
RBFN(k=10;f=1;c=2)	98,88	1,12	8,52	0,05	0,22	3,14	2	0,02	5	24841
RBFN(k=15;f=1;c=2)	99,44	0,56	8,20	0,05	0,21	2,77	4	0,02	8	25556
RBFN(k=20;f=1;c=2)	100,00	0,00	6,35	0,04	0,19	2,82	5	0,03	10	26271
RBFN(k=5;f=2;c=2)	69,10	30,90	36,80	0,21	0,45	4,20	2	0,00	2	23562
RBFN(k=10;f=2;c=2)	68,54	31,46	36,29	0,20	0,45	4,21	1	0,01	3	23707
RBFN(k=15;f=2;c=2)	70,79	29,21	37,84	0,21	0,46	4,07	2	0,02	5	23852
RBFN(k=20;f=2;c=2)	69,10	30,90	45,71	0,26	0,51	3,10	4	0,02	8	23997
PNN(sigma=0,1)	100,00	0,00	0,00	0,00	0,00	NaN	0	0,49	87	25533
PNN(sigma=1)	100,00	0,00	0,00	0,00	0,00	1,01	0	0,45	80	25533
PNN(sigma=10)	90,45	9,55	14,62	0,08	0,29	1,84	0	0,43	77	25533
PNN(sigma=100)	73,60	26,40	35,06	0,20	0,44	3,31	0	0,44	79	25533

Tablica 6.5: Wine, dane znormalizowane, test na całym zbiorze treningowym

Algorytm	Q_COR	Q_ERR	Q_SSE	Q_MSE	Q_RMSE	Q_ARV	E_TTR	E_TCL	E_TOT	E_MEM
SONN-3	100,00	0,00	20,18	0,11	0,34	13,16	63	0,05	72	124314
KNN(k=1)	100,00	0,00	0,00	0,00	0,00	NaN	1	0,53	95	24148
KNN(k=3)	96,63	3,37	2,78	0,02	0,12	1,05	0	0,60	107	24148
KNN(k=5)	93,26	6,74	3,60	0,02	0,14	1,11	0	0,71	127	24148
KNN(k=10)	92,13	7,87	4,56	0,03	0,16	1,20	0	1,11	198	24148
KNN(k=15)	88,20	11,80	4,94	0,03	0,17	1,27	0	1,38	245	24148
KNN(k=30)	81,46	18,54	6,34	0,04	0,19	1,49	0	2,42	430	24148
SVML(c=1;e=0,1)	99,44	0,56	9,23	0,05	0,23	3,75	31	0,03	36	38246
SVML(c=10;e=0,1)	99,44	0,56	9,88	0,06	0,24	4,23	67	0,01	68	38246
SVML(c=50;e=0,1)	100,00	0,00	9,87	0,06	0,24	4,30	102	0,01	103	38246
SVML(c=100;e=0,1)	100,00	0,00	10,10	0,06	0,24	4,38	78	0,01	79	38246
SVML(c=20;e=0,1)	100,00	0,00	9,44	0,05	0,23	4,21	94	0,01	96	38246
SVML(c=20;e=0,01)	100,00	0,00	9,69	0,05	0,23	4,57	225	0,01	226	38246
SVML(c=20;e=0,001)	100,00	0,00	9,56	0,05	0,23	5,02	290	0,01	291	38246
SVMR(c=10;s=0,5;e=0,1)	100,00	0,00	1,94	0,01	0,10	3,68	344	0,11	364	37980
SVMR(c=10;s=1;e=0,1)	100,00	0,00	6,40	0,04	0,19	4,90	680	0,08	694	37980
SVMR(c=10;s=5;e=0,1)	97,75	2,25	9,75	0,05	0,23	3,49	240	0,11	259	37980
SVMR(c=10;s=10;e=0,1)	98,88	1,12	10,71	0,06	0,25	3,59	472	0,16	500	37980
SVMP(c=10;d=3;e=0,01)	100,00	0,00	8,26	0,05	0,22	4,62	1145	0,06	1156	38326
SVMP(c=10;d=4;e=0,01)	100,00	0,00	8,57	0,05	0,22	4,49	1526	0,06	1537	38326
SVMP(c=10;d=6;e=0,01)	98,31	1,69	16,37	0,09	0,30	4,08	1394	0,10	1411	38326
RBFN(k=5;f=2;c=1)	95,51	4,49	14,86	0,08	0,29	3,55	15	0,01	17	23540
RBFN(k=10;f=2;c=1)	98,88	1,12	8,68	0,05	0,22	3,35	1	0,01	3	23685
RBFN(k=15;f=2;c=1)	100,00	0,00	6,78	0,04	0,20	3,18	2	0,01	4	23830
RBFN(k=20;f=2;c=1)	100,00	0,00	5,58	0,03	0,18	3,26	4	0,02	8	23975
RBFN(k=5;f=1;c=2)	98,88	1,12	8,37	0,05	0,22	3,11	6	0,00	6	24126
RBFN(k=10;f=1;c=2)	99,44	0,56	7,96	0,04	0,21	3,13	6	0,02	9	24841
RBFN(k=15;f=1;c=2)	99,44	0,56	6,53	0,04	0,19	3,01	13	0,02	16	25556
RBFN(k=20;f=1;c=2)	99,44	0,56	5,56	0,03	0,18	3,07	15	0,02	19	26271
RBFN(k=5;f=2;c=2)	98,88	1,12	11,45	0,06	0,25	4,08	1	0,01	2	23562
RBFN(k=10;f=2;c=2)	99,44	0,56	9,59	0,05	0,23	3,59	2	0,01	4	23707
RBFN(k=15;f=2;c=2)	100,00	0,00	6,86	0,04	0,20	3,24	2	0,02	5	23852
RBFN(k=20;f=2;c=2)	100,00	0,00	5,36	0,03	0,17	3,10	2	0,03	7	23997
PNN(sigma=0,1)	100,00	0,00	0,00	0,00	0,00	1,03	0	0,49	87	25533
PNN(sigma=1)	93,82	6,18	59,61	0,33	0,58	202,91	0	0,54	96	25533
PNN(sigma=10)	93,26	6,74	78,90	0,44	0,67	3,00E+06	0	0,51	90	25533
PNN(sigma=100)	93,26	6,74	79,11	0,44	0,67	3,01E+10	0	0,48	85	25533

Tablica 6.6: Wine, dane zestandaryzowane, test na całym zbiorze treningowym

Algorytm	Q_COR	Q_ERR	Q_SSE	Q_MSE	Q_RMSE	Q_ARV	E_TTR	E_TCL	E_TOT	E_MEM
SONN-3	100,00	0,00	20,18	0,11	0,34	13,16	60	0,05	69	124314
KNN(k=1)	100,00	0,00	0,00	0,00	0,00	NaN	1	0,60	107	24148
KNN(k=3)	75,84	24,16	10,44	0,06	0,24	1,38	0	0,57	101	24148
KNN(k=5)	76,40	23,60	14,04	0,08	0,28	1,55	0	0,72	128	24148
KNN(k=10)	71,35	28,65	17,98	0,10	0,32	1,89	0	1,02	181	24148
KNN(k=15)	67,98	32,02	21,52	0,12	0,35	2,30	0	1,29	229	24148
KNN(k=30)	56,74	43,26	28,50	0,16	0,40	3,91	0	2,34	416	24148
SVML(c=1;e=0,1)	92,70	7,30	20,55	0,12	0,34	5,64	272	0,03	278	38246
SVML(c=10;e=0,1)	96,07	3,93	14,86	0,08	0,29	4,61	993	0,01	994	38246
SVML(c=50;e=0,1)	95,51	4,49	17,63	0,10	0,31	3,70	2686	0,01	2688	38246
SVML(c=100;e=0,1)	94,94	5,06	15,05	0,08	0,29	4,29	1235	0,01	1236	38246
SVML(c=20;e=0,1)	97,75	2,25	14,13	0,08	0,28	3,88	1643	0,02	1646	38246
SVML(c=20;e=0,01)	98,88	1,12	12,27	0,07	0,26	4,74	1383	0,01	1385	38246
SVML(c=20;e=0,001)	99,44	0,56	11,00	0,06	0,25	4,46	3768	0,01	3770	38246
SVMR(c=10;s=0,5;e=0,1)	100,00	0,00	0,36	0,00	0,05	2,71	488	0,31	543	37980
SVMR(c=10;s=1;e=0,1)	100,00	0,00	0,33	0,00	0,04	3,27	364	0,30	418	37980
SVMR(c=10;s=5;e=0,1)	100,00	0,00	3,37	0,02	0,14	3,04	1905	0,13	1928	37980
SVMR(c=10;s=10;e=0,1)	98,31	1,69	9,04	0,05	0,23	3,10	1981	0,12	2002	37980
SVMP(c=10;d=3;e=0,01)	33,15	66,85	NaN	NaN	NaN	NaN	2824	0,06	2834	38326
SVMP(c=10;d=4;e=0,01)	33,15	66,85	NaN	NaN	NaN	NaN	2781	0,06	2791	38326
SVMP(c=10;d=6;e=0,01)	33,15	66,85	NaN	NaN	NaN	NaN	2854	0,06	2864	38326
RBFN(k=5;f=2;c=1)	84,83	15,17	28,98	0,16	0,40	3,39	15	0,02	18	23540
RBFN(k=10;f=2;c=1)	96,63	3,37	11,97	0,07	0,26	3,23	1	0,02	4	23685
RBFN(k=15;f=2;c=1)	98,88	1,12	9,82	0,06	0,23	3,51	2	0,01	4	23830
RBFN(k=20;f=2;c=1)	99,44	0,56	9,83	0,06	0,24	3,50	3	0,02	7	23975
RBFN(k=5;f=1;c=2)	90,45	9,55	19,88	0,11	0,33	3,56	4	0,01	5	24126
RBFN(k=10;f=1;c=2)	97,19	2,81	13,88	0,08	0,28	3,32	2	0,00	2	24841
RBFN(k=15;f=1;c=2)	98,31	1,69	10,39	0,06	0,24	3,10	4	0,02	7	25556
RBFN(k=20;f=1;c=2)	98,31	1,69	8,77	0,05	0,22	2,94	12	0,02	16	26271
RBFN(k=5;f=2;c=2)	81,46	18,54	33,69	0,19	0,44	3,97	2	0,01	4	23562
RBFN(k=10;f=2;c=2)	94,94	5,06	14,22	0,08	0,28	2,99	1	0,01	3	23707
RBFN(k=15;f=2;c=2)	98,31	1,69	9,83	0,06	0,24	3,31	1	0,02	5	23852
RBFN(k=20;f=2;c=2)	98,88	1,12	9,63	0,05	0,23	3,47	2	0,03	7	23997
PNN(sigma=0,1)	100,00	0,00	0,00	0,00	0,00	NaN	0	0,49	87	25533
PNN(sigma=1)	100,00	0,00	0,03	0,00	0,01	1,07	0	0,45	80	25533
PNN(sigma=10)	63,48	36,52	58,65	0,33	0,57	42,27	0	0,44	78	25533
PNN(sigma=100)	58,99	41,01	78,84	0,44	0,67	3,71E+05	0	0,48	85	25533

Tablica 6.7: Wine, zastosowanie algorytmu PCA i normalizacji danych, test na całym zbiorze treningowym

Algorytm	Q_COR	Q_ERR	Q_SSE	Q_MSE	Q_RMSE	Q_ARV	E_TTR	E_TCL	E_TOT	E_MEM
SONN-3	100,00	0,00	19,69	0,11	0,33	6,41	70	0,06	81	343926
KNN(k=1)	100,00	0,00	0,00	0,00	0,00	NaN	0	0,49	88	7060
KNN(k=3)	65,17	34,83	19,33	0,11	0,33	1,77	0	0,53	94	7060
KNN(k=5)	64,04	35,96	27,12	0,15	0,39	2,07	0	0,67	120	7060
KNN(k=10)	67,42	32,58	30,50	0,17	0,41	2,37	0	1,01	180	7060
KNN(k=15)	65,17	34,83	32,15	0,18	0,43	2,40	0	1,26	225	7060
KNN(k=30)	65,17	34,83	32,39	0,18	0,43	2,69	0	2,39	425	7060
SVML(c=1;e=0,1)	69,10	30,90	51,65	0,29	0,54	2,84	27	0,03	33	20870
SVML(c=10;e=0,1)	68,54	31,46	44,33	0,25	0,50	4,54	26	0,01	28	20870
SVML(c=50;e=0,1)	68,54	31,46	44,20	0,25	0,50	5,55	21	0,01	23	20870
SVML(c=100;e=0,1)	68,54	31,46	43,13	0,24	0,49	6,29	32	0,01	34	20870
SVML(c=20;e=0,1)	68,54	31,46	44,57	0,25	0,50	4,47	23	0,01	24	20870
SVML(c=20;e=0,01)	68,54	31,46	44,29	0,25	0,50	4,69	35	0,01	36	20870
SVML(c=20;e=0,001)	68,54	31,46	44,36	0,25	0,50	4,58	51	0,01	53	20870
SVMR(c=10;s=0,5;e=0,1)	71,35	28,65	35,91	0,20	0,45	3,30	528	0,11	547	20892
SVMR(c=10;s=1;e=0,1)	68,54	31,46	42,98	0,24	0,49	3,78	678	0,10	696	20892
SVMR(c=10;s=5;e=0,1)	67,42	32,58	51,88	0,29	0,54	2,34	389	0,12	411	20892
SVMR(c=10;s=10;e=0,1)	67,98	32,02	53,43	0,30	0,55	2,36	721	0,12	743	20892
SVMP(c=10;d=3;e=0,01)	65,73	34,27	54,95	0,31	0,56	2,22	527	0,11	547	21238
SVMP(c=10;d=4;e=0,01)	65,73	34,27	55,48	0,31	0,56	2,00	704	0,12	725	21238
SVMP(c=10;d=6;e=0,01)	65,73	34,27	49,66	0,28	0,53	1,96	686	0,13	710	21238
RBFN(k=5;f=2;c=1)	71,91	28,09	33,26	0,19	0,43	2,56	8	0,01	9	6452
RBFN(k=10;f=2;c=1)	73,60	26,40	31,29	0,18	0,42	2,52	1	0,01	2	6597
RBFN(k=15;f=2;c=1)	72,47	27,53	33,49	0,19	0,43	2,21	2	0,01	4	6742
RBFN(k=20;f=2;c=1)	72,47	27,53	32,23	0,18	0,43	2,34	2	0,02	5	6887
RBFN(k=5;f=1;c=2)	71,35	28,65	33,18	0,19	0,43	2,37	4	0,01	5	6558
RBFN(k=10;f=1;c=2)	70,22	29,78	33,08	0,19	0,43	2,36	4	0,01	5	6793
RBFN(k=15;f=1;c=2)	71,35	28,65	33,39	0,19	0,43	2,41	3	0,01	5	7028
RBFN(k=20;f=1;c=2)	72,47	27,53	33,04	0,19	0,43	2,39	5	0,01	7	7263
RBFN(k=5;f=2;c=2)	70,22	29,78	33,16	0,19	0,43	2,32	1	0,01	2	6474
RBFN(k=10;f=2;c=2)	72,47	27,53	31,89	0,18	0,42	2,27	1	0,01	3	6619
RBFN(k=15;f=2;c=2)	72,47	27,53	33,29	0,19	0,43	2,96	1	0,01	3	6764
RBFN(k=20;f=2;c=2)	71,35	28,65	33,26	0,19	0,43	2,29	2	0,01	3	6909
PNN(sigma=0,1)	73,60	26,40	37,57	0,21	0,46	4,14	0	0,37	66	8258
PNN(sigma=1)	73,03	26,97	76,46	0,43	0,66	5,03E+03	0	0,35	63	8258
PNN(sigma=10)	73,03	26,97	79,08	0,44	0,67	5,29E+07	0	0,36	64	8258
PNN(sigma=100)	73,03	26,97	79,11	0,44	0,67	5,29E+11	0	0,36	64	8258

Tablica 6.8: Wine, dane znormalizowane, 10-fold cross-validation, trening

Algorytm	Q_COR(avg)	Q_COR(min)	Q_COR(max)	Q_ERR(avg)	Q_ERR(min)	Q_ERR(max)	Q_SSE(avg)	Q_SSE(min)	Q_SSE(max)	Q_MSE(avg)	Q_MSE(min)	Q_MSE(max)	Q_RMSE(avg)	Q_RMSE(min)	Q_RMSE(max)	Q_ARV(avg)	Q_ARV(min)	Q_ARV(max)	E_TTR(avg)	E_TTR(min)	E_TTR(max)	E_TCL(avg)	E_TCL(min)	E_TCL(max)	E_TOT(avg)	E_TOT(min)	E_TOT(max)	E_MEM(avg)	E_MEM(min)	E_MEM(max)
SONN-3	100,00	100,00	100,00	0,00	0,00	0,00	15,82	11,57	18,97	0,10	0,07	0,12	0,31	0,27	0,34	11,56	4,39	25,01	15,20	13	21	0,01	0	1	17,40	13	24	112990,10	106354	118297
KNN(k=1)	100,00	100,00	100,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	NaN	NaN	NaN	0,00	0	0	0,48	0	38	76,60	69	106	21762,80	21736	21870
KNN(k=3)	96,01	95,63	96,88	4,00	3,13	4,38	2,56	1,89	2,89	0,02	0,01	0,02	0,13	0,11	0,13	1,06	1,05	1,07	0,10	0	1	0,53	0	4	85,60	83	89	21762,80	21736	21870
KNN(k=5)	92,82	90,06	95,63	7,18	4,38	9,94	3,42	2,28	4,36	0,02	0,01	0,03	0,15	0,12	0,17	1,12	1,09	1,14	0,10	0	1	0,66	0	4	105,20	102	110	21762,80	21736	21870
KNN(k=10)	91,13	88,75	93,79	8,87	6,21	11,25	4,05	2,99	4,91	0,03	0,02	0,03	0,16	0,14	0,18	1,22	1,19	1,25	0,00	0	0	0,95	0	6	152,80	150	158	21762,80	21736	21870
KNN(k=15)	87,76	84,38	90,68	12,24	9,32	15,63	4,46	3,31	5,21	0,03	0,02	0,03	0,17	0,14	0,18	1,29	1,26	1,32	0,00	0	0	1,24	1	7	198,30	189	207	21762,80	21736	21870
KNN(k=30)	83,08	79,38	84,47	16,92	15,53	20,63	5,94	5,01	6,52	0,04	0,03	0,04	0,19	0,18	0,20	1,55	1,50	1,62	0,00	0	0	2,19	2	6	351,30	340	370	21762,80	21736	21870
SVML(c=1;e=0,1)	99,06	98,13	100,00	0,94	0,00	1,88	8,35	8,02	8,71	0,05	0,05	0,05	0,23	0,22	0,23	3,80	3,51	4,30	19,10	15	28	0,01	0	2	20,00	15	28	34846,20	34808	34999
SVML(c=10;e=0,1)	99,88	99,38	100,00	0,13	0,00	0,63	8,21	7,63	8,61	0,05	0,05	0,05	0,23	0,22	0,23	4,48	3,88	4,95	54,30	46	61	0,01	0	1	55,50	47	63	34846,20	34808	34999
SVML(c=50;e=0,1)	99,88	99,38	100,00	0,13	0,00	0,63	8,45	7,59	8,87	0,05	0,05	0,06	0,23	0,22	0,24	4,46	3,90	5,00	77,30	59	101	0,01	0	1	78,20	60	101	34846,20	34808	34999
SVML(c=100;e=0,1)	99,94	99,38	100,00	0,06	0,00	0,63	8,91	7,76	9,92	0,06	0,05	0,06	0,24	0,22	0,25	4,90	4,34	5,66	86,20	67	110	0,01	0	7	87,90	68	110	34846,20	34808	34999
SVML(c=20;e=0,1)	100,00	100,00	100,00	0,00	0,00	0,00	8,49	7,85	9,50	0,05	0,05	0,06	0,23	0,22	0,24	4,49	4,10	4,77	76,70	47	106	0,01	0	1	77,80	49	106	34846,20	34808	34999
SVML(c=20;e=0,01)	99,81	99,38	100,00	0,19	0,00	0,63	8,67	8,06	9,79	0,05	0,05	0,06	0,23	0,22	0,25	4,57	4,10	5,32	119,10	78	179	0,01	0	1	119,90	80	180	34846,20	34808	34999
SVML(c=20;e=0,001)	99,94	99,38	100,00	0,06	0,00	0,63	8,55	7,90	9,55	0,05	0,05	0,06	0,23	0,22	0,24	4,64	4,19	5,28	327,00	170	442	0,01	0	1	328,10	171	443	34846,20	34808	34999
SVMR(c=10;s=0,5;e=0,1)	100,00	100,00	100,00	0,00	0,00	0,00	1,67	1,48	1,86	0,01	0,01	0,01	0,10	0,10	0,11	3,37	2,80	3,67	291,00	247	337	0,10	0	7	306,30	263	358	34580,20	34542	34733
SVMR(c=10;s=1;e=0,1)	100,00	100,00	100,00	0,00	0,00	0,00	5,51	4,91	6,06	0,03	0,03	0,04	0,19	0,18	0,19	4,39	4,08	4,86	344,50	275	409	0,07	0	1	356,40	286	421	34580,20	34542	34733
SVMR(c=10;s=5;e=0,1)	98,81	98,13	100,00	1,19	0,00	1,88	8,65	8,40	9,06	0,05	0,05	0,06	0,23	0,23	0,24	3,54	3,32	3,81	181,80	141	231	0,11	0	1	199,20	158	248	34580,20	34542	34733
SVMR(c=10;s=10;e=0,1)	98,56	97,50	100,00	1,44	0,00	2,50	9,71	9,42	10,14	0,06	0,06	0,06	0,25	0,24	0,25	3,54	3,34	3,70	430,70	323	567	0,16	0	3	456,00	347	592	34580,20	34542	34733
SVMP(c=10;d=3;e=0,01)	100,00	100,00	100,00	0,00	0,00	0,00	7,84	7,11	9,08	0,05	0,04	0,06	0,22	0,21	0,24	4,77	4,10	5,39	1044,10	718	1487	0,06	0	1	1053,40	728	1495	34926,20	34888	35079
SVMP(c=10;d=4;e=0,01)	99,94	99,38	100,00	0,06	0,00	0,63	8,04	6,49	10,06	0,05	0,04	0,06	0,22	0,20	0,25	4,69	4,05	6,09	1145,70	694	2077	0,06	0	3	1155,70	703	2087	34926,20	34888	35079
SVMP(c=10;d=6;e=0,01)	99,44	98,75	100,00	0,56	0,00	1,25	14,69	8,56	22,86	0,09	0,05	0,14	0,30	0,23	0,38	6,48	4,01	10,70	1318,50	737	1916	0,06	0	1	1327,70	746	1924	34926,20	34888	35079
RBFN(k=5;f=2;c=1)	97,75	95,03	99,38	2,25	0,63	4,97	10,37	9,55	12,55	0,06	0,06	0,08	0,25	0,24	0,28	3,73	3,19	3,85	0,40	0	1	0,01	0	1	1,80	1	3	21297,20	21272	21398
RBFN(k=10;f=2;c=1)	99,19	98,75	100,00	0,81	0,00	1,25	7,55	6,84	8,98	0,05	0,04	0,06	0,22	0,21	0,24	3,30	3,01	3,94	1,10	1	2	0,01	0	2	3,40	2	7	21442,20	21417	21543
RBFN(k=15;f=2;c=1)	99,75	99,38	100,00	0,25	0,00	0,63	6,29	5,84	6,91	0,04	0,04	0,04	0,20	0,19	0,21	3,24	2,99	3,65	2,00	2	2	0,02	0	2	4,90	4	8	21587,20	21562	21688
RBFN(k=20;f=2;c=1)	100,00	100,00	100,00	0,00	0,00	0,00	5,38	4,69	5,86	0,03	0,03	0,04	0,18	0,17	0,19	3,28	2,92	3,55	2,80	2	5	0,02	0	1	6,20	5	8	21732,20	21707	21833
RBFN(k=5;f=1;c=2)	98,56	97,50	100,00	1,44	0,00	2,50	8,26	7,47	9,18	0,05	0,05	0,06	0,23	0,22	0,24	3,30	3,07	3,56	3,10	2	6	0,01	0	1	4,30	3	7	21883,20	21858	21984
RBFN(k=10;f=1;c=2)	99,44	98,75	100,00	0,56	0,00	1,25	6,66	5,93	7,14	0,04	0,04	0,04	0,20	0,19	0,21	3,07	2,94	3,26	6,70	5	9	0,02	0	1	9,10	7	11	22598,20	22573	22699
RBFN(k=15;f=1;c=2)	99,69	99,38	100,00	0,31	0,00	0,63	6,03	5,56	6,51	0,04	0,03	0,04	0,19	0,19	0,20	3,17	2,99	3,35	8,40	7	9	0,02	0	1	10,90	9	12	23313,20	23288	23414
RBFN(k=20;f=1;c=2)	99,81	99,38	100,00	0,19	0,00	0,63	5,15	4,95	5,53	0,03	0,03	0,03	0,18	0,18	0,19	3,08	2,93	3,32	12,40	10	19	0,03	0	7	16,80	14	23	24028,20	24003	24129
RBFN(k=5;f=2;c=2)	96,94	95,00	100,00	3,06	0,00	5,00	11,39	9,83	13,39	0,07	0,06	0,08	0,27	0,25	0,29	3,69	2,99	4,29	0,70	0	1	0,00	0	1	1,40	1	2	21319,20	21294	21420
RBFN(k=10;f=2;c=2)	98,75	98,13	99,38	1,25	0,62	1,88	8,01	7,13	8,74	0,05	0,04	0,05	0,22	0,21	0,23	3,25	2,89	3,58	1,10	1	2	0,01	0	1	3,00	2	4	21464,20	21439	21565
RBFN(k=15;f=2;c=2)	99,88	99,38	100,00	0,12	0,00	0,63	6,03	4,90	6,77	0,04	0,03	0,04	0,19	0,18	0,21	3,15	2,93	3,48	1,70	1	2	0,02	0	1	4,10	2	5	21609,20	21584	21710
RBFN(k=20;f=2;c=2)	99,75	99,38	100,00	0,25	0,00	0,63	5,33	4,88	5,82	0,03	0,03	0,04	0,18	0,17	0,19	3,11	2,92	3,35	3,20	2	6	0,02	0	1	6,50	5	10	21754,20	21729	21855
PNN(sigma=0,1)	100,00	100,00	100,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,03	1,02	1,03	0,00	0	0	0,44	0	4	70,50	66	77	23219,00	23193	23323
PNN(sigma=1)	94,13	93,13	95,03	5,87	4,97	6,88	53,63	53,11	53,99	0,33	0,33	0,34	0,58	0,58	0,58	203,14	197,63	215,28	0,00	0	0	0,44	0	4	71,00	66	76	23219,00	23193	23323
PNN(sigma=10)	92,82	90,63	94,38	7,18	5,63	9,38	71,01	70,92	71,37	0,44	0,44	0,44	0,67	0,67	0,67	3,00E+06	2,90E+06	3,18E+06	0,00	0	0	0,44	0	4	70,80	66	76	23219,00	23193	23323
PNN(sigma=100)	92,82	90,63	94,38	7,18	5,63	9,38	71,20	71,11	71,55	0,44	0,44	0,44	0,67	0,67	0,67	3,02E+10	2,92E+10	3,20E+10	0,00	0	0	0,44	0	3	70,70	68	76	23219,00	23193	23323

Tablica 6.9: Wine, dane znormalizowane, 10-fold cross-validation, walidacja

Algorytm	Q_COR(avg)	Q_COR(min)	Q_COR(max)	Q_ERR(avg)	Q_ERR(min)	Q_ERR(max)	Q_SSE(avg)	Q_SSE(min)	Q_SSE(max)	Q_MSE(avg)	Q_MSE(min)	Q_MSE(max)	Q_RMSE(avg)	Q_RMSE(min)	Q_RMSE(max)	Q_ARV(avg)	Q_ARV(min)	Q_ARV(max)	E_TTR(avg)	E_TTR(min)	E_TTR(max)	E_TCL(avg)	E_TCL(min)	E_TCL(max)	E_TOT(avg)	E_TOT(min)	E_TOT(max)	E_MEM(avg)	E_MEM(min)	E_MEM(max)
SONN-3	91,08	77,78	100,00	8,92	0,00	22,22	2,48	1,36	4,05	0,14	0,08	0,22	0,37	0,27	0,47	8,34	2,96	33,62	15,20	13	21	0,00	0	0	15,20	13	21	112990,10	106354	118297
KNN(k=1)	95,52	83,33	100,00	4,48	0,00	16,67	0,80	0,00	3,00	0,04	0,00	0,17	0,15	0,00	0,41	NaN	NaN	1,20	0,00	0	0	0,43	0	1	7,60	6	8	21762,80	21736	21870
KNN(k=3)	93,27	88,24	100,00	6,73	0,00	11,76	0,60	0,11	2,00	0,03	0,01	0,11	0,16	0,08	0,33	1,11	1,06	1,19	0,10	0	1	0,52	0	1	9,40	7	11	21762,80	21736	21870
KNN(k=5)	88,73	70,59	100,00	11,27	0,00	29,41	0,62	0,08	2,12	0,04	0,00	0,12	0,17	0,07	0,34	1,18	1,06	1,32	0,10	0	1	0,62	0	1	11,20	10	12	21762,80	21736	21870
KNN(k=10)	90,39	76,47	100,00	9,61	0,00	23,53	0,55	0,04	1,38	0,03	0,00	0,08	0,16	0,05	0,28	1,31	1,18	1,49	0,00	0	0	0,97	0	4	17,30	16	20	21762,80	21736	21870
KNN(k=15)	88,79	83,33	94,44	11,21	5,56	16,67	0,52	0,09	1,62	0,03	0,01	0,09	0,16	0,07	0,30	1,39	1,22	1,56	0,00	0	0	1,24	1	4	22,10	20	25	21762,80	21736	21870
KNN(k=30)	84,87	77,78	94,44	15,13	5,56	22,22	0,67	0,17	1,46	0,04	0,01	0,08	0,19	0,10	0,28	1,70	1,32	2,12	0,00	0	0	2,19	2	4	38,90	37	40	21762,80	21736	21870
SVML(c=1;e=0,1)	97,71	88,24	100,00	2,29	0,00	11,76	1,12	0,62	1,91	0,06	0,03	0,11	0,25	0,19	0,34	3,79	2,46	4,65	19,10	15	28	0,01	0	1	19,20	15	29	34846,20	34808	34999
SVML(c=10;e=0,1)	98,30	94,12	100,00	1,70	0,00	5,88	1,11	0,67	1,85	0,06	0,04	0,11	0,25	0,19	0,33	4,06	2,13	4,79	54,30	46	61	0,00	0	0	54,30	46	61	34846,20	34808	34999
SVML(c=50;e=0,1)	97,75	88,89	100,00	2,25	0,00	11,11	1,12	0,54	1,90	0,06	0,03	0,11	0,25	0,17	0,33	4,14	2,37	5,38	77,30	59	101	0,02	0	1	77,70	60	101	34846,20	34808	34999
SVML(c=100;e=0,1)	97,75	88,89	100,00	2,25	0,00	11,11	1,26	0,65	1,89	0,07	0,04	0,11	0,26	0,19	0,33	4,91	2,15	7,40	86,20	67	110	0,01	0	1	86,40	67	110	34846,20	34808	34999
SVML(c=20;e=0,1)	97,75	88,89	100,00	2,25	0,00	11,11	1,13	0,59	1,97	0,06	0,03	0,12	0,25	0,18	0,34	4,04	2,55	5,31	76,70	47	106	0,00	0	0	76,70	47	106	34846,20	34808	34999
SVML(c=20;e=0,01)	98,30	94,12	100,00	1,70	0,00	5,88	1,18	0,78	1,93	0,07	0,04	0,11	0,26	0,21	0,34	4,36	2,46	6,30	119,10	78	179	0,00	0	0	119,10	78	179	34846,20	34808	34999
SVML(c=20;e=0,001)	98,30	94,12	100,00	1,70	0,00	5,88	1,16	0,56	1,93	0,07	0,03	0,11	0,25	0,18	0,34	4,77	2,44	5,73	327,00	170	442	0,00	0	0	327,00	170	442	34846,20	34808	34999
SVMR(c=10;s=0,5;e=0,1)	98,89	94,44	100,00	1,11	0,00	5,56	0,45	0,22	0,75	0,03	0,01	0,04	0,16	0,11	0,20	3,12	2,09	5,36	291,00	247	337	0,10	0	1	292,70	249	338	34580,20	34542	34733
SVMR(c=10;s=1;e=0,1)	97,75	94,12	100,00	2,25	0,00	5,88	0,90	0,42	1,87	0,05	0,02	0,11	0,22	0,15	0,33	3,75	2,18	4,51	344,50	275	409	0,16	0	17	347,40	276	410	34580,20	34542	34733
SVMR(c=10;s=5;e=0,1)	97,16	88,24	100,00	2,84	0,00	11,76	1,09	0,68	1,53	0,06	0,04	0,09	0,25	0,19	0,30	3,70	2,52	5,07	181,80	141	231	0,11	0	1	183,70	143	233	34580,20	34542	34733
SVMR(c=10;s=10;e=0,1)	98,30	88,89	100,00	1,70	0,00	11,11	1,14	0,70	1,47	0,06	0,04	0,08	0,25	0,20	0,29	3,98	2,62	6,94	430,70	323	567	0,16	0	3	433,50	325	570	34580,20	34542	34733
SVMP(c=10;d=3;e=0,01)	97,71	88,24	100,00	2,29	0,00	11,76	1,17	0,62	2,18	0,07	0,03	0,13	0,25	0,19	0,36	4,04	2,59	6,28	1044,10	718	1487	0,06	0	1	1045,20	719	1488	34926,20	34888	35079
SVMP(c=10;d=4;e=0,01)	97,75	94,12	100,00	2,25	0,00	5,88	1,16	0,99	1,55	0,07	0,06	0,09	0,26	0,23	0,29	4,49	3,11	8,16	1145,70	694	2077	0,06	0	1	1146,70	695	2078	34926,20	34888	35079
SVMP(c=10;d=6;e=0,01)	96,11	88,89	100,00	3,89	0,00	11,11	1,86	1,04	3,46	0,10	0,06	0,19	0,32	0,24	0,44	6,85	3,81	20,42	1318,50	737	1916	0,06	0	1	1319,50	738	1917	34926,20	34888	35079
RBFN(k=5;f=2;c=1)	97,78	94,44	100,00	2,22	0,00	5,56	1,23	0,85	1,69	0,07	0,05	0,09	0,26	0,22	0,31	4,43	3,59	5,83	0,40	0	1	0,01	0	1	0,60	0	2	21297,20	21272	21398
RBFN(k=10;f=2;c=1)	98,86	94,12	100,00	1,14	0,00	5,88	1,05	0,58	1,74	0,06	0,03	0,10	0,24	0,18	0,32	3,59	1,81	4,84	1,10	1	2	0,01	0	1	1,20	1	2	21442,20	21417	21543
RBFN(k=15;f=2;c=1)	99,44	94,44	100,00	0,56	0,00	5,56	0,79	0,43	0,98	0,04	0,02	0,05	0,21	0,15	0,23	3,28	2,39	4,21	2,00	2	2	0,00	0	0	2,00	2	2	21587,20	21562	21688
RBFN(k=20;f=2;c=1)	98,30	94,12	100,00	1,70	0,00	5,88	0,85	0,57	1,25	0,05	0,03	0,07	0,22	0,18	0,27	3,35	2,38	3,89	2,80	2	5	0,03	0	1	3,30	2	6	21732,20	21707	21833
RBFN(k=5;f=1;c=2)	98,86	94,12	100,00	1,14	0,00	5,88	0,96	0,51	1,21	0,05	0,03	0,07	0,23	0,17	0,26	3,68	2,64	5,43	3,10	2	6	0,02	0	1	3,40	2	6	21883,20	21858	21984
RBFN(k=10;f=1;c=2)	98,89	94,44	100,00	1,11	0,00	5,56	0,90	0,59	1,14	0,05	0,03	0,06	0,22	0,18	0,25	3,55	2,33	5,92	6,70	5	9	0,00	0	0	6,70	5	9	22598,20	22573	22699
RBFN(k=15;f=1;c=2)	98,89	94,44	100,00	1,11	0,00	5,56	0,83	0,38	1,02	0,05	0,02	0,06	0,21	0,15	0,24	3,54	2,41	4,95	8,40	7	9	0,01	0	1	8,50	8	9	23313,20	23288	23414
RBFN(k=20;f=1;c=2)	98,30	94,12	100,00	1,70	0,00	5,88	0,75	0,31	1,08	0,04	0,02	0,06	0,20	0,13	0,24	2,99	2,42	3,73	12,40	10	19	0,02	0	1	12,80	10	20	24028,20	24003	24129
RBFN(k=5;f=2;c=2)	96,11	88,89	100,00	3,89	0,00	11,11	1,36	0,95	1,77	0,08	0,06	0,10	0,28	0,24	0,31	4,13	2,78	5,45	0,70	0	1	0,01	0	1	0,90	0	2	21319,20	21294	21420
RBFN(k=10;f=2;c=2)	98,30	94,12	100,00	1,70	0,00	5,88	0,95	0,62	1,19	0,05	0,03	0,07	0,23	0,19	0,26	3,48	2,42	4,61	1,10	1	2	0,02	0	1	1,40	1	2	21464,20	21439	21565
RBFN(k=15;f=2;c=2)	99,41	94,12	100,00	0,59	0,00	5,88	0,79	0,46	1,22	0,04	0,03	0,07	0,21	0,16	0,26	3,49	2,17	5,58	1,70	1	2	0,02	0	1	2,00	1	3	21609,20	21584	21710
RBFN(k=20;f=2;c=2)	98,30	94,12	100,00	1,70	0,00	5,88	0,87	0,39	1,60	0,05	0,02	0,09	0,22	0,15	0,31	3,33	1,78	4,92	3,20	2	6	0,00	0	0	3,20	2	6	21754,20	21729	21855
PNN(sigma=0,1)	95,52	83,33	100,00	4,48	0,00	16,67	0,69	0,00	2,41	0,04	0,00	0,13	0,14	0,01	0,37	1,10	1,06	1,22	0,00	0	0	0,46	0	3	8,10	7	10	23219,00	23193	23323
PNN(sigma=1)	93,24	82,35	100,00	6,76	0,00	17,65	5,99	5,76	6,24	0,34	0,33	0,35	0,58	0,57	0,59	215,79	132,20	302,71	0,00	0	0	0,44	0	2	7,80	7	8	23219,00	23193	23323
PNN(sigma=10)	92,12	82,35	100,00	7,88	0,00	17,65	7,89	7,54	7,98	0,44	0,44	0,44	0,67	0,67	0,67	3,17E+06	2,05E+06	4,45E+06	0,00	0	0	0,46	0	3	8,20	7	10	23219,00	23193	23323
PNN(sigma=100)	92,12	82,35	100,00	7,88	0,00	17,65	7,91	7,56	8,00	0,44	0,44	0,44	0,67	0,67	0,67	3,18E+10	2,06E+10	4,47E+10	0,00	0	0	0,46	0	3	8,20	7	11	23219,00	23193	23323

6.3. Porównania dla zbioru Iris

Zbiór Iris jest uznawany za jeden z łatwiejszych dla algorytmów klasyfikacji. Mimo tego, analiza wyników uzyskanych dzięki wybranym metodom walidacji (tabele 6.10, 6.11 oraz 6.12) może dostarczyć ciekawych wniosków.

Warto zauważyć, że popularna metoda SVM z funkcją Gaussa jako funkcją radialną, która niewątpliwie jest bardzo popularnym narzędziem klasyfikacji, uzyskuje zadowalające wyniki tylko wtedy, gdy odpowiednio dobrane są parametry. W praktyce, jest to związane z wydłużonym czasem poszukiwań najlepszego klasyfikatora dla danego problemu, ponieważ konieczne jest sprawdzenie kilku różnych konfiguracji parametrów. Bardzo często wiąże się to z doświadczeniem i wiedzą ekspercką o konkretnej dziedzinie i własnościach stosowanej metody klasyfikacji.

Tablica 6.10: Iris, dane znormalizowane, test na całym zbiorze treningowym

Algorytm	Q_COR	Q_ERR	Q_SSE	Q_MSE	Q_RMSE	Q_ARV	E_TTR	E_TCL	E_TOT	E_MEM
SONN-3	100,00	0,00	19,28	0,13	0,36	6,28	53	0,13	73	130505
KNN(k=1)	100,00	0,00	0,00	0,00	0,00	NaN	2	0,76	116	11446
KNN(k=3)	95,33	4,67	2,67	0,02	0,13	1,06	0	0,48	72	11446
KNN(k=5)	93,33	6,67	3,80	0,03	0,16	1,09	0	0,60	90	11446
KNN(k=10)	95,33	4,67	2,90	0,02	0,14	1,15	0	0,87	130	11446
KNN(k=15)	90,67	9,33	3,56	0,02	0,15	1,21	0	1,16	174	11446
KNN(k=30)	86,00	14,00	5,81	0,04	0,20	1,47	0	2,15	323	11446
SVML(c=1;e=0,1)	86,67	13,33	23,71	0,16	0,40	8,02	28	0,12	46	21919
SVML(c=10;e=0,1)	94,00	6,00	20,79	0,14	0,37	7,94	23	0,01	24	21919
SVML(c=50;e=0,1)	94,67	5,33	20,58	0,14	0,37	8,25	111	0,00	111	21919
SVML(c=100;e=0,1)	94,67	5,33	21,01	0,14	0,37	8,88	140	0,01	141	21919
SVML(c=20;e=0,1)	94,67	5,33	20,87	0,14	0,37	9,82	27	0,01	29	21919
SVML(c=20;e=0,01)	94,67	5,33	20,71	0,14	0,37	9,33	55	0,00	55	21919
SVML(c=20;e=0,001)	94,67	5,33	20,70	0,14	0,37	8,80	123	0,00	123	21919
SVMR(c=10;s=0,5;e=0,1)	97,33	2,67	6,29	0,04	0,20	3,50	191	0,09	204	21869
SVMR(c=10;s=1;e=0,1)	97,33	2,67	11,82	0,08	0,28	4,59	166	0,09	179	21869
SVMR(c=10;s=5;e=0,1)	82,00	18,00	24,91	0,17	0,41	7,58	328	0,11	345	21869
SVMR(c=10;s=10;e=0,1)	66,67	33,33	48,10	0,32	0,57	2,62	371	0,12	389	21869
SVMP(c=10;d=3;e=0,01)	95,33	4,67	22,11	0,15	0,38	11,43	2059	0,10	2074	22215
SVMP(c=10;d=4;e=0,01)	97,33	2,67	23,20	0,15	0,39	11,86	2543	0,09	2557	22215
SVMP(c=10;d=6;e=0,01)	96,67	3,33	25,85	0,17	0,42	13,31	6674	0,10	6689	22215
RBFN(k=5;f=2;c=1)	96,00	4,00	9,07	0,06	0,25	3,22	15	0,01	17	11062
RBFN(k=10;f=2;c=1)	98,00	2,00	4,90	0,03	0,18	2,29	1	0,01	3	11207
RBFN(k=15;f=2;c=1)	98,00	2,00	5,48	0,04	0,19	2,46	1	0,01	3	11352
RBFN(k=20;f=2;c=1)	98,67	1,33	4,62	0,03	0,18	2,26	2	0,01	4	11497
RBFN(k=5;f=1;c=2)	95,33	4,67	8,68	0,06	0,24	3,26	5	0,01	6	11288
RBFN(k=10;f=1;c=2)	97,33	2,67	6,95	0,05	0,22	2,65	3	0,01	5	11643
RBFN(k=15;f=1;c=2)	98,00	2,00	6,39	0,04	0,21	2,55	3	0,01	5	11998
RBFN(k=20;f=1;c=2)	98,00	2,00	4,57	0,03	0,17	2,11	5	0,01	7	12353
RBFN(k=5;f=2;c=2)	94,67	5,33	9,66	0,06	0,25	3,14	1	0,01	2	11084
RBFN(k=10;f=2;c=2)	98,67	1,33	6,34	0,04	0,21	2,57	1	0,01	3	11229
RBFN(k=15;f=2;c=2)	98,67	1,33	6,10	0,04	0,20	2,56	1	0,01	3	11374
RBFN(k=20;f=2;c=2)	98,67	1,33	4,87	0,03	0,18	2,45	2	0,01	3	11519
PNN(sigma=0,1)	97,33	2,67	2,93	0,02	0,14	1,30	0	0,39	59	12816
PNN(sigma=1)	91,33	8,67	53,12	0,35	0,60	375,87	0	0,38	57	12816
PNN(sigma=10)	91,33	8,67	66,52	0,44	0,67	4,76E+06	0	0,39	58	12816
PNN(sigma=100)	91,33	8,67	66,67	0,44	0,67	4,78E+10	0	0,40	60	12816

Tablica 6.11: Iris, dane znormalizowane, 10-fold cross-validation, trening

Algorytm	Q_COR(avg)	Q_COR(min)	Q_COR(max)	Q_ERR(avg)	Q_ERR(min)	Q_ERR(max)	Q_SSE(avg)	Q_SSE(min)	Q_SSE(max)	Q_MSE(avg)	Q_MSE(min)	Q_MSE(max)	Q_RMSE(avg)	Q_RMSE(min)	Q_RMSE(max)	Q_ARV(avg)	Q_ARV(min)	Q_ARV(max)	E_TTR(avg)	E_TTR(min)	E_TTR(max)	E_TCL(avg)	E_TCL(min)	E_TCL(max)	E_TOT(avg)	E_TOT(min)	E_TOT(max)	E_MEM(avg)	E_MEM(min)	E_MEM(max)
SONN-3	99,85	99,26	100,00	0,15	0,00	0,74	16,42	13,94	18,97	0,12	0,10	0,14	0,35	0,32	0,37	7,68	4,13	21,28	7,60	7	9	0,01	0,00	1,00	9,60	7	12	107467,70	91465	129815
KNN(k=1)	100,00	100,00	100,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	NaN	NaN	NaN	0,10	0	1	0,36	0,00	3,00	48,80	45	55	10331,00	10331	10331
KNN(k=3)	95,04	94,07	96,30	4,96	3,70	5,93	2,49	1,78	2,89	0,02	0,01	0,02	0,14	0,11	0,15	1,06	1,05	1,08	0,00	0	0	0,42	0,00	8,00	57,30	54	67	10331,00	10331	10331
KNN(k=5)	93,11	89,63	95,56	6,89	4,44	10,37	3,31	2,36	4,20	0,02	0,02	0,03	0,16	0,13	0,18	1,10	1,08	1,15	0,00	0	0	0,55	0,00	5,00	74,60	73	79	10331,00	10331	10331
KNN(k=10)	93,41	89,63	97,78	6,59	2,22	10,37	2,79	2,10	3,34	0,02	0,02	0,02	0,14	0,12	0,16	1,16	1,13	1,19	0,00	0	0	0,79	0,00	4,00	106,60	103	109	10331,00	10331	10331
KNN(k=15)	89,48	84,44	93,33	10,52	6,67	15,56	3,37	2,75	3,98	0,03	0,02	0,03	0,16	0,14	0,17	1,24	1,21	1,26	0,00	0	0	1,04	0,00	14,00	140,20	134	155	10331,00	10331	10331
KNN(k=30)	78,44	74,07	81,48	21,56	18,52	25,93	5,62	4,88	6,06	0,04	0,04	0,04	0,20	0,19	0,21	1,56	1,48	1,61	0,00	0	0	1,91	1,00	23,00	258,40	251	274	10331,00	10331	10331
SVML(c=1;e=0,1)	85,70	82,22	86,67	14,30	13,33	17,78	21,19	20,08	23,33	0,16	0,15	0,17	0,40	0,39	0,42	8,02	6,68	10,32	13,00	9	15	0,01	0,00	1,00	14,10	11	17	20134,00	20134	20134
SVML(c=10;e=0,1)	93,93	92,59	95,56	6,07	4,44	7,41	18,89	18,26	19,29	0,14	0,14	0,14	0,37	0,37	0,38	9,54	5,81	11,44	18,40	14	22	0,01	0,00	1,00	19,50	15	24	20134,00	20134	20134
SVML(c=100;e=0,1)	94,52	92,59	97,04	5,48	2,96	7,41	19,11	18,45	20,74	0,14	0,14	0,15	0,38	0,37	0,39	7,50	5,13	11,03	142,40	109	189	0,00	0,00	1,00	143,00	110	190	20134,00	20134	20134
SVML(c=50;e=0,1)	94,59	93,33	96,30	5,41	3,70	6,67	18,68	18,18	19,59	0,14	0,13	0,15	0,37	0,37	0,38	7,57	5,21	9,21	62,20	39	94	0,01	0,00	1,00	63,20	41	95	20134,00	20134	20134
SVML(c=20;e=0,1)	94,22	92,59	94,81	5,78	5,19	7,41	18,83	18,48	19,39	0,14	0,14	0,14	0,37	0,37	0,38	7,95	5,30	10,12	29,40	25	35	0,01	0,00	1,00	30,70	25	37	20134,00	20134	20134
SVML(c=20;e=0,01)	94,22	92,59	95,56	5,78	4,44	7,41	18,65	18,28	19,52	0,14	0,14	0,14	0,37	0,37	0,38	8,40	5,71	10,21	42,70	34	56	0,01	0,00	1,00	43,70	34	57	20134,00	20134	20134
SVML(c=20;e=0,001)	94,44	92,59	95,56	5,56	4,44	7,41	18,71	18,23	19,53	0,14	0,14	0,14	0,37	0,37	0,38	8,34	6,02	9,90	96,80	64	157	0,01	0,00	1,00	98,00	65	159	20134,00	20134	20134
SVMR(c=10;s=0,5;e=0,1)	97,41	97,04	98,52	2,59	1,48	2,96	5,97	5,31	6,80	0,04	0,04	0,05	0,21	0,20	0,22	3,58	3,03	4,48	184,60	141	271	0,07	0,00	3,00	194,20	153	278	20084,00	20084	20084
SVMR(c=10;s=1;e=0,1)	97,78	97,04	99,26	2,22	0,74	2,96	10,95	10,38	11,64	0,08	0,08	0,09	0,28	0,28	0,29	4,62	4,40	4,85	149,80	92	229	0,08	0,00	1,00	160,10	102	239	20084,00	20084	20084
SVMR(c=10;s=5;e=0,1)	78,22	74,81	82,96	21,78	17,04	25,19	23,98	22,03	25,46	0,18	0,16	0,19	0,42	0,40	0,43	6,60	5,90	7,69	174,50	121	215	0,11	0,00	3,00	189,00	139	229	20084,00	20084	20084
SVMR(c=10;s=10;e=0,1)	66,67	66,67	66,67	33,33	33,33	33,33	44,46	43,38	45,19	0,33	0,32	0,33	0,57	0,57	0,58	2,48	2,25	2,66	260,10	234	376	0,12	0,00	4,00	276,10	249	392	20084,00	20084	20084
SVMP(c=10;d=3;e=0,01)	96,22	95,56	97,04	3,78	2,96	4,44	19,34	18,06	20,37	0,14	0,13	0,15	0,38	0,37	0,39	10,17	8,94	12,29	1521,10	857	2489	0,08	0,00	1,00	1531,80	868	2500	20430,00	20430	20430
SVMP(c=10;d=4;e=0,01)	97,11	96,30	98,52	2,89	1,48	3,70	20,87	20,51	21,12	0,15	0,15	0,16	0,39	0,39	0,40	11,08	9,90	12,51	3156,70	1619	6330	0,09	0,00	1,00	3168,30	1631	6342	20430,00	20430	20430
SVMP(c=10;d=6;e=0,01)	97,04	94,07	100,00	2,96	0,00	5,93	23,15	22,22	24,08	0,17	0,16	0,18	0,41	0,41	0,42	13,60	12,10	15,27	5558,10	2214	11040	0,09	0,00	1,00	5570,50	2225	11054	20430,00	20430	20430
RBFN(k=5;f=2;c=1)	95,04	93,33	97,04	4,96	2,96	6,67	7,77	6,35	9,09	0,06	0,05	0,07	0,24	0,22	0,26	2,89	2,55	3,21	0,70	0	1	0,00	0,00	1,00	1,30	1	2	10067,00	10067	10067
RBFN(k=10;f=2;c=1)	98,30	97,04	100,00	1,70	0,00	2,96	5,22	4,14	5,85	0,04	0,03	0,04	0,20	0,18	0,21	2,49	2,29	2,62	1,60	1	7	0,01	0,00	1,00	2,90	1	8	10212,00	10212	10212
RBFN(k=15;f=2;c=1)	98,22	97,78	100,00	1,78	0,00	2,22	4,66	4,11	5,16	0,03	0,03	0,04	0,19	0,17	0,20	2,40	2,20	2,57	3,60	1	24	0,01	0,00	1,00	5,00	2	26	10357,00	10357	10357
RBFN(k=20;f=2;c=1)	98,30	97,78	100,00	1,70	0,00	2,22	4,05	3,80	4,70	0,03	0,03	0,03	0,17	0,17	0,19	2,22	1,97	2,56	2,00	1	4	0,01	0,00	1,00	3,80	1	6	10502,00	10502	10502
RBFN(k=5;f=1;c=2)	94,81	92,59	97,04	5,19	2,96	7,41	8,68	7,89	9,79	0,06	0,06	0,07	0,25	0,24	0,27	3,25	2,80	3,77	1,60	1	3	0,01	0,00	1,00	2,60	2	4	10293,00	10293	10293
RBFN(k=10;f=1;c=2)	97,48	95,56	98,52	2,52	1,48	4,44	6,33	5,47	7,21	0,05	0,04	0,05	0,22	0,20	0,23	2,67	2,56	2,85	2,50	2	5	0,01	0,00	1,00	4,10	3	7	10648,00	10648	10648
RBFN(k=15;f=1;c=2)	98,52	97,78	100,00	1,48	0,00	2,22	5,11	4,38	5,58	0,04	0,03	0,04	0,19	0,18	0,20	2,60	2,49	2,77	4,60	4	6	0,01	0,00	1,00	6,50	5	8	11003,00	11003	11003
RBFN(k=20;f=1;c=2)	98,52	97,78	100,00	1,48	0,00	2,22	4,18	3,85	4,55	0,03	0,03	0,03	0,18	0,17	0,18	2,28	2,12	2,47	6,30	5	9	0,02	0,00	3,00	8,70	6	11	11358,00	11358	11358
RBFN(k=5;f=2;c=2)	96,07	94,07	97,78	3,93	2,22	5,93	7,85	7,03	8,88	0,06	0,05	0,07	0,24	0,23	0,26	3,00	2,73	3,31	0,40	0	1	0,01	0,00	1,00	1,30	0	2	10089,00	10089	10089
RBFN(k=10;f=2;c=2)	98,22	97,04	100,00	1,78	0,00	2,96	5,26	4,75	6,25	0,04	0,04	0,05	0,20	0,19	0,22	2,59	2,44	2,67	1,00	0	3	0,01	0,00	1,00	2,20	1	5	10234,00	10234	10234
RBFN(k=15;f=2;c=2)	98,22	97,04	99,26	1,78	0,74	2,96	4,91	4,38	5,47	0,04	0,03	0,04	0,19	0,18	0,20	2,47	2,26	2,87	1,40	1	2	0,01	0,00	1,00	3,10	1	5	10379,00	10379	10379
RBFN(k=20;f=2;c=2)	98,59	97,78	99,26	1,41	0,74	2,22	3,97	3,17	4,47	0,03	0,02	0,03	0,17	0,15	0,18	2,20	1,85	2,55	1,50	1	2	0,02	0,00	1,00	3,60	2	5	10524,00	10524	10524
PNN(sigma=0,1)	97,85	96,30	99,26	2,15	0,74	3,70	2,57	1,91	2,99	0,02	0,01	0,02	0,14	0,12	0,15	1,30	1,25	1,33	0,00	0	0	0,35	0,00	4,00	47,30	42	52	11761,00	11761	11761
PNN(sigma=1)	91,33	88,89	92,59	8,67	7,41	11,11	47,81	47,51	48,10	0,35	0,35	0,36	0,60	0,59	0,60	376,26	359,92	395,96	0,00	0	0	0,38	0,00	3,00	50,80	47	65	11761,00	11761	11761
PNN(sigma=10)	91,04	88,89	91,85	8,96	8,15	11,11	59,87	59,86	59,87	0,44	0,44	0,44	0,67	0,67	0,67	4,77E+06	4,59E+06	4,98E+06	0,00	0	0	0,36	0,00	3,00	49,20	43	53	11761,00	11761	11761
PNN(sigma=100)	91,04	88,89	91,85	8,96	8,15	11,11	60,00	60,00	60,00	0,44	0,44	0,44	0,67	0,67	0,67	4,78E+10	4,60E+10	4,99E+10	0,00	0	0	0,37	0,00	32,00	50,40	46	77	11761,00	11761	11761

Tablica 6.12: Iris, dane znormalizowane, 10-fold cross-validation, walidacja

Algorytm	Q_COR(avg)	Q_COR(min)	Q_COR(max)	Q_ERR(avg)	Q_ERR(min)	Q_ERR(max)	Q_SSE(avg)	Q_SSE(min)	Q_SSE(max)	Q_MSE(avg)	Q_MSE(min)	Q_MSE(max)	Q_RMSE(avg)	Q_RMSE(min)	Q_RMSE(max)	Q_ARV(avg)	Q_ARV(min)	Q_ARV(max)	E_TTR(avg)	E_TTR(min)	E_TTR(max)	E_TCL(avg)	E_TCL(min)	E_TCL(max)	E_TOT(avg)	E_TOT(min)	E_TOT(max)	E_MEM(avg)	E_MEM(min)	E_MEM(max)
SONN-3	91,33	80,00	100,00	8,67	0,00	20,00	2,14	1,56	3,96	0,14	0,10	0,26	0,37	0,32	0,51	6,86	2,98	16,89	7,60	7	9	0,01	0	1	7,70	7	9	107467,70	91465	129815
KNN(k=1)	95,33	86,67	100,00	4,67	0,00	13,33	0,70	0,00	2,00	0,05	0,00	0,13	0,17	0,00	0,37	NaN	NaN	1,15	0,10	0	1	0,45	0	14	6,90	5	19	10331,00	10331	10331
KNN(k=3)	94,67	86,67	100,00	5,33	0,00	13,33	0,54	0,00	1,11	0,04	0,00	0,07	0,16	0,00	0,27	NaN	NaN	1,15	0,00	0	0	0,42	0	3	6,30	5	9	10331,00	10331	10331
KNN(k=5)	94,67	80,00	100,00	5,33	0,00	20,00	0,42	0,00	1,00	0,03	0,00	0,07	0,15	0,00	0,26	NaN	NaN	1,27	0,00	0	0	0,57	0	3	8,50	7	10	10331,00	10331	10331
KNN(k=10)	93,33	80,00	100,00	6,67	0,00	20,00	0,38	0,06	1,18	0,03	0,00	0,08	0,15	0,06	0,28	1,19	1,07	1,34	0,00	0	0	0,79	0	2	11,80	11	14	10331,00	10331	10331
KNN(k=15)	88,67	80,00	93,33	11,33	6,67	20,00	0,39	0,07	0,88	0,03	0,00	0,06	0,15	0,07	0,24	1,29	1,21	1,42	0,00	0	0	1,03	0	2	15,40	14	16	10331,00	10331	10331
KNN(k=30)	82,67	73,33	100,00	17,33	0,00	26,67	0,64	0,34	1,31	0,04	0,02	0,09	0,20	0,15	0,30	1,64	1,37	2,05	0,00	0	0	1,91	1	5	28,60	26	31	10331,00	10331	10331
SVML(c=1;e=0,1)	85,33	66,67	100,00	14,67	0,00	33,33	2,40	1,65	3,14	0,16	0,11	0,21	0,40	0,33	0,46	10,99	5,40	24,91	13,00	9	15	0,01	0	1	13,20	9	15	20134,00	20134	20134
SVML(c=10;e=0,1)	91,33	80,00	100,00	8,67	0,00	20,00	2,16	1,91	2,76	0,14	0,13	0,18	0,38	0,36	0,43	11,10	5,80	20,16	18,40	14	22	0,00	0	0	18,40	14	22	20134,00	20134	20134
SVML(c=50;e=0,1)	92,67	86,67	100,00	7,33	0,00	13,33	2,30	2,00	2,69	0,15	0,13	0,18	0,39	0,36	0,42	8,96	4,14	21,74	142,40	109	189	0,01	0	1	142,50	109	189	20134,00	20134	20134
SVML(c=100;e=0,1)	92,67	86,67	100,00	7,33	0,00	13,33	2,22	2,02	2,60	0,15	0,13	0,17	0,38	0,37	0,42	9,99	4,12	19,98	62,20	39	94	0,00	0	0	62,20	39	94	20134,00	20134	20134
SVML(c=20;e=0,1)	92,00	86,67	100,00	8,00	0,00	13,33	2,11	1,65	2,72	0,14	0,11	0,18	0,37	0,33	0,43	9,55	4,88	18,08	29,40	25	35	0,01	0	1	29,50	25	35	20134,00	20134	20134
SVML(c=20;e=0,01)	92,00	86,67	100,00	8,00	0,00	13,33	2,18	1,83	2,76	0,15	0,12	0,18	0,38	0,35	0,43	9,29	3,70	16,00	42,70	34	56	0,01	0	1	42,80	34	57	20134,00	20134	20134
SVML(c=20;e=0,001)	92,67	80,00	100,00	7,33	0,00	20,00	2,16	1,74	3,00	0,14	0,12	0,20	0,38	0,34	0,45	9,18	5,02	16,50	96,80	64	157	0,00	0	0	96,80	64	157	20134,00	20134	20134
SVMR(c=10;s=0,5;e=0,1)	96,00	93,33	100,00	4,00	0,00	6,67	0,76	0,54	1,38	0,05	0,04	0,09	0,22	0,19	0,30	5,04	2,44	10,81	184,60	141	271	0,07	0	1	185,70	142	272	20084,00	20084	20084
SVMR(c=10;s=1;e=0,1)	96,67	86,67	100,00	3,33	0,00	13,33	1,26	0,89	1,75	0,08	0,06	0,12	0,29	0,24	0,34	4,97	3,07	9,91	149,80	92	229	0,07	0	1	150,90	93	231	20084,00	20084	20084
SVMR(c=10;s=5;e=0,1)	76,67	66,67	93,33	23,33	6,67	33,33	2,79	1,81	3,88	0,19	0,12	0,26	0,43	0,35	0,51	8,97	4,32	27,71	174,50	121	215	0,11	0	1	176,20	122	216	20084,00	20084	20084
SVMR(c=10;s=10;e=0,1)	66,67	66,67	66,67	33,33	33,33	33,33	4,88	4,20	5,44	0,33	0,28	0,36	0,57	0,53	0,60	2,54	2,32	2,91	260,10	234	376	0,11	0	1	261,80	236	378	20084,00	20084	20084
SVMP(c=10;d=3;e=0,01)	96,00	86,67	100,00	4,00	0,00	13,33	2,19	1,94	2,49	0,15	0,13	0,17	0,38	0,36	0,41	12,07	7,29	20,83	1521,10	857	2489	0,07	0	1	1522,20	858	2490	20430,00	20430	20430
SVMP(c=10;d=4;e=0,01)	94,67	86,67	100,00	5,33	0,00	13,33	2,35	1,97	2,73	0,16	0,13	0,18	0,40	0,36	0,43	13,29	4,78	23,66	3156,70	1619	6330	0,09	0	3	3158,10	1620	6331	20430,00	20430	20430
SVMP(c=10;d=6;e=0,01)	94,00	80,00	100,00	6,00	0,00	20,00	2,62	2,30	2,93	0,17	0,15	0,20	0,42	0,39	0,44	19,28	7,27	55,90	5558,10	2214	11040	0,08	0	1	5559,30	2216	11041	20430,00	20430	20430
RBFN(k=5;f=2;c=1)	96,00	86,67	100,00	4,00	0,00	13,33	0,95	0,50	1,49	0,06	0,03	0,10	0,25	0,18	0,32	3,20	1,94	4,40	0,70	0	1	0,00	0	0	0,70	0	1	10067,00	10067	10067
RBFN(k=10;f=2;c=1)	96,00	86,67	100,00	4,00	0,00	13,33	0,73	0,27	1,35	0,05	0,02	0,09	0,22	0,14	0,30	2,61	1,85	3,58	1,60	1	7	0,00	0	0	1,60	1	7	10212,00	10212	10212
RBFN(k=15;f=2;c=1)	97,33	86,67	100,00	2,67	0,00	13,33	0,65	0,28	1,20	0,04	0,02	0,08	0,20	0,14	0,28	2,70	2,00	3,91	3,60	1	24	0,01	0	1	3,80	1	24	10357,00	10357	10357
RBFN(k=20;f=2;c=1)	98,00	86,67	100,00	2,00	0,00	13,33	0,60	0,23	1,38	0,04	0,02	0,09	0,19	0,12	0,30	2,70	2,06	4,61	2,00	1	4	0,03	0	1	2,40	1	4	10502,00	10502	10502
RBFN(k=5;f=1;c=2)	94,67	86,67	100,00	5,33	0,00	13,33	1,04	0,66	1,68	0,07	0,04	0,11	0,26	0,21	0,33	3,60	2,14	5,01	1,60	1	3	0,00	0	0	1,60	1	3	10293,00	10293	10293
RBFN(k=10;f=1;c=2)	94,67	86,67	100,00	5,33	0,00	13,33	0,84	0,49	1,53	0,06	0,03	0,10	0,23	0,18	0,32	3,05	2,02	4,98	2,50	2	5	0,01	0	1	2,70	2	5	10648,00	10648	10648
RBFN(k=15;f=1;c=2)	97,33	93,33	100,00	2,67	0,00	6,67	0,70	0,31	1,36	0,05	0,02	0,09	0,21	0,14	0,30	2,95	1,96	5,77	4,60	4	6	0,00	0	0	4,60	4	6	11003,00	11003	11003
RBFN(k=20;f=1;c=2)	96,67	86,67	100,00	3,33	0,00	13,33	0,60	0,29	1,23	0,04	0,02	0,08	0,20	0,14	0,29	2,66	1,91	4,95	6,30	5	9	0,01	0	1	6,40	5	9	11358,00	11358	11358
RBFN(k=5;f=2;c=2)	96,00	86,67	100,00	4,00	0,00	13,33	0,93	0,41	1,60	0,06	0,03	0,11	0,24	0,17	0,33	3,36	1,82	5,10	0,40	0	1	0,01	0	1	0,60	0	2	10089,00	10089	10089
RBFN(k=10;f=2;c=2)	98,00	86,67	100,00	2,00	0,00	13,33	0,68	0,30	1,15	0,05	0,02	0,08	0,21	0,14	0,28	3,04	2,22	4,42	1,00	0	3	0,01	0	1	1,10	0	3	10234,00	10234	10234
RBFN(k=15;f=2;c=2)	98,00	86,67	100,00	2,00	0,00	13,33	0,69	0,28	1,39	0,05	0,02	0,09	0,21	0,14	0,30	2,78	2,18	3,39	1,40	1	2	0,01	0	1	1,50	1	3	10379,00	10379	10379
RBFN(k=20;f=2;c=2)	97,33	86,67	100,00	2,67	0,00	13,33	0,59	0,22	1,15	0,04	0,02	0,08	0,19	0,12	0,28	2,29	1,91	2,93	1,50	1	2	0,03	0	1	1,90	1	3	10524,00	10524	10524
PNN(sigma=0,1)	96,00	86,67	100,00	4,00	0,00	13,33	0,45	0,05	0,94	0,03	0,00	0,06	0,16	0,06	0,25	1,36	1,13	1,75	0,00	0	0	0,34	0	1	5,10	5	6	11761,00	11761	11761
PNN(sigma=1)	90,67	80,00	100,00	9,33	0,00	20,00	5,32	5,18	5,46	0,35	0,35	0,36	0,60	0,59	0,60	389,89	310,82	502,42	0,00	0	0	0,35	0	1	5,20	5	6	11761,00	11761	11761
PNN(sigma=10)	90,67	80,00	100,00	9,33	0,00	20,00	6,65	6,65	6,65	0,44	0,44	0,44	0,67	0,67	0,67	4,93E+06	4,13E+06	6,29E+06	0,00	0	0	0,38	0	3	5,70	4	8	11761,00	11761	11761
PNN(sigma=100)	90,67	80,00	100,00	9,33	0,00	20,00	6,67	6,67	6,67	0,44	0,44	0,44	0,67	0,67	0,67	4,95E+10	4,14E+10	6,31E+10	0,00	0	0	0,37	0	1	5,50	4	6	11761,00	11761	11761

6.4. Porównania dla zbioru Yeast

Zbiór Yeast ze względu na nierównomierny rozkład wzorców według klas i stosunkowo dużą ilość danych może stanowić problem dla wielu algorytmów klasyfikacji. Już podczas walidacji dla całego zbioru danych (patrz tabela 6.13) można zauważyć, że większość metod osiąga wyniki dużo gorsze niż dla innych zbiorów danych. Metoda SONN dla wspomnianej metody walidacji osiągnęła stuprocentową skuteczność, jednak analiza wskaźników takich, jak Q_MSE pozwala sądzić, że jest to związane ze specyficzną cechą tego algorytmu, dzięki której osiąga on stuprocentową skuteczność dla wzorców niesprzecznych. Zostaje to zweryfikowane podczas testu krzyżowego (tabele 6.14, 6.15), dla którego okazuje się, że metoda SONN mimo całkowitej skuteczności dla danych treningowych osiąga najgorsze rezultaty dla danych wcześniej nieznanych.

Tablica 6.13: Yeast, dane znormalizowane, test na całym zbiorze treningowym

Algorytm	Q_COR	Q_ERR	Q_SSE	Q_MSE	Q_RMSE	Q_ARV	E_TTR	E_TCL	E_TOT	E_MEM
SONN-3	100,00	0,00	1107,51	0,75	0,86	10910,34	20222	0,72	21288	24020508
KNN(k=1)	100,00	0,00	0,00	0,00	0,00	NaN	2	4,11	6100	142760
KNN(k=3)	49,80	50,20	262,11	0,18	0,42	2,48	0	4,74	7037	142760
KNN(k=5)	47,64	52,36	350,00	0,24	0,49	3,29	0	6,26	9294	142760
KNN(k=10)	44,81	55,19	422,82	0,28	0,53	4,46	0	8,97	13305	142760
KNN(k=15)	46,09	53,91	454,67	0,31	0,55	5,09	0	11,33	16821	142760
KNN(k=30)	43,53	56,47	493,53	0,33	0,58	6,09	0	20,12	29856	142760
SVML(c=1;e=0,1)	55,59	44,41	1040,11	0,70	0,84	223,69	1180	0,02	1217	480911
SVML(c=10;e=0,1)	55,12	44,88	1077,01	0,73	0,85	627,19	3197	0,02	3223	480911
SVML(c=50;e=0,1)	52,63	47,37	1095,52	0,74	0,86	1027,74	13081	0,02	13105	480911
SVML(c=100;e=0,1)	49,53	50,47	1079,48	0,73	0,85	661,44	34452	0,01	34474	480911
SVML(c=20;e=0,1)	51,75	48,25	1080,97	0,73	0,85	734,39	4735	0,01	4754	480911
SVML(c=20;e=0,01)	50,74	49,26	1096,78	0,74	0,86	1026,62	11068	0,01	11089	480911
SVML(c=20;e=0,001)	46,36	53,64	1093,89	0,74	0,86	948,73	57674	0,02	57699	480911
SVMR(c=10;s=0,5;e=0,1)	64,15	35,85	1051,24	0,71	0,84	393,73	2374605	2,65	2378533	480268
SVMR(c=10;s=1;e=0,1)	60,58	39,42	1045,15	0,70	0,84	283,32	353933	2,74	357995	480268
SVMR(c=10;s=5;e=0,1)	54,65	45,35	991,39	0,67	0,82	80,56	62177	3,50	67378	480268
SVMR(c=10;s=10;e=0,1)	47,17	52,83	988,98	0,67	0,82	69,12	122436	3,13	127077	480268
SVMP(c=10;d=3;e=0,01)	nieakceptowalny czas treningu									
SVMP(c=10;d=4;e=0,01)	nieakceptowalny czas treningu									
SVMP(c=10;d=6;e=0,01)	nieakceptowalny czas treningu									
RBFN(k=5;f=2;c=1)	53,50	46,50	767,59	0,52	0,72	28,68	90	0,02	120	132180
RBFN(k=10;f=2;c=1)	58,29	41,71	738,31	0,50	0,71	27,93	21	0,04	78	132605
RBFN(k=15;f=2;c=1)	59,23	40,77	732,21	0,49	0,70	28,80	27	0,05	105	133030
RBFN(k=20;f=2;c=1)	59,97	40,03	670,90	0,45	0,67	17,99	39	0,06	135	133455
RBFN(k=5;f=1;c=2)	48,92	51,08	796,72	0,54	0,73	30,58	88	0,02	122	132566
RBFN(k=10;f=1;c=2)	57,68	42,32	749,79	0,51	0,71	30,13	115	0,04	172	133361
RBFN(k=15;f=1;c=2)	58,76	41,24	731,23	0,49	0,70	27,26	150	0,06	236	134156
RBFN(k=20;f=1;c=2)	59,37	40,63	716,25	0,48	0,69	25,25	437	0,08	549	134951
RBFN(k=5;f=2;c=2)	52,36	47,64	789,68	0,53	0,73	32,12	11	0,03	56	132202
RBFN(k=10;f=2;c=2)	58,36	41,64	737,16	0,50	0,70	29,36	18	0,04	77	132627
RBFN(k=15;f=2;c=2)	60,04	39,96	721,82	0,49	0,70	26,56	29	0,05	101	133052
RBFN(k=20;f=2;c=2)	60,98	39,02	687,06	0,46	0,68	22,16	37	0,07	135	133477
PNN(sigma=0,1)	63,27	36,73	674,22	0,45	0,67	9,53	0	10,15	15066	138947
PNN(sigma=1)	50,27	49,73	1172,04	0,79	0,89	2,43E+04	0	10,05	14919	138947
PNN(sigma=10)	48,25	51,75	1201,71	0,81	0,90	3,24E+08	0	10,21	15150	138947
PNN(sigma=100)	48,18	51,82	1202,04	0,81	0,90	3,25E+12	0	10,34	15351	138947

Tablica 6.14: Yeast, dane znormalizowane, 10-fold cross-validation, trening

Algorytm	Q_COR(avg)	Q_COR(min)	Q_COR(max)	Q_ERR(avg)	Q_ERR(min)	Q_ERR(max)	Q_SSE(avg)	Q_SSE(min)	Q_SSE(max)	Q_MSE(avg)	Q_MSE(min)	Q_MSE(max)	Q_RMSE(avg)	Q_RMSE(min)	Q_RMSE(max)	Q_ARV(avg)	Q_ARV(min)	Q_ARV(max)	E_TTR(avg)	E_TTR(min)	E_TTR(max)	E_TCL(avg)	E_TCL(min)	E_TCL(max)	E_TOT(avg)	E_TOT(min)	E_TOT(max)	E_MEM(avg)	E_MEM(min)	E_MEM(max)
SONN-3	100,00	100,00	100,00	0,00	0,00	0,00	991,52	975,69	1001,67	0,74	0,73	0,75	0,86	0,85	0,87	8418,71	3415,70	15393,89	15645,90	14764	16399	0,64	0	18	16503,20	15594	17215	2131980,50	2106899	2162686
KNN(k=1)	100,00	100,00	100,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	NaN	NaN	NaN	0,40	0	3	3,33	3	41	4445,90	4407	4506	128513,60	128456	128552
KNN(k=3)	49,19	48,20	50,97	50,81	49,03	51,80	238,34	232,11	244,22	0,18	0,17	0,18	0,42	0,42	0,43	2,51	2,43	2,58	0,20	0	1	3,71	3	12	4950,90	4933	4986	128513,60	128456	128552
KNN(k=5)	48,29	46,67	49,89	51,71	50,11	53,33	314,62	306,04	320,04	0,24	0,23	0,24	0,49	0,48	0,49	3,32	3,18	3,45	0,10	0	1	4,74	4	84	6334,30	6237	6619	128513,60	128456	128552
KNN(k=10)	44,41	41,27	46,71	55,59	53,29	58,73	382,80	376,62	389,27	0,29	0,28	0,29	0,54	0,53	0,54	4,55	4,32	4,73	0,20	0	1	6,88	6	19	9193,70	9142	9336	128513,60	128456	128552
KNN(k=15)	44,53	41,84	46,71	55,47	53,29	58,16	412,49	404,66	419,91	0,31	0,30	0,31	0,56	0,55	0,56	5,15	4,88	5,42	0,00	0	0	8,63	8	35	11519,90	11478	11594	128513,60	128456	128552
KNN(k=30)	42,06	38,02	44,42	57,94	55,58	61,98	448,63	439,74	455,03	0,34	0,33	0,34	0,58	0,57	0,58	6,23	5,98	6,61	0,10	0	1	15,27	14	1065	20392,60	20198	21367	128513,60	128456	128552
SVML(c=1;e=0,1)	50,44	44,91	55,54	49,56	44,46	55,09	930,78	917,49	946,58	0,70	0,69	0,71	0,83	0,83	0,84	174,17	111,19	260,65	875,00	786	1077	0,02	0	14	895,30	806	1109	433571,40	433380	433699
SVML(c=10;e=0,1)	51,13	44,27	55,66	48,87	44,34	55,73	971,50	960,11	984,46	0,73	0,72	0,74	0,85	0,85	0,86	698,46	514,04	1020,25	2011,00	1776	2314	0,01	0	4	2029,20	1797	2332	433571,40	433380	433699
SVML(c=50;e=0,1)	49,57	44,91	55,16	50,43	44,84	55,09	969,36	958,40	976,21	0,73	0,72	0,73	0,85	0,85	0,86	681,73	445,11	904,46	25476,50	21354	32131	0,01	0	1	25494,10	21370	32147	433571,40	433380	433699
SVML(c=100;e=0,1)	49,46	45,58	54,72	50,54	45,28	54,42	972,56	967,31	986,09	0,73	0,72	0,74	0,85	0,85	0,86	740,07	584,52	1037,95	8479,10	7508	9441	0,01	0	1	8497,00	7527	9459	433571,40	433380	433699
SVML(c=20;e=0,1)	50,68	45,32	56,29	49,32	43,71	54,68	973,66	963,45	987,84	0,73	0,72	0,74	0,85	0,85	0,86	760,71	503,15	1141,55	3581,60	3196	3950	0,01	0	19	3601,20	3235	3968	433571,40	433380	433699
SVML(c=20;e=0,01)	49,65	45,28	52,51	50,35	47,49	54,72	981,91	970,48	992,37	0,74	0,73	0,74	0,86	0,85	0,86	918,65	625,47	1177,76	8010,10	6766	8833	0,01	0	1	8027,10	6783	8852	433571,40	433380	433699
SVML(c=20;e=0,001)	49,56	45,99	54,16	50,44	45,84	54,01	985,04	979,91	992,29	0,74	0,73	0,74	0,86	0,86	0,86	991,83	877,06	1102,43	44234,90	32208	59021	0,01	0	3	44253,10	32226	59038	433571,40	433380	433699
SVMR(c=10;s=0,5;e=0,1)	63,79	62,17	64,67	36,21	35,33	37,83	948,17	935,39	955,68	0,71	0,70	0,72	0,84	0,84	0,85	409,92	311,56	528,74	1155647,50	1016849	1305630	1,90	1	16	1158186,70	1019386	1308172	432928,40	432737	433056
SVMR(c=10;s=1;e=0,1)	60,57	58,95	62,80	39,43	37,20	41,05	935,08	925,92	945,73	0,70	0,69	0,71	0,84	0,83	0,84	258,59	207,29	297,99	218574,70	172362	297160	1,91	1	6	221124,90	174939	299702	432928,40	432737	433056
SVMR(c=10;s=5;e=0,1)	54,28	48,39	57,30	45,72	42,70	51,61	900,37	886,34	929,12	0,67	0,66	0,70	0,82	0,81	0,83	96,98	61,28	161,76	34357,60	26413	43158	1,97	1	5	36984,00	29052	45781	432928,40	432737	433056
SVMR(c=10;s=10;e=0,1)	50,85	47,75	53,63	49,15	46,37	52,25	878,19	854,26	898,14	0,66	0,64	0,67	0,81	0,80	0,82	78,45	66,74	103,38	31840,80	27585	38285	1,97	1	35	34470,20	30236	40871	432928,40	432737	433056
SVMP(c=10;d=3;e=0,01)	nieakceptowalny czas treningu																													
SVMP(c=10;d=4;e=0,01)	nieakceptowalny czas treningu																													
SVMP(c=10;d=6;e=0,01)	nieakceptowalny czas treningu																													
RBFN(k=5;f=2;c=1)	50,13	43,64	54,27	49,87	45,73	56,36	716,73	691,14	744,17	0,54	0,52	0,56	0,73	0,72	0,75	38,81	27,18	46,21	16,20	5	112	0,02	0	1	37,90	25	135	119120,80	119068	119156
RBFN(k=10;f=2;c=1)	57,77	55,73	59,51	42,23	40,49	44,27	662,24	651,46	678,44	0,50	0,49	0,51	0,70	0,70	0,71	26,76	23,62	30,61	11,30	11	14	0,03	0	38	50,60	45	86	119545,80	119493	119581
RBFN(k=15;f=2;c=1)	59,70	58,16	60,48	40,30	39,52	41,84	642,61	628,80	654,08	0,48	0,47	0,49	0,69	0,69	0,70	24,40	21,30	28,72	17,70	17	19	0,04	0	1	70,10	68	74	119970,80	119918	120006
RBFN(k=20;f=2;c=1)	nie udało się przeprowadzić treningu																													
RBFN(k=5;f=1;c=2)	46,80	43,19	54,04	53,20	45,96	56,81	736,92	697,30	764,21	0,55	0,52	0,57	0,74	0,72	0,76	55,08	32,40	82,27	34,00	11	58	0,02	0	1	58,50	35	84	119506,80	119454	119542
RBFN(k=10;f=1;c=2)	57,70	55,99	59,21	42,30	40,79	44,01	674,64	662,82	688,20	0,51	0,50	0,52	0,71	0,70	0,72	30,34	28,59	35,88	76,80	41	114	0,03	0	1	119,00	85	159	120301,80	120249	120337
RBFN(k=15;f=1;c=2)	59,31	58,50	60,33	40,69	39,67	41,50	657,60	638,29	672,47	0,49	0,48	0,50	0,70	0,69	0,71	27,94	25,02	31,41	135,30	103	187	0,05	0	1	195,50	160	248	121096,80	121044	121132
RBFN(k=20;f=1;c=2)	60,16	59,33	60,85	39,84	39,15	40,67	642,76	629,78	661,98	0,48	0,47	0,50	0,69	0,69	0,70	25,41	22,83	29,06	185,00	120	245	0,06	0	4	261,50	196	320	121891,80	121839	121927
RBFN(k=5;f=2;c=2)	51,34	47,08	54,46	48,66	45,54	52,92	707,73	684,25	726,41	0,53	0,51	0,54	0,73	0,72	0,74	35,42	25,15	50,77	5,80	5	6	0,02	0	1	27,60	26	30	119142,80	119090	119178
RBFN(k=10;f=2;c=2)	58,15	56,63	59,58	41,85	40,42	43,37	663,19	638,79	688,56	0,50	0,48	0,52	0,70	0,69	0,72	27,15	22,19	34,61	11,50	11	12	0,03	0	67	55,10	46	115	119567,80	119515	119603
RBFN(k=15;f=2;c=2)	59,61	57,45	61,30	40,39	38,70	42,55	645,63	626,80	666,02	0,48	0,47	0,50	0,70	0,69	0,71	25,53	22,12	28,62	17,50	17	19	0,04	0	2	69,30	65	72	119992,80	119940	120028
RBFN(k=20;f=2;c=2)	60,36	59,03	61,80	39,64	38,20	40,97	624,39	609,08	642,97	0,47	0,46	0,48	0,68	0,68	0,69	22,08	19,16	25,96	24,20	24	25	0,05	0	4	90,40	86	97	120417,80	120365	120453
PNN(sigma=0,1)	63,61	62,57	64,57	36,39	35,43	37,43	602,27	592,02	611,65	0,45	0,44	0,46	0,67	0,67	0,68	9,33	8,79	9,75	0,00	0	0	6,69	6	55	8929,40	8890	8992	125294,20	125239	125331
PNN(sigma=1)	49,56	48,28	52,13	50,44	47,87	51,72	1054,82	1053,55	1055,58	0,79	0,79	0,79	0,89	0,89	0,89	2,44E+04	2,21E+04	2,96E+04	0,00	0	0	6,68	6	14	8923,60	8869	8996	125294,20	125239	125331
PNN(sigma=10)	48,07	46,03	51,01	51,93	48,99	53,97	1081,54	1081,05	1081,87	0,81	0,81	0,81	0,90	0,90	0,90	3,25E+08	3,00E+08	3,75E+08	0,00	0	0	6,68	6	13	8924,30	8882	8967	125294,20	125239	125331
PNN(sigma=100)	48,05	46,03	51,01	51,95	48,99	53,97	1081,83	1081,35	1082,16	0,81	0,81	0,81	0,90	0,90	0,90	3,26E+12	3,01E+12	3,76E+12	0,00	0	0	6,71	6	11	8968,10	8898	9249	125294,20	125239	125331

Tablica 6.15: Yeast, dane znormalizowane, 10-fold cross-validation, walidacja

Algorytm	Q_COR(avg)	Q_COR(min)	Q_COR(max)	Q_ERR(avg)	Q_ERR(min)	Q_ERR(max)	Q_SSE(avg)	Q_SSE(min)	Q_SSE(max)	Q_MSE(avg)	Q_MSE(min)	Q_MSE(max)	Q_RMSE(avg)	Q_RMSE(min)	Q_RMSE(max)	Q_ARV(avg)	Q_ARV(min)	Q_ARV(max)	E_TTR(avg)	E_TTR(min)	E_TTR(max)	E_TCL(avg)	E_TCL(min)	E_TCL(max)	E_TOT(avg)	E_TOT(min)	E_TOT(max)	E_MEM(avg)	E_MEM(min)	E_MEM(max)
SONN-3	35,92	31,76	41,89	64,08	58,11	68,24	112,28	110,96	112,87	0,76	0,75	0,76	0,87	0,87	0,87	3612,36	1551,75	5171,82	15645,90	14764	16399	0,64	0	2	15740,80	14860	16491	2131980,50	2106989,92	1626886
KNN(k=1)	52,30	44,30	56,76	47,70	43,24	55,70	70,80	64,00	83,00	0,48	0,43	0,56	0,69	0,66	0,75	1,92	1,76	2,26	0,40	0	3	3,32	3	12	493,10	478	506	128513,60	128456	128552
KNN(k=3)	47,24	40,27	52,70	52,76	47,30	59,73	55,72	51,33	63,11	0,38	0,35	0,42	0,61	0,59	0,65	3,04	2,58	3,51	0,20	0	1	3,72	3	8	552,20	548	558	128513,60	128456	128552
KNN(k=5)	47,30	40,54	53,38	52,70	46,62	59,46	53,38	48,32	61,80	0,36	0,33	0,41	0,60	0,57	0,64	3,64	3,13	3,95	0,10	0	1	4,76	4	16	706,90	687	771	128513,60	128456	128552
KNN(k=10)	44,68	38,93	51,68	55,32	48,32	61,07	52,47	48,04	57,89	0,35	0,32	0,39	0,59	0,57	0,62	4,80	4,04	5,26	0,20	0	1	6,89	6	26	1023,00	1011	1062	128513,60	128456	128552
KNN(k=15)	44,81	37,84	53,38	55,19	46,62	62,16	52,66	46,57	58,38	0,35	0,31	0,39	0,60	0,56	0,63	5,32	4,65	6,10	0,00	0	0	8,64	8	12	1282,30	1268	1294	128513,60	128456	128552
KNN(k=30)	44,88	36,91	53,02	55,12	46,98	63,09	53,54	47,48	59,89	0,36	0,32	0,40	0,60	0,56	0,63	6,37	5,72	7,38	0,10	0	1	15,19	14	23	2254,00	2223	2300	128513,60	128456	128552
SVML(c=1;e=0,1)	48,79	44,59	56,08	51,21	43,92	55,41	103,77	101,88	105,76	0,70	0,69	0,71	0,84	0,83	0,84	186,54	96,47	366,13	875,00	786	1077	0,01	0	1	877,10	787	1079	433571,40	433380	433699
SVML(c=10;e=0,1)	49,59	41,22	59,46	50,41	40,54	58,78	108,15	106,52	109,35	0,73	0,72	0,74	0,85	0,85	0,86	815,45	337,73	1140,47	2011,00	1776	2314	0,01	0	1	2012,90	1778	2316	433571,40	433380	433699
SVML(c=50;e=0,1)	47,30	38,51	54,73	52,70	45,27	61,49	107,95	106,22	109,59	0,73	0,72	0,74	0,85	0,85	0,86	767,26	427,54	1110,29	25476,50	21354	32131	0,01	0	1	25478,40	21356	32133	433571,40	433380	433699
SVML(c=100;e=0,1)	48,45	41,22	54,05	51,55	45,95	58,78	108,30	106,44	110,29	0,73	0,72	0,74	0,85	0,85	0,86	846,91	474,53	1177,35	8479,10	7508	9441	0,01	0	1	8481,00	7510	9442	433571,40	433380	433699
SVML(c=20;e=0,1)	48,45	45,95	54,73	51,55	45,27	54,05	108,42	106,30	109,59	0,73	0,72	0,74	0,85	0,85	0,86	887,77	488,75	1231,92	3581,60	3196	3950	0,01	0	1	3583,50	3198	3950	433571,40	433380	433699
SVML(c=20;e=0,01)	48,99	43,62	61,07	51,01	38,93	56,38	109,31	107,49	110,88	0,74	0,73	0,74	0,86	0,85	0,86	1047,49	598,40	1421,39	8010,10	6766	8833	0,01	0	1	8012,00	6768	8834	433571,40	433380	433699
SVML(c=20;e=0,001)	48,65	42,57	53,38	51,35	46,62	57,43	109,56	108,04	111,16	0,74	0,73	0,75	0,86	0,85	0,86	1129,57	552,07	2102,90	44234,90	32208	59021	0,01	0	1	44237,10	32210	59023	433571,40	433380	433699
SVMR(c=10;s=0,5;e=0,1)	60,04	54,05	69,13	39,96	30,87	45,95	106,38	103,68	108,78	0,72	0,70	0,73	0,85	0,84	0,85	427,90	263,06	510,82	1155647,50	1016849	1305630	1,91	1	6	1155931,10	1017128	1305916	432928,40	432737	433056
SVMR(c=10;s=1;e=0,1)	58,89	52,03	66,44	41,11	33,56	47,97	104,43	101,32	106,65	0,70	0,68	0,72	0,84	0,83	0,85	267,95	163,99	334,99	218574,70	172362	297160	1,91	1	5	218858,60	172637	297448	432928,40	432737	433056
SVMR(c=10;s=5;e=0,1)	52,97	45,95	62,84	47,03	37,16	54,05	100,64	98,53	103,08	0,68	0,67	0,70	0,82	0,82	0,83	101,45	70,72	133,21	34357,60	26413	43158	1,96	1	3	34648,00	26699	43454	432928,40	432737	433056
SVMR(c=10;s=10;e=0,1)	50,74	43,24	57,05	49,26	42,95	56,76	98,10	93,33	102,27	0,66	0,63	0,69	0,81	0,79	0,83	83,86	59,65	116,26	31840,80	27585	38285	1,96	1	5	32131,90	27877	38575	432928,40	432737	433056
SVMP(c=10;d=3;e=0,01)	nieakceptowalny czas treningu																													
SVMP(c=10;d=4;e=0,01)	nieakceptowalny czas treningu																													
SVMP(c=10;d=6;e=0,01)	nieakceptowalny czas treningu																													
RBFN(k=5;f=2;c=1)	50,67	39,86	62,42	49,33	37,58	60,14	79,30	74,94	83,83	0,53	0,51	0,56	0,73	0,71	0,75	38,07	26,93	47,57	16,20	5	112	0,02	0	1	18,90	7	115	119120,80	119068	119156
RBFN(k=10;f=2;c=1)	57,34	47,97	70,47	42,66	29,53	52,03	74,37	69,14	79,08	0,50	0,46	0,53	0,71	0,68	0,73	27,57	22,80	34,95	11,30	11	14	0,03	0	1	15,20	14	18	119545,80	119493	119581
RBFN(k=15;f=2;c=1)	58,56	47,97	69,13	41,44	30,87	52,03	72,54	69,42	77,73	0,49	0,47	0,53	0,70	0,68	0,72	24,89	18,74	33,17	17,70	17	19	0,04	0	1	23,30	21	25	119970,80	119918	120006
RBFN(k=20;f=2;c=1)	nie udało się przeprowadzić treningu																													
RBFN(k=5;f=1;c=2)	45,36	33,56	56,08	54,64	43,92	66,44	82,81	77,92	87,90	0,56	0,53	0,59	0,75	0,73	0,77	57,00	34,36	83,68	34,00	11	58	0,02	0	1	36,60	14	61	119506,80	119454	119542
RBFN(k=10;f=1;c=2)	56,67	43,24	67,11	43,33	32,89	56,76	75,75	72,21	81,03	0,51	0,48	0,55	0,71	0,70	0,74	31,26	24,38	37,39	76,80	41	114	0,10	0	99	91,70	47	218	120301,80	120249	120337
RBFN(k=15;f=1;c=2)	58,15	47,30	68,46	41,85	31,54	52,70	73,80	69,85	77,94	0,50	0,47	0,53	0,71	0,68	0,73	28,16	24,66	32,34	135,30	103	187	0,05	0	1	142,30	110	192	121096,80	121044	121132
RBFN(k=20;f=1;c=2)	58,29	47,30	72,48	41,72	27,52	52,70	72,68	68,14	77,48	0,49	0,46	0,52	0,70	0,68	0,72	25,83	22,01	33,68	185,00	120	245	0,06	0	1	193,50	128	253	121891,80	121839	121927
RBFN(k=5;f=2;c=2)	50,60	42,57	59,06	49,40	40,94	57,43	79,22	76,03	84,11	0,53	0,51	0,56	0,73	0,72	0,75	36,15	25,82	49,22	5,80	5	6	0,02	0	1	8,30	7	9	119142,80	119090	119178
RBFN(k=10;f=2;c=2)	57,14	45,95	68,46	42,86	31,54	54,05	74,85	70,03	78,97	0,50	0,47	0,53	0,71	0,69	0,73	27,60	20,89	35,89	11,50	11	12	0,03	0	1	15,30	14	17	119567,80	119515	119603
RBFN(k=15;f=2;c=2)	57,95	49,32	69,80	42,05	30,20	50,68	72,86	69,56	78,18	0,49	0,47	0,53	0,70	0,69	0,73	26,23	21,06	33,34	17,50	17	19	0,04	0	1	23,00	20	25	119992,80	119940	120028
RBFN(k=20;f=2;c=2)	59,30	52,35	69,80	40,70	30,20	47,65	70,78	66,15	75,64	0,48	0,44	0,51	0,69	0,67	0,71	22,31	19,21	26,37	24,20	24	25	0,05	0	2	31,70	31	33	120417,80	120365	120453
PNN(sigma=0,1)	56,54	48,99	65,10	43,46	34,90	51,01	74,60	70,48	79,42	0,50	0,47	0,53	0,71	0,69	0,73	11,85	8,20	13,93	0,00	0	0	6,68	6	10	991,10	977	1017	125294,20	125239	125331
PNN(sigma=1)	48,78	42,57	57,05	51,22	42,95	57,43	117,22	116,80	117,82	0,79	0,79	0,79	0,89	0,89	0,89	3,21E+04	1,30E+04	7,16E+04	0,00	0	0	6,65	6	10	986,30	974	1004	125294,20	125239	125331
PNN(sigma=10)	47,10	40,54	55,03	52,90	44,97	59,46	120,17	119,85	120,66	0,81	0,81	0,81	0,90	0,90	0,90	3,75E+08	2,10E+08	7,95E+08	0,00	0	0	6,68	6	10	992,00	976	1007	125294,20	125239	125331
PNN(sigma=100)	47,10	40,54	55,03	52,90	44,97	59,46	120,20	119,88	120,69	0,81	0,81	0,81	0,90	0,90	0,90	3,76E+122	1,11E+12	7,96E+12	0,00	0	0	6,72	6	10	997,40	981	1041	125294,20	125239	125331

6.5. Porównania dla zbioru Congressional Voting Records

Congressional Voting Records jest jednym z ciekawszych spośród analizowanych w niniejszej pracy zbiorów. Ponieważ zbiór ten składa się wyłącznie z wartości $+1$, -1 oraz 0 można przypuszczać, że sieci SONN, które pierwotnie były algorytmami działającymi wyłacznie dla zbiorów złożonych z cech o wartościach binarnych, osiągną dla tego zbioru bardzo dobre wyniki. Analiza wartości w tabelach 6.16, 6.17 oraz 6.18 zdaje się potwierdzać takie przypuszczenia. Szczególnie warto zwrócić uwagę na wysokie wartości wskaźników Q_COR oraz niskie wartości wskaźnika Q_MSE.

Można także zauważyć, że dla tego zbioru metoda radialnych sieci funkcji bazowych w pięciu z dwunastu przypadków nie była w stanie przeprowadzić procesu treningu.

Tablica 6.16: Congressional Voting Records, dane znormalizowane, test na całym zbiorze treningowym

Algorytm	Q_COR	Q_ERR	Q_SSE	Q_MSE	Q_RMSE	Q_ARV	E_TTR	E_TCL	E_TOT	E_MEM
SONN-3	100,00	0,00	0,00	0,00	0,00	NaN	112	0,03	127	921824
KNN(k=1)	100,00	0,00	0,00	0,00	0,00	NaN	0	1,40	611	72407
KNN(k=3)	92,64	7,36	9,89	0,02	0,15	1,11	0	1,49	648	72407
KNN(k=5)	90,80	9,20	13,72	0,03	0,18	1,13	0	1,84	800	72407
KNN(k=10)	90,80	9,20	17,10	0,04	0,20	1,17	0	2,81	1223	72407
KNN(k=15)	89,66	10,34	18,84	0,04	0,21	1,19	0	3,38	1470	72407
KNN(k=30)	89,20	10,80	22,53	0,05	0,23	1,22	0	5,93	2581	72407
SVML(c=1;e=0,1)	97,01	2,99	13,00	0,03	0,17	1,03	227	0,00	229	86876
SVML(c=10;e=0,1)	97,01	2,99	13,00	0,03	0,17	1,03	2188	0,00	2189	86876
SVML(c=50;e=0,1)	97,01	2,99	13,00	0,03	0,17	1,03	13899	0,01	13903	86876
SVML(c=100;e=0,1)	96,78	3,22	14,00	0,03	0,18	1,03	18006	0,00	18008	86876
SVML(c=20;e=0,1)	97,47	2,53	11,00	0,03	0,16	1,03	4235	0,01	4238	86876
SVML(c=20;e=0,01)	97,01	2,99	13,00	0,03	0,17	1,03	8486	0,01	8489	86876
SVML(c=20;e=0,001)	97,47	2,53	11,00	0,03	0,16	1,03	9790	0,00	9792	86876
SVMR(c=10;s=0,5;e=0,1)	100,00	0,00	0,00	0,00	0,00	NaN	4143	0,34	4292	86673
SVMR(c=10;s=1;e=0,1)	100,00	0,00	0,00	0,00	0,00	NaN	4210	0,20	4295	86673
SVMR(c=10;s=5;e=0,1)	96,78	3,22	14,00	0,03	0,18	1,03	2375	0,14	2437	86673
SVMR(c=10;s=10;e=0,1)	95,86	4,14	18,00	0,04	0,20	1,04	530	0,16	601	86673
SVMP(c=10;d=3;e=0,01)	100,00	0,00	0,00	0,00	0,00	NaN	89185	0,11	89232	86987
SVMP(c=10;d=4;e=0,01)	66,67	33,33	145,00	0,33	0,58	1,50	28428	0,09	28465	86987
SVMP(c=10;d=6;e=0,01)	38,62	61,38	435,00	1,00	1,00	∞	23846	0,09	23885	86987
RBFN(k=5;f=2;c=1)	90,34	9,66	42,00	0,10	0,31	1,11	4	0,01	8	69675
RBFN(k=10;f=2;c=1)	94,48	5,52	24,00	0,06	0,23	1,06	3	0,01	6	69780
RBFN(k=15;f=2;c=1)	95,86	4,14	18,00	0,04	0,20	1,04	4	0,01	10	69885
RBFN(k=20;f=2;c=1)	nie udało się przeprowadzić treningu									
RBFN(k=5;f=1;c=2)	92,41	7,59	33,00	0,08	0,28	1,08	16	0,00	18	70381
RBFN(k=10;f=1;c=2)	95,63	4,37	19,00	0,04	0,21	1,05	36	0,01	39	71176
RBFN(k=15;f=1;c=2)	96,78	3,22	14,00	0,03	0,18	1,03	26	0,01	32	71971
RBFN(k=20;f=1;c=2)	96,32	3,68	16,00	0,04	0,19	1,04	38	0,02	46	72766
RBFN(k=5;f=2;c=2)	90,57	9,43	41,00	0,09	0,31	1,10	2	0,00	4	69697
RBFN(k=10;f=2;c=2)	92,18	7,82	34,00	0,08	0,28	1,08	3	0,01	7	69802
RBFN(k=15;f=2;c=2)	94,94	5,06	22,00	0,05	0,22	1,05	5	0,01	11	69907
RBFN(k=20;f=2;c=2)	95,86	4,14	18,00	0,04	0,20	1,04	7	0,02	14	70012
PNN(sigma=0,1)	100,00	0,00	0,00	0,00	0,00	1,01	0	1,09	476	72661
PNN(sigma=1)	90,57	9,43	30,49	0,07	0,26	1,74	0	1,52	663	72661
PNN(sigma=10)	89,43	10,57	106,26	0,24	0,49	1,44E+04	0	1,12	489	72661
PNN(sigma=100)	89,43	10,57	108,72	0,25	0,50	1,47E+08	0	1,10	480	72661

Tablica 6.17: Congressional Voting Records, dane znormalizowane, 10-fold cross-validation, trening

Algorytm	Q_COR(avg)	Q_COR(min)	Q_COR(max)	Q_ERR(avg)	Q_ERR(min)	Q_ERR(max)	Q_SSE(avg)	Q_SSE(min)	Q_SSE(max)	Q_MSE(avg)	Q_MSE(min)	Q_MSE(max)	Q_RMSE(avg)	Q_RMSE(min)	Q_RMSE(max)	Q_ARV(avg)	Q_ARV(min)	Q_ARV(max)	E_TTR(avg)	E_TTR(min)	E_TTR(max)	E_TCL(avg)	E_TCL(min)	E_TCL(max)	E_TOT(avg)	E_TOT(min)	E_TOT(max)	E_MEM(avg)	E_MEM(min)	E_MEM(max)
SONN-3	100,00	100,00	100,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	NaN	NaN	NaN	96,00	87	128	0,03	0	1	108,10	98	140	829865,70	818930	834636
KNN(k=1)	100,00	100,00	100,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	NaN	NaN	NaN	0,00	0	0	1,18	1	12	463,70	454	490	65195,90	65103	65286
KNN(k=3)	92,49	91,84	93,88	7,51	6,12	8,16	9,11	8,11	10,00	0,02	0,02	0,03	0,15	0,14	0,16	1,11	1,09	1,12	0,00	0	0	1,43	1	74	561,10	515	691	65195,90	65103	65286
KNN(k=5)	91,16	89,29	92,60	8,84	7,40	10,71	12,49	11,44	13,84	0,03	0,03	0,04	0,18	0,17	0,19	1,13	1,12	1,15	0,00	0	0	1,71	1	16	669,00	649	693	65195,90	65103	65286
KNN(k=10)	91,24	90,54	92,07	8,76	7,93	9,46	15,40	14,37	16,22	0,04	0,04	0,04	0,20	0,19	0,20	1,17	1,15	1,19	0,00	0	0	2,50	2	78	980,00	933	1157	65195,90	65103	65286
KNN(k=15)	89,22	86,99	91,33	10,78	8,67	13,01	17,21	15,34	18,28	0,04	0,04	0,05	0,21	0,20	0,22	1,19	1,17	1,21	0,00	0	0	3,18	2	60	1246,80	1203	1367	65195,90	65103	65286
KNN(k=30)	88,48	86,73	89,80	11,52	10,20	13,27	20,51	18,60	21,97	0,05	0,05	0,06	0,23	0,22	0,24	1,23	1,20	1,25	0,00	0	0	5,55	5	71	2171,60	2129	2219	65195,90	65103	65286
SVML(c=1;e=0,1)	97,32	96,68	97,70	2,68	2,30	3,32	10,50	9,00	13,00	0,03	0,02	0,03	0,16	0,15	0,18	1,03	1,02	1,03	237,00	174	295	0,01	0	1	239,20	176	298	78567,50	78472	78663
SVML(c=10;e=0,1)	97,42	96,68	97,96	2,58	2,04	3,32	10,10	8,00	13,00	0,03	0,02	0,03	0,16	0,14	0,18	1,03	1,02	1,03	2182,10	1290	2852	0,01	0	1	2184,30	1293	2854	78567,50	78472	78663
SVML(c=50;e=0,1)	97,04	96,42	97,96	2,96	2,04	3,58	11,60	8,00	14,00	0,03	0,02	0,04	0,17	0,14	0,19	1,03	1,02	1,04	8191,40	2939	13771	0,01	0	1	8194,40	2942	13774	78567,50	78472	78663
SVML(c=100;e=0,1)	97,11	96,42	97,45	2,89	2,55	3,58	11,30	10,00	14,00	0,03	0,03	0,04	0,17	0,16	0,19	1,03	1,03	1,04	8890,40	4675	14751	0,01	0	1	8892,70	4676	14755	78567,50	78472	78663
SVML(c=20;e=0,1)	97,45	96,94	98,21	2,55	1,79	3,06	10,00	7,00	12,00	0,03	0,02	0,03	0,16	0,13	0,18	1,03	1,02	1,03	3742,10	1680	7297	0,00	0	1	3743,90	1680	7299	78567,50	78472	78663
SVML(c=20;e=0,01)	97,42	96,93	97,96	2,58	2,04	3,07	10,10	8,00	12,00	0,03	0,02	0,03	0,16	0,14	0,18	1,03	1,02	1,03	3797,50	1561	9125	0,01	0	1	3799,60	1563	9127	78567,50	78472	78663
SVML(c=20;e=0,001)	97,50	96,94	98,47	2,50	1,53	3,06	9,80	6,00	12,00	0,03	0,02	0,03	0,16	0,12	0,18	1,03	1,02	1,03	5915,20	3942	9352	0,01	0	1	5917,70	3944	9354	78567,50	78472	78663
SVMR(c=10;s=0,5;e=0,1)	100,00	100,00	100,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	NaN	NaN	NaN	3152,70	2406	4647	0,34	0	67	3284,50	2531	4770	78364,50	78269	78460
SVMR(c=10;s=1;e=0,1)	99,82	99,49	100,00	0,18	0,00	0,51	0,70	0,00	2,00	0,00	0,00	0,01	0,03	0,00	0,07	NaN	NaN	1,01	3687,50	2578	5484	0,18	0	2	3758,80	2650	5556	78364,50	78269	78460
SVMR(c=10;s=5;e=0,1)	96,70	96,16	97,70	3,30	2,30	3,84	12,90	9,00	15,00	0,03	0,02	0,04	0,18	0,15	0,20	1,03	1,02	1,04	1783,10	1209	2425	0,12	0	1	1831,60	1258	2474	78364,50	78269	78460
SVMR(c=10;s=10;e=0,1)	95,81	95,14	96,43	4,19	3,57	4,86	16,40	14,00	19,00	0,04	0,04	0,05	0,20	0,19	0,22	1,04	1,04	1,05	556,30	393	769	0,15	0	1	613,60	452	826	78364,50	78269	78460
SVMP(c=10;d=3;e=0,01)	99,67	99,23	100,00	0,33	0,00	0,77	1,30	0,00	3,00	0,00	0,00	0,01	0,05	0,00	0,09	NaN	NaN	1,01	89473,80	53106	172734	0,10	0	2	89514,30	53147	172777	78678,50	78583	78774
SVMP(c=10;d=4;e=0,01)	61,10	37,34	67,77	38,90	32,23	62,66	182,40	126,00	392,00	0,47	0,32	1,00	0,66	0,57	1,00	∞	1,48	∞	21163,10	17628	22667	0,08	0	6	21195,90	17659	22697	78678,50	78583	78774
SVMP(c=10;d=6;e=0,01)	38,62	37,34	40,15	61,38	59,85	62,66	391,50	391,00	392,00	1,00	1,00	1,00	1,00	1,00	1,00	∞	∞	∞	18317,70	17991	19253	0,08	0	1	18349,40	18022	19287	78678,50	78583	78774
RBFN(k=5;f=2;c=1)	91,39	87,50	95,15	8,61	4,85	12,50	33,70	19,00	49,00	0,09	0,05	0,13	0,29	0,22	0,35	1,10	1,05	1,14	1,40	1	2	0,01	0	1	4,00	2	5	62811,90	62723	62898
RBFN(k=10;f=2;c=1)	93,92	92,07	95,91	6,08	4,09	7,93	23,80	16,00	31,00	0,06	0,04	0,08	0,25	0,20	0,28	1,06	1,04	1,09	2,70	2	3	0,01	0	1	7,10	6	10	62916,90	62828	63003
RBFN(k=15;f=2;c=1)	nie udało się przeprowadzić treningu																													
RBFN(k=20;f=2;c=1)	nie udało się przeprowadzić treningu																													
RBFN(k=5;f=1;c=2)	93,44	90,54	94,64	6,56	5,36	9,46	25,70	21,00	37,00	0,07	0,05	0,09	0,26	0,23	0,31	1,07	1,06	1,10	10,70	7	16	0,01	0	2	13,30	10	18	63517,90	63429	63604
RBFN(k=10;f=1;c=2)	95,25	93,62	96,17	4,75	3,83	6,38	18,60	15,00	25,00	0,05	0,04	0,06	0,22	0,20	0,25	1,05	1,04	1,07	25,10	16	35	0,01	0	1	29,20	18	40	64312,90	64224	64399
RBFN(k=15;f=1;c=2)	96,12	94,90	96,94	3,88	3,06	5,10	15,20	12,00	20,00	0,04	0,03	0,05	0,20	0,18	0,23	1,04	1,03	1,05	28,40	22	40	0,02	0	2	34,50	31	44	65107,90	65019	65194
RBFN(k=20;f=1;c=2)	96,48	95,91	96,93	3,53	3,07	4,09	13,80	12,00	16,00	0,04	0,03	0,04	0,19	0,18	0,20	1,04	1,03	1,04	45,40	26	96	0,02	0	1	51,70	30	104	65902,90	65814	65989
RBFN(k=5;f=2;c=2)	91,85	88,24	95,15	8,15	4,85	11,76	31,90	19,00	46,00	0,08	0,05	0,12	0,28	0,22	0,34	1,09	1,05	1,13	1,30	1	2	0,01	0	1	3,50	3	4	62833,90	62745	62920
RBFN(k=10;f=2;c=2)	nie udało się przeprowadzić treningu																													
RBFN(k=15;f=2;c=2)	nie udało się przeprowadzić treningu																													
RBFN(k=20;f=2;c=2)	nie udało się przeprowadzić treningu																													
PNN(sigma=0,1)	100,00	100,00	100,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,01	1,00	1,01	0,00	0	0	1,01	0	10	396,50	389	405	65623,90	65533	65712
PNN(sigma=1)	90,88	90,05	92,09	9,12	7,91	9,95	27,25	24,64	29,47	0,07	0,06	0,08	0,26	0,25	0,27	1,74	1,68	1,83	0,00	0	0	1,00	0	6	392,90	384	402	65623,90	65533	65712
PNN(sigma=10)	89,37	88,78	90,56	10,63	9,44	11,22	95,63	95,50	95,90	0,24	0,24	0,24	0,49	0,49	0,49	1,44E+04	1,39E+04	1,52E+04	0,00	0	0	1,05	0	184	412,10	387	594	65623,90	65533	65712
PNN(sigma=100)	89,35	88,75	90,56	10,65	9,44	11,25	97,85	97,73	97,98	0,25	0,25	0,25	0,50	0,50	0,50	1,47E+08	1,42E+08	1,55E+08	0,00	0	0	1,00	0	13	390,10	382	406	65623,90	65533	65712

Tablica 6.18: Congressional Voting Records, dane znormalizowane, 10-fold cross-validation, walidacja

Algorytm	Q_COR(avg)	Q_COR(min)	Q_COR(max)	Q_ERR(avg)	Q_ERR(min)	Q_ERR(max)	Q_SSE(avg)	Q_SSE(min)	Q_SSE(max)	Q_MSE(avg)	Q_MSE(min)	Q_MSE(max)	Q_RMSE(avg)	Q_RMSE(min)	Q_RMSE(max)	Q_ARV(avg)	Q_ARV(min)	Q_ARV(max)	E_TTR(avg)	E_TTR(min)	E_TTR(max)	E_TCL(avg)	E_TCL(min)	E_TCL(max)	E_TOT(avg)	E_TOT(min)	E_TOT(max)	E_MEM(avg)	E_MEM(min)	E_MEM(max)
SONN-3	94,71	90,91	97,73	5,29	2,27	9,09	2,58	1	4,00	0,05	0,02	0,09	0,22	0,15	0,30	1,06	1,02	1,10	96,00	87	128	0,03	0	1	97,40	88	129	829865,70	81893	834636
KNN(k=1)	92,88	88,64	97,73	7,12	2,27	11,36	3,10	1,00	5,00	0,07	0,02	0,11	0,26	0,15	0,34	1,08	1,02	1,13	0,00	0	0	1,26	1	14	54,70	50	68	65195,90	65103	65286
KNN(k=3)	92,39	83,72	97,73	7,61	2,27	16,28	2,08	0,44	3,56	0,05	0,01	0,08	0,21	0,10	0,29	1,13	1,07	1,23	0,00	0	0	1,34	1	7	58,20	56	68	65195,90	65103	65286
KNN(k=5)	91,93	81,40	100,00	8,07	0,00	18,60	2,06	0,16	3,96	0,05	0,00	0,09	0,21	0,06	0,30	1,16	1,08	1,34	0,00	0	0	1,72	1	9	75,00	71	94	65195,90	65103	65286
KNN(k=10)	90,32	81,40	97,67	9,68	2,33	18,60	2,11	0,11	5,16	0,05	0,00	0,12	0,21	0,05	0,35	1,20	1,09	1,35	0,00	0	0	2,49	2	30	108,40	101	148	65195,90	65103	65286
KNN(k=15)	89,20	76,74	100,00	10,80	0,00	23,26	2,21	0,11	5,28	0,05	0,00	0,12	0,21	0,05	0,35	1,23	1,11	1,36	0,00	0	0	3,24	2	36	140,70	130	188	65195,90	65103	65286
KNN(k=30)	88,94	79,07	95,45	11,06	4,55	20,93	2,45	0,08	6,03	0,06	0,00	0,14	0,22	0,04	0,37	1,27	1,15	1,40	0,00	0	0	5,46	5	15	237,70	227	252	65195,90	65103	65286
SVML(c=1;e=0,1)	96,31	88,37	100,00	3,69	0,00	11,63	1,60	0,00	5,00	0,04	0,00	0,12	0,16	0,00	0,34	NaN	NaN	1,13	237,00	174	295	0,01	0	1	237,30	174	295	78567,50	78472	78663
SVML(c=10;e=0,1)	95,85	90,70	100,00	4,15	0,00	9,30	1,80	0,00	4,00	0,04	0,00	0,09	0,17	0,00	0,31	NaN	NaN	1,10	2182,10	1290	2852	0,01	0	1	2182,50	1290	2852	78567,50	78472	78663
SVML(c=50;e=0,1)	95,16	86,05	100,00	4,84	0,00	13,95	2,10	0,00	6,00	0,05	0,00	0,14	0,19	0,00	0,37	NaN	NaN	1,16	8191,40	2939	13771	0,00	0	1	8191,50	2939	13772	78567,50	78472	78663
SVML(c=100;e=0,1)	95,39	88,37	100,00	4,61	0,00	11,63	2,00	0,00	5,00	0,05	0,00	0,12	0,20	0,00	0,34	NaN	NaN	1,13	8890,40	4675	14751	0,00	0	1	8890,50	4675	14751	78567,50	78472	78663
SVML(c=20;e=0,1)	96,08	88,37	100,00	3,92	0,00	11,63	1,70	0,00	5,00	0,04	0,00	0,12	0,17	0,00	0,34	NaN	NaN	1,13	3742,10	1680	7297	0,01	0	1	3742,50	1680	7297	78567,50	78472	78663
SVML(c=20;e=0,01)	95,85	86,05	100,00	4,15	0,00	13,95	1,80	0,00	6,00	0,04	0,00	0,14	0,16	0,00	0,37	NaN	NaN	1,16	3797,50	1561	9125	0,01	0	1	3797,80	1562	9125	78567,50	78472	78663
SVML(c=20;e=0,001)	96,32	90,70	100,00	3,68	0,00	9,30	1,60	0,00	4,00	0,04	0,00	0,09	0,17	0,00	0,31	NaN	NaN	1,10	5915,20	3942	9352	0,00	0	1	5915,40	3942	9352	78567,50	78472	78663
SVMR(c=10;s=0,5;e=0,1)	86,21	75,00	93,02	13,79	6,98	25,00	6,00	3,00	11,00	0,14	0,07	0,25	0,36	0,26	0,50	1,17	1,08	1,33	3152,70	2406	4647	0,32	0	2	3166,40	2419	4661	78364,50	78269	78460
SVMR(c=10;s=1;e=0,1)	95,86	93,02	100,00	4,14	0,00	6,98	1,80	0,00	3,00	0,04	0,00	0,07	0,19	0,00	0,26	NaN	NaN	1,08	3687,50	2578	5484	0,19	0	1	3695,70	2586	5492	78364,50	78269	78460
SVMR(c=10;s=5;e=0,1)	95,85	90,70	100,00	4,15	0,00	9,30	1,80	0,00	4,00	0,04	0,00	0,09	0,18	0,00	0,31	NaN	NaN	1,10	1783,10	1209	2425	0,13	0	1	1788,60	1214	2430	78364,50	78269	78460
SVMR(c=10;s=10;e=0,1)	95,39	90,70	100,00	4,61	0,00	9,30	2,00	0,00	4,00	0,05	0,00	0,09	0,19	0,00	0,31	NaN	NaN	1,10	556,30	393	769	0,15	0	2	562,80	400	775	78364,50	78269	78460
SVMP(c=10;d=3;e=0,01)	93,11	86,05	97,67	6,89	2,33	13,95	3,00	1,00	6,00	0,07	0,02	0,14	0,26	0,15	0,37	1,08	1,02	1,16	89473,80	53106	172734	0,10	0	1	89478,10	53110	172739	78678,50	78583	78774
SVMP(c=10;d=4;e=0,01)	60,70	30,23	76,74	39,30	23,26	69,77	20,60	10,00	44,00	0,47	0,23	1,00	0,67	0,48	1,00	∞	1,30	∞	21163,10	17628	22667	0,09	0	3	21167,00	17632	22671	78678,50	78583	78774
SVMP(c=10;d=6;e=0,01)	38,64	25,00	50,00	61,36	50,00	75,00	43,50	43,00	44,00	1,00	1,00	1,00	1,00	1,00	1,00	∞	∞	∞	18317,70	17991	19253	0,08	0	1	18321,10	17994	19257	78678,50	78583	78774
RBFN(k=5;f=2;c=1)	89,87	76,74	95,35	10,13	4,65	23,26	4,40	2,00	10,00	0,10	0,05	0,23	0,31	0,22	0,48	1,12	1,05	1,30	1,40	1	2	0,00	0	1	1,60	1	3	62811,90	62723	62898
RBFN(k=10;f=2;c=1)	93,77	83,72	100,00	6,23	0,00	16,28	2,70	0,00	7,00	0,06	0,00	0,16	0,23	0,00	0,40	NaN	NaN	1,19	2,70	2	3	0,01	0	1	3,20	2	4	62916,90	62828	63003
RBFN(k=15;f=2;c=1)	nie udało się przeprowadzić treningu																													
RBFN(k=20;f=2;c=1)	nie udało się przeprowadzić treningu																													
RBFN(k=5;f=1;c=2)	91,48	79,07	100,00	8,52	0,00	20,93	3,70	0,00	9,00	0,09	0,00	0,21	0,27	0,00	0,46	NaN	NaN	1,26	10,70	7	16	0,01	0	1	11,10	7	16	63517,90	63429	63604
RBFN(k=10;f=1;c=2)	94,69	83,72	100,00	5,31	0,00	16,28	2,30	0,00	7,00	0,05	0,00	0,16	0,20	0,00	0,40	NaN	NaN	1,19	25,10	16	35	0,00	0	1	25,30	16	36	64312,90	64224	64399
RBFN(k=15;f=1;c=2)	96,53	90,70	100,00	3,47	0,00	9,30	1,50	0,00	4,00	0,03	0,00	0,09	0,13	0,00	0,31	NaN	NaN	1,10	28,40	22	40	0,01	0	1	29,00	22	41	65107,90	65019	65194
RBFN(k=20;f=1;c=2)	95,17	90,70	100,00	4,83	0,00	9,30	2,10	0,00	4,00	0,05	0,00	0,09	0,19	0,00	0,31	NaN	NaN	1,10	45,40	26	96	0,02	0	1	46,40	27	97	65902,90	65814	65989
RBFN(k=5;f=2;c=2)	91,95	83,72	95,45	8,06	4,55	16,28	3,50	2,00	7,00	0,08	0,05	0,16	0,28	0,21	0,40	1,09	1,05	1,19	1,30	1	2	0,00	0	1	1,50	1	2	62833,90	62745	62920
RBFN(k=10;f=2;c=2)	nie udało się przeprowadzić treningu																													
RBFN(k=15;f=2;c=2)	nie udało się przeprowadzić treningu																													
RBFN(k=20;f=2;c=2)	nie udało się przeprowadzić treningu																													
PNN(sigma=0,1)	93,10	88,37	97,73	6,90	2,27	11,63	2,72	1,00	4,05	0,06	0,02	0,09	0,24	0,15	0,30	1,08	1,02	1,13	0,00	0	0	1,00	0	4	43,60	40	46	65623,90	65533	65712
PNN(sigma=1)	90,32	76,74	100,00	9,68	0,00	23,26	3,27	0,62	6,54	0,08	0,01	0,15	0,27	0,12	0,39	1,87	1,69	2,66	0,00	0	0	1,00	0	4	43,60	41	47	65623,90	65533	65712
PNN(sigma=10)	88,95	76,74	97,67	11,05	2,33	23,26	10,63	10,44	10,76	0,24	0,24	0,25	0,49	0,49	0,50	1,82E+04	7,83E+03	5,28E+04	0,00	0	0	1,01	0	4	44,10	41	45	65623,90	65533	65712
PNN(sigma=100)	88,95	76,74	97,67	11,05	2,33	23,26	10,87	10,75	11,00	0,25	0,25	0,25	0,50	0,50	0,50	1,86E+08	7,95E+07	5,43E+08	0,00	0	0	1,00	0	4	43,50	41	46	65623,90	65533	65712

6.6. Porównania dla zbioru Car Evaluation

Tabele 6.19, 6.20 oraz 6.21 przedstawiają wyniki z przeprowadzonych porównań dla zbioru Car Evaluation. Ciekawą cechą tego zbioru jest to, że domyślnie składał się z cech o wartościach symbolicznych, które zostały zamienione na wartości liczbowe. Ponieważ dane symboliczne często występują w zbiorach danych warto zastanowić się nad wpływem ich transformacji na wartości liczbowe dla badanych metod klasyfikacji.

Tablica 6.19: Car Evaluation, dane znormalizowane, test na całym zbiorze treningowym

Algorytm	Q_COR	Q_ERR	Q_SSE	Q_MSE	Q_RMSE	Q_ARV	E_TTR	E_TCL	E_TOT	E_MEM
SONN-3	100,00	0,00	399,35	0,23	0,48	25,60	1331	0,21	1697	2091165
KNN(k=1)	100,00	0,00	0,00	0,00	0,00	NaN	2	5,44	9404	141155
KNN(k=3)	89,18	10,82	44,11	0,03	0,16	1,20	1	7,23	12500	141155
KNN(k=5)	87,33	12,67	56,96	0,03	0,18	1,34	0	8,40	14520	141155
KNN(k=10)	84,49	15,51	81,60	0,05	0,22	1,52	0	12,17	21028	141155
KNN(k=15)	81,94	18,06	93,81	0,05	0,23	1,70	0	15,66	27061	141155
KNN(k=30)	74,77	25,23	129,02	0,07	0,27	1,99	1	27,48	47487	141155
SVML(c=1;e=0,1)	81,13	18,87	517,97	0,30	0,55	21,36	2190	0,02	2229	279310
SVML(c=10;e=0,1)	81,83	18,17	604,03	0,35	0,59	48,21	23432	0,01	23451	279310
SVML(c=50;e=0,1)	80,96	19,04	594,79	0,34	0,59	47,90	287783	0,01	287796	279310
SVML(c=100;e=0,1)	82,12	17,88	607,78	0,35	0,59	52,52	937107	0,01	937119	279310
SVML(c=20;e=0,1)	82,52	17,48	601,87	0,35	0,59	48,99	61595	0,01	61612	279310
SVML(c=20;e=0,01)	82,47	17,53	612,45	0,35	0,60	51,03	30796	0,01	30814	279310
SVML(c=20;e=0,001)	82,18	17,82	609,04	0,35	0,59	48,37	480439	0,01	480461	279310
SVMR(c=10;s=0,5;e=0,1)	99,88	0,12	178,02	0,10	0,32	7,71	1670218	1,17	1672233	279157
SVMR(c=10;s=1;e=0,1)	98,03	1,97	258,10	0,15	0,39	8,57	1561742	1,21	1563829	279157
SVMR(c=10;s=5;e=0,1)	81,60	18,40	419,49	0,24	0,49	12,06	104226	1,59	106967	279157
SVMR(c=10;s=10;e=0,1)	75,06	24,94	341,93	0,20	0,44	3,15	52179	1,68	55074	279157
SVMP(c=10;d=3;e=0,01)	nieakceptowalny czas treningu									
SVMP(c=10;d=4;e=0,01)	nieakceptowalny czas treningu									
SVMP(c=10;d=6;e=0,01)	nieakceptowalny czas treningu									
RBFN(k=5;f=2;c=1)	81,54	18,46	285,48	0,17	0,41	2,73	82	0,01	102	128215
RBFN(k=10;f=2;c=1)	81,02	18,98	272,40	0,16	0,40	2,91	14	0,02	40	128400
RBFN(k=15;f=2;c=1)	82,70	17,30	278,89	0,16	0,40	3,59	21	0,02	61	128585
RBFN(k=20;f=2;c=1)	87,15	12,85	233,55	0,14	0,37	3,21	27	0,03	79	128770
RBFN(k=5;f=1;c=2)	72,74	27,26	383,05	0,22	0,47	2,75	105	0,01	130	128521
RBFN(k=10;f=1;c=2)	81,94	18,06	264,10	0,15	0,39	2,46	160	0,02	194	128996
RBFN(k=15;f=1;c=2)	83,62	16,38	237,65	0,14	0,37	2,63	249	0,03	295	129471
RBFN(k=20;f=1;c=2)	85,94	14,06	236,30	0,14	0,37	3,31	328	0,03	383	129946
RBFN(k=5;f=2;c=2)	80,38	19,62	287,12	0,17	0,41	2,74	7	0,01	21	128237
RBFN(k=10;f=2;c=2)	81,37	18,63	270,87	0,16	0,40	2,66	18	0,02	46	128422
RBFN(k=15;f=2;c=2)	84,14	15,86	246,38	0,14	0,38	3,07	24	0,02	63	128607
RBFN(k=20;f=2;c=2)	84,90	15,10	245,40	0,14	0,38	3,25	31	0,02	72	128792
PNN(sigma=0,1)	100,00	0,00	1,64	0,00	0,03	1,07	0	7,09	12259	136235
PNN(sigma=1)	47,51	52,49	909,64	0,53	0,73	364,31	0	7,13	12315	136235
PNN(sigma=10)	41,38	58,62	971,47	0,56	0,75	3,79E+06	0	7,05	12188	136235
PNN(sigma=100)	41,32	58,68	971,99	0,56	0,75	3,79E+10	0	7,10	12265	136235

Tablica 6.20: Car Evaluation, dane znormalizowane, 10-fold cross-validation, trening

Algorytm	Q_COR(avg)	Q_COR(min)	Q_COR(max)	Q_ERR(avg)	Q_ERR(min)	Q_ERR(max)	Q_SSE(avg)	Q_SSE(min)	Q_SSE(max)	Q_MSE(avg)	Q_MSE(min)	Q_MSE(max)	Q_RMSE(avg)	Q_RMSE(min)	Q_RMSE(max)	Q_ARV(avg)	Q_ARV(min)	Q_ARV(max)	E_TTR(avg)	E_TTR(min)	E_TTR(max)	E_TCL(avg)	E_TCL(min)	E_TCL(max)	E_TOT(avg)	E_TOT(min)	E_TOT(max)	E_MEM(avg)	E_MEM(min)	E_MEM(max)
SONN-3	100,00	100,00	100,00	0,00	0,00	0,00	357,36	348,76	367,28	0,23	0,22	0,24	0,48	0,47	0,49	27,64	25,83	29,68	1021,80	981	1077	0,20	0	35	1326,40	1273	1424	1889035,80	1875000	1902356
KNN(k=1)	100,00	100,00	100,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	NaN	NaN	NaN	0,50	0	1	4,54	3	141	7065,40	6771	7372	127069,10	127044	127131
KNN(k=3)	88,68	88,23	89,01	11,32	10,99	11,77	41,72	39,67	43,78	0,03	0,03	0,03	0,16	0,16	0,17	1,21	1,20	1,21	0,30	0	1	4,96	4	24	7714,30	7608	7909	127069,10	127044	127131
KNN(k=5)	86,22	85,21	87,07	13,78	12,93	14,79	54,90	53,44	58,08	0,04	0,03	0,04	0,19	0,19	0,19	1,34	1,34	1,35	0,30	0	1	6,61	5	39	10287,20	9919	10631	127069,10	127044	127131
KNN(k=10)	82,83	81,74	84,25	17,17	15,75	18,26	75,80	74,53	76,81	0,05	0,05	0,05	0,22	0,22	0,22	1,54	1,53	1,55	0,40	0	1	10,26	8	132	15955,60	15250	16308	127069,10	127044	127131
KNN(k=15)	79,79	78,71	81,30	20,21	18,70	21,29	87,84	86,83	88,71	0,06	0,06	0,06	0,24	0,24	0,24	1,73	1,72	1,73	0,20	0	1	13,80	10	1073	21462,00	20744	23215	127069,10	127044	127131
KNN(k=30)	72,40	70,61	74,34	27,60	25,66	29,39	123,22	121,85	124,35	0,08	0,08	0,08	0,28	0,28	0,28	2,02	2,01	2,04	0,10	0	1	25,42	18	107	39538,30	37606	41303	127069,10	127044	127131
SVML(c=1;e=0,1)	80,57	78,20	81,17	19,43	18,83	21,80	468,37	452,55	502,63	0,30	0,29	0,32	0,55	0,54	0,57	22,74	18,24	33,02	1742,70	1650	1843	0,01	0	4	1757,10	1665	1856	251834,80	251803	251962
SVML(c=10;e=0,1)	81,83	80,71	82,25	18,17	17,75	19,29	536,32	524,44	550,15	0,34	0,34	0,35	0,59	0,58	0,59	45,44	40,88	57,18	14457,30	13137	17195	0,01	0	3	14471,80	13150	17214	251834,80	251803	251962
SVML(c=50;e=0,1)	82,05	81,16	82,83	17,95	17,17	18,84	547,48	511,12	573,06	0,35	0,33	0,37	0,59	0,57	0,61	52,94	39,71	61,60	717951,65	578240	922379	0,01	0	20	717971,15	578258	922392	251834,80	251803	251962
SVML(c=100;e=0,1)	81,73	80,26	82,44	18,27	17,56	19,74	551,85	537,29	564,98	0,35	0,35	0,36	0,60	0,59	0,60	53,81	45,21	64,25	254308,92	189230	331287	0,01	0	31,2	254328,46	189244	331304	251834,80	251803	251962
SVML(c=20;e=0,1)	81,82	80,51	82,57	18,18	17,43	19,49	546,92	535,87	565,12	0,35	0,34	0,36	0,59	0,59	0,60	52,48	45,60	64,59	52113,52	39795,6	70978	0,01	0	15,6	52125,46	39811,2	70996	251834,80	251803	251962
SVML(c=20;e=0,01)	82,29	81,93	82,57	17,71	17,43	18,07	549,57	541,56	559,67	0,35	0,35	0,36	0,59	0,59	0,60	49,36	44,67	53,23	31711,20	21973	38866	0,01	0	57	31734,00	21990	38879	251834,80	251803	251962
SVML(c=20;e=0,001)	82,00	78,91	82,51	18,00	17,49	21,09	550,51	543,97	577,22	0,35	0,35	0,37	0,59	0,59	0,61	51,35	44,38	82,97	404011,50	246965	761130	0,01	0	17	404029,00	246975	761146	251834,80	251803	251962
SVMR(c=10;s=0,5;e=0,1)	99,88	99,68	100,00	0,12	0,00	0,32	159,05	153,38	163,49	0,10	0,10	0,11	0,32	0,31	0,32	7,29	7,00	7,62	1026966,10	884485	1183766	0,97	0	40	1028479,00	885875	1185209	251681,80	251650	251809
SVMR(c=10;s=1;e=0,1)	97,99	97,75	98,33	2,01	1,67	2,25	234,78	228,19	254,58	0,15	0,15	0,16	0,39	0,38	0,40	8,14	7,55	9,11	912072,25	709154	1288150	0,91	0	31	913485,85	710385	1289816	251681,80	251650	251809
SVMR(c=10;s=5;e=0,1)	81,08	80,45	81,87	18,92	18,14	19,55	368,65	340,98	400,31	0,24	0,22	0,26	0,49	0,47	0,51	10,76	8,58	14,47	47468,40	32899	80651	1,15	0	37	49257,90	34803	82300	251681,80	251650	251809
SVMR(c=10;s=10;e=0,1)	75,00	74,73	75,37	25,00	24,63	25,27	312,72	299,56	328,10	0,20	0,19	0,21	0,45	0,44	0,46	3,17	2,92	3,55	42336,60	31431	83334	1,21	0	47	44210,90	33532	85338	251681,80	251650	251809
SVMP(c=10;d=3;e=0,01)	niekceptowalny czas treningu																													
SVMP(c=10;d=4;e=0,01)	niekceptowalny czas treningu																													
SVMP(c=10;d=6;e=0,01)	niekceptowalny czas treningu																													
RBFN(k=5;f=2;c=1)	76,95	71,38	80,96	23,05	19,04	28,62	293,23	253,38	337,94	0,19	0,16	0,22	0,43	0,40	0,47	2,73	2,59	2,94	10,50	4	63	0,01	0	2	23,30	15	76	115511,50	115488	115567
RBFN(k=10;f=2;c=1)	81,98	80,91	83,42	18,02	16,58	19,09	243,96	232,06	253,46	0,16	0,15	0,16	0,40	0,39	0,40	2,98	2,79	3,33	13,30	8	18	0,01	0	4	32,80	27	38	115696,50	115673	115752
RBFN(k=15;f=2;c=1)	82,94	81,17	84,50	17,06	15,50	18,83	232,05	216,76	247,39	0,15	0,14	0,16	0,39	0,37	0,40	3,17	3,06	3,26	18,20	15	35	0,02	0	50	49,90	38	90	115881,50	115858	115937
RBFN(k=20;f=2;c=1)	85,60	83,92	87,52	14,40	12,48	16,08	218,34	208,93	227,11	0,14	0,13	0,15	0,37	0,37	0,38	3,45	3,30	3,73	22,70	21	25	0,02	0	16	55,70	48	72	116066,50	116043	116122
RBFN(k=5;f=1;c=2)	78,11	72,60	79,94	21,89	20,06	27,40	282,39	262,92	332,95	0,18	0,17	0,21	0,43	0,41	0,46	2,61	2,44	2,70	40,80	29	65	0,01	0	16	56,30	39	79	115817,50	115794	115873
RBFN(k=10;f=1;c=2)	81,51	79,61	83,09	18,49	16,91	20,39	241,96	224,57	264,97	0,16	0,14	0,17	0,39	0,38	0,41	2,71	2,48	2,93	150,70	128	188	0,02	0	144	189,20	148	299	116292,50	116269	116348
RBFN(k=15;f=1;c=2)	83,40	82,20	86,17	16,60	13,83	17,80	224,27	209,05	235,24	0,14	0,13	0,15	0,38	0,37	0,39	2,97	2,71	3,21	160,20	108	309	0,02	0	14	189,40	134	346	116767,50	116744	116823
RBFN(k=20;f=1;c=2)	85,87	85,22	87,15	14,13	12,85	14,78	211,01	201,60	217,49	0,14	0,13	0,14	0,37	0,36	0,37	3,30	3,16	3,47	220,80	125	335	0,02	0	15	258,90	163	373	117242,50	117219	117298
RBFN(k=5;f=2;c=2)	76,43	69,99	81,35	23,57	18,65	30,01	295,90	257,42	347,49	0,19	0,17	0,22	0,44	0,41	0,47	2,74	2,59	2,95	4,80	4	5	0,01	0	2	18,10	16	20	115533,50	115510	115589
RBFN(k=10;f=2;c=2)	81,98	80,71	83,22	18,02	16,78	19,29	244,99	240,06	254,03	0,16	0,15	0,16	0,40	0,39	0,40	2,98	2,76	3,21	9,80	9	11	0,01	0	4	28,10	24	33	115718,50	115695	115774
RBFN(k=15;f=2;c=2)	83,71	81,61	85,27	16,29	14,73	18,39	232,38	223,34	241,56	0,15	0,14	0,16	0,39	0,38	0,39	3,27	2,91	3,66	15,70	14	17	0,02	0	14	41,40	39	49	115903,50	115880	115959
RBFN(k=20;f=2;c=2)	85,20	83,67	86,95	14,80	13,05	16,33	219,23	213,48	230,02	0,14	0,14	0,15	0,38	0,37	0,38	3,35	3,05	3,60	22,70	20	24	0,03	0	115	64,40	50	166	116088,50	116065	116144
PNN(sigma=0,1)	100,00	100,00	100,00	0,00	0,00	0,00	1,31	1,18	1,44	0,00	0,00	0,00	0,03	0,03	0,03	1,07	1,06	1,07	0,00	0	0	5,23	4	138	8128,50	7686	8590	122840,30	122816	122899
PNN(sigma=1)	47,66	47,14	48,01	52,34	51,99	52,86	818,66	817,56	819,71	0,53	0,53	0,53	0,73	0,72	0,73	364,19	359,68	368,78	0,00	0	0	5,32	4	121	8269,90	7607	11218	122840,30	122816	122899
PNN(sigma=10)	41,60	41,29	42,16	58,40	57,84	58,71	874,32	874,20	874,77	0,56	0,56	0,56	0,75	0,75	0,75	3,79E+06	3,76E+06	3,82E+06	0,00	0	0	5,04	4	40	7839,60	7565	8061	122840,30	122816	122899
PNN(sigma=100)	41,53	41,22	42,03	58,47	57,97	58,78	874,80	874,68	875,25	0,56	0,56	0,56	0,75	0,75	0,75	3,79E+10	3,76E+10	3,82E+10	0,00	0	0	4,87	4	38	7573,30	7275	7755	122840,30	122816	122899

Tablica 6.21: Car Evaluation, dane znormalizowane, 10-fold cross-validation, walidacja

Algorytm	Q_COR(avg)	Q_COR(min)	Q_COR(max)	Q_ERR(avg)	Q_ERR(min)	Q_ERR(max)	Q_SSE(avg)	Q_SSE(min)	Q_SSE(max)	Q_MSE(avg)	Q_MSE(min)	Q_MSE(max)	Q_RMSE(avg)	Q_RMSE(min)	Q_RMSE(max)	Q_ARV(avg)	Q_ARV(min)	Q_ARV(max)	E_TTR(avg)	E_TTR(min)	E_TTR(max)	E_TCL(avg)	E_TCL(min)	E_TCL(max)	E_TOT(avg)	E_TOT(min)	E_TOT(max)	E_MEM(avg)	E_MEM(min)	E_MEM(max)
SONN-3	94,39	92,49	98,26	5,61	1,74	7,51	42,27	41,20	43,18	0,24	0,24	0,25	0,49	0,49	0,50	23,79	20,95	28,28	1021,80	981	1077	0,19	0	5	1055,30	1015	1112	1889035,80	1875000	1902356
KNN(k=1)	91,09	90,12	92,49	8,91	7,51	9,88	15,40	13,00	17,00	0,09	0,08	0,10	0,30	0,27	0,31	1,10	1,08	1,11	0,50	0	1	4,47	3	18	772,80	749	806	127069,10	127044	127131
KNN(k=3)	87,96	86,63	89,60	12,04	10,40	13,37	8,77	8,11	9,67	0,05	0,05	0,06	0,23	0,22	0,24	1,28	1,25	1,30	0,30	0	1	4,96	4	12	858,20	841	894	127069,10	127044	127131
KNN(k=5)	85,18	81,50	86,71	14,82	13,29	18,50	8,48	7,24	9,12	0,05	0,04	0,05	0,22	0,20	0,23	1,36	1,33	1,39	0,30	0	1	6,66	5	19	1150,40	1116	1204	127069,10	127044	127131
KNN(k=10)	82,41	78,61	84,39	17,59	15,61	21,39	9,45	8,61	10,03	0,05	0,05	0,06	0,23	0,22	0,24	1,55	1,51	1,58	0,40	0	1	10,37	8	44	1792,30	1703	1863	127069,10	127044	127131
KNN(k=15)	76,91	72,25	82,08	23,09	17,92	27,75	10,47	9,72	11,37	0,06	0,06	0,07	0,25	0,24	0,26	1,74	1,70	1,80	0,20	0	1	13,74	11	26	2373,90	2319	2436	127069,10	127044	127131
KNN(k=30)	70,78	68,21	75,72	29,22	24,28	31,79	14,11	13,19	15,20	0,08	0,08	0,09	0,29	0,28	0,30	2,02	1,95	2,08	0,10	0	1	25,57	18	108	4418,60	4186	4673	127069,10	127044	127131
SVML(c=1;e=0,1)	80,21	77,46	82,08	19,79	17,92	22,54	52,21	50,40	55,02	0,30	0,29	0,32	0,55	0,54	0,56	23,44	17,29	34,79	1742,70	1650	1843	0,02	0	8	1745,30	1659	1845	251834,80	251803	251962
SVML(c=10;e=0,1)	81,42	79,77	83,24	18,58	16,76	20,23	59,78	58,54	60,94	0,35	0,34	0,35	0,59	0,58	0,59	48,78	37,08	62,26	14457,30	13137	17195	0,01	0	1	14459,10	13138	17196	251834,80	251803	251962
SVML(c=50;e=0,1)	81,89	80,35	83,82	18,11	16,19	19,65	60,97	56,26	64,35	0,35	0,33	0,37	0,59	0,57	0,61	60,08	38,60	85,35	717951,65	578240	922379	0,01	0	3	717953,95	578243	922381	251834,80	251803	251962
SVML(c=100;e=0,1)	81,60	79,07	83,24	18,40	16,76	20,93	61,45	59,06	63,19	0,36	0,34	0,37	0,60	0,58	0,60	59,06	37,91	77,48	254308,92	189230	331287	0,01	0	1	254310,12	189231	331290	251834,80	251803	251962
SVML(c=20;e=0,1)	81,13	79,19	83,24	18,87	16,76	20,81	60,92	58,24	63,95	0,35	0,34	0,37	0,59	0,58	0,61	55,39	31,34	73,14	52113,52	39795,6	70978	0,00	0	2	52114,22	39795,6	70981	251834,80	251803	251962
SVML(c=20;e=0,01)	82,00	80,35	82,66	18,00	17,34	19,65	61,12	58,54	62,82	0,35	0,34	0,36	0,59	0,58	0,60	51,47	37,97	68,74	31711,20	21973	38866	0,01	0	3	31713,10	21974	38869	251834,80	251803	251962
SVML(c=20;e=0,001)	81,66	78,61	83,24	18,34	16,76	21,39	61,33	58,87	63,45	0,35	0,34	0,37	0,60	0,58	0,61	57,07	31,05	109,74	404011,50	246965	761130	0,01	0	1	404013,40	246967	761131	251834,80	251803	251962
SVMR(c=10;s=0,5;e=0,1)	99,02	98,26	100,00	0,98	0,00	1,74	18,68	17,98	19,33	0,11	0,10	0,11	0,33	0,32	0,33	6,93	5,93	7,69	1026966,10	884485	1183766	0,96	0	36	1027132,70	884640	1183926	251681,80	251650	251809
SVMR(c=10;s=1;e=0,1)	96,82	94,77	98,84	3,18	1,16	5,23	26,48	24,86	28,48	0,15	0,14	0,16	0,39	0,38	0,41	7,92	7,34	8,78	912072,25	709154	1288150	0,94	0	38	912234,75	709293	1288416	251681,80	251650	251809
SVMR(c=10;s=5;e=0,1)	81,31	79,77	82,66	18,69	17,34	20,23	41,09	38,92	43,26	0,24	0,23	0,25	0,49	0,47	0,50	10,81	8,51	13,64	47468,40	32899	80651	1,12	0	16	47662,30	33104	80834	251681,80	251650	251809
SVMR(c=10;s=10;e=0,1)	74,94	73,26	77,46	25,06	22,54	26,74	34,95	32,87	36,69	0,20	0,19	0,21	0,45	0,44	0,46	3,16	2,87	3,64	42336,60	31431	83334	1,25	0	17	42553,00	31780	83551	251681,80	251650	251809
SVMP(c=10;d=3;e=0,01)	niekceptowalny czas treningu																													
SVMP(c=10;d=4;e=0,01)	niekceptowalny czas treningu																													
SVMP(c=10;d=6;e=0,01)	niekceptowalny czas treningu																													
RBFN(k=5;f=2;c=1)	77,14	70,52	82,66	22,86	17,34	29,48	32,70	28,89	38,54	0,19	0,17	0,22	0,43	0,41	0,47	2,77	2,64	2,98	10,50	4	63	0,01	0	1	11,90	5	63	115511,50	115488	115567
RBFN(k=10;f=2;c=1)	81,71	79,77	83,82	18,29	16,19	20,23	27,17	24,94	28,66	0,16	0,14	0,17	0,40	0,38	0,41	2,99	2,79	3,27	13,30	8	18	0,02	0	6	17,00	10	25	115696,50	115673	115752
RBFN(k=15;f=2;c=1)	83,10	79,19	84,97	16,90	15,03	20,81	26,03	24,28	28,18	0,15	0,14	0,16	0,39	0,37	0,40	3,22	3,05	3,43	18,20	15	35	0,02	0	5	21,50	17	38	115881,50	115858	115937
RBFN(k=20;f=2;c=1)	85,24	82,66	87,86	14,76	12,14	17,34	24,56	23,33	26,45	0,14	0,13	0,15	0,38	0,37	0,39	3,47	3,21	3,87	22,70	21	25	0,02	0	1	25,80	23	29	116066,50	116043	116122
RBFN(k=5;f=1;c=2)	78,36	75,72	81,50	21,64	18,50	24,28	31,23	28,70	34,69	0,18	0,17	0,20	0,43	0,41	0,45	2,63	2,50	2,69	40,80	29	65	0,01	0	2	42,20	31	66	115817,50	115794	115873
RBFN(k=10;f=1;c=2)	81,77	77,33	83,24	18,23	16,76	22,67	27,06	25,04	28,72	0,16	0,14	0,17	0,40	0,38	0,41	2,69	2,55	2,81	150,70	128	188	0,01	0	1	152,80	130	191	116292,50	116269	116348
RBFN(k=15;f=1;c=2)	83,45	77,33	86,13	16,55	13,87	22,67	25,10	23,82	27,61	0,15	0,14	0,16	0,38	0,37	0,40	2,97	2,73	3,16	160,20	108	309	0,02	0	3	163,70	111	313	116767,50	116744	116823
RBFN(k=20;f=1;c=2)	85,30	83,14	86,71	14,70	13,29	16,86	24,06	22,95	26,57	0,14	0,13	0,15	0,37	0,36	0,39	3,27	3,12	3,64	220,80	125	335	0,02	0	1	225,00	128	339	117242,50	117219	117298
RBFN(k=5;f=2;c=2)	76,56	69,77	82,08	23,44	17,92	30,23	32,85	28,78	39,48	0,19	0,17	0,23	0,44	0,41	0,48	2,72	2,54	2,92	4,80	4	5	0,01	0	1	6,30	6	7	115533,50	115510	115589
RBFN(k=10;f=2;c=2)	81,77	79,65	84,39	18,23	15,61	20,35	27,28	25,08	29,22	0,16	0,15	0,17	0,40	0,38	0,41	2,96	2,76	3,15	9,80	9	11	0,01	0	1	11,90	10	13	115718,50	115695	115774
RBFN(k=15;f=2;c=2)	83,62	79,77	87,79	16,38	12,21	20,23	25,57	23,79	27,20	0,15	0,14	0,16	0,38	0,37	0,40	3,29	2,96	3,59	15,70	14	17	0,02	0	1	18,80	17	20	115903,50	115880	115959
RBFN(k=20;f=2;c=2)	84,89	81,98	87,28	15,11	12,72	18,02	24,51	23,05	26,26	0,14	0,13	0,15	0,38	0,37	0,39	3,31	2,94	3,54	22,70	20	24	0,02	0	4	26,10	22	29	116088,50	116065	116144
PNN(sigma=0,1)	92,88	89,60	95,38	7,12	4,62	10,40	10,41	7,98	14,26	0,06	0,05	0,08	0,24	0,21	0,29	1,20	1,16	1,24	0,00	0	0	5,17	4	139	894,10	845	987	122840,30	122816	122899
PNN(sigma=1)	47,86	46,24	48,55	52,14	51,45	53,76	90,99	90,50	91,53	0,53	0,52	0,53	0,73	0,72	0,73	365,75	343,12	399,07	0,00	0	0	5,08	4	38	877,80	824	966	122840,30	122816	122899
PNN(sigma=10)	41,49	39,31	43,35	58,51	56,65	60,69	97,15	96,70	97,26	0,56	0,56	0,56	0,75	0,75	0,75	3,81E+06	3,51E+06	4,25E+06	0,00	0	0	5,06	4	28	873,70	846	949	122840,30	122816	122899
PNN(sigma=100)	41,38	38,73	42,77	58,62	57,23	61,27	97,20	96,75	97,31	0,56	0,56	0,56	0,75	0,75	0,75	3,81E+10	3,51E+10	4,25E+10	0,00	0	0	5,08	4	51	878,40	800	1281	122840,30	122816	122899

Warto zauważyć, że dla zbioru Car Evaluation metoda SONN osiągnęła korzystne rezultaty w zakresie poprawności klasyfikacji nieznanych wzorców. Może to świadczyć o wyjątkowej skuteczności algorytmu binaryzacji ADLBCA dla zbiorów, w których wartości symboliczne zostały zamienione na wartości liczbowe.

Poza skutecznością klasyfikacji sieci SONN warto również zauważyć, że ta metoda dla zbioru Car Evaluation potrafiła w stosunkowo krótkim czasie przeprowadzić proces tworzenia klasyfikatora (warto porównać z algorytmami SVML, SVMR oraz SVMP).

6.7. Porównania dla zbioru Ionosphere

Tablica 6.22: Ionosphere, dane znormalizowane, test na całym zbiorze treningowym

Algorytm	Q_COR	Q_ERR	Q_SSE	Q_MSE	Q_RMSE	Q_ARV	E_TTR	E_TCL	E_TOT	E_MEM
SONN-3	94,30	5,70	8,50	0,02	0,16	1,09	138	0,05	155	400230
KNN(k=1)	100,00	0,00	0,00	0,00	0,00	NaN	1	1,55	545	106298
KNN(k=3)	86,32	13,68	17,33	0,05	0,22	1,19	0	1,94	682	106298
KNN(k=5)	83,19	16,81	25,48	0,07	0,27	1,25	0	2,37	832	106298
KNN(k=10)	79,49	20,51	34,32	0,10	0,31	1,34	0	3,87	1357	106298
KNN(k=15)	75,50	24,50	39,44	0,11	0,34	1,41	0	5,11	1794	106298
KNN(k=30)	70,66	29,34	50,26	0,14	0,38	1,59	0	9,42	3305	106298
SVML(c=1;e=0,1)	92,02	7,98	28,00	0,08	0,28	1,09	523	0,06	545	121648
SVML(c=10;e=0,1)	93,45	6,55	23,00	0,07	0,26	1,07	6784	0,01	6788	121648
SVML(c=50;e=0,1)	92,88	7,12	25,00	0,07	0,27	1,08	25616	0,01	25618	121648
SVML(c=100;e=0,1)	92,88	7,12	25,00	0,07	0,27	1,08	54561	0,01	54564	121648
SVML(c=20;e=0,1)	92,59	7,41	26,00	0,07	0,27	1,08	12465	0,01	12468	121648
SVML(c=20;e=0,01)	93,16	6,84	24,00	0,07	0,26	1,07	12211	0,01	12213	121648
SVML(c=20;e=0,001)	93,73	6,27	22,00	0,06	0,25	1,07	25017	0,01	25020	121648
SVMR(c=10;s=0,5;e=0,1)	100,00	0,00	0,00	0,00	0,00	NaN	3797	0,43	3949	121157
SVMR(c=10;s=1;e=0,1)	99,43	0,57	2,00	0,01	0,08	1,01	12864	0,25	12952	121157
SVMR(c=10;s=5;e=0,1)	94,59	5,41	19,00	0,05	0,23	1,06	6378	0,31	6488	121157
SVMR(c=10;s=10;e=0,1)	90,03	9,97	35,00	0,10	0,32	1,11	1538	0,44	1692	121157
SVMP(c=10;d=3;e=0,01)	97,72	2,28	8,00	0,02	0,15	1,02	2181079	0,19	2181146	121471
SVMP(c=10;d=4;e=0,01)	79,77	20,23	71,00	0,20	0,45	1,25	15232	0,09	15265	121471
SVMP(c=10;d=6;e=0,01)	64,10	35,90	351,00	1,00	1,00	∞	15980	0,08	16008	121471
RBFN(k=5;f=2;c=1)	84,90	15,10	53,00	0,15	0,39	1,18	20	0,01	25	104238
RBFN(k=10;f=2;c=1)	85,47	14,53	51,00	0,15	0,38	1,17	5	0,02	12	104343
RBFN(k=15;f=2;c=1)	91,17	8,83	31,00	0,09	0,30	1,10	8	0,03	18	104448
RBFN(k=20;f=2;c=1)	91,45	8,55	30,00	0,09	0,29	1,09	11	0,04	24	104553
RBFN(k=5;f=1;c=2)	91,45	8,55	30,00	0,09	0,29	1,09	25	0,01	29	105664
RBFN(k=10;f=1;c=2)	92,59	7,41	26,00	0,07	0,27	1,08	43	0,02	50	107179
RBFN(k=15;f=1;c=2)	90,31	9,69	34,00	0,10	0,31	1,11	57	0,03	68	108694
RBFN(k=20;f=1;c=2)	92,59	7,41	26,00	0,07	0,27	1,08	71	0,03	83	110209
RBFN(k=5;f=2;c=2)	76,07	23,93	84,00	0,24	0,49	1,31	3	0,01	7	104260
RBFN(k=10;f=2;c=2)	90,60	9,40	33,00	0,09	0,31	1,10	5	0,02	13	104365
RBFN(k=15;f=2;c=2)	91,74	8,26	29,00	0,08	0,29	1,09	7	0,03	17	104470
RBFN(k=20;f=2;c=2)	91,45	8,55	30,00	0,09	0,29	1,09	10	0,04	23	104575
PNN(sigma=0,1)	100,00	0,00	0,10	0,00	0,02	1,01	0	1,51	530	107038
PNN(sigma=1)	78,35	21,65	54,54	0,16	0,39	6,36	0	1,36	477	107038
PNN(sigma=10)	65,24	34,76	87,26	0,25	0,50	3,83E+04	0	1,36	479	107038
PNN(sigma=100)	65,24	34,76	87,75	0,25	0,50	3,82E+08	0	1,60	561	107038

Zbiór Ionosphere został uwzględniony w porównaniach ze względu na obecność wzorców sprzecznych. Dla takiego przypadku sieci SONN nie potrafią skutecznie sklasyfikować wszystkich wzorców treningowych, co można zauważyć analizując tabele 6.22, 6.23 oraz 6.24. Mimo tego, algorytm SONN dla danych walidacyjnych testu krzyżowego uzyskał rezultaty porównywalne z innymi metodami klasyfikacji.

Tablica 6.23: Ionosphere, dane znormalizowane, 10-fold cross-validation, trening

Algorytm	Q_COR(avg)	Q_COR(min)	Q_COR(max)	Q_ERR(avg)	Q_ERR(min)	Q_ERR(max)	Q_SSE(avg)	Q_SSE(min)	Q_SSE(max)	Q_MSE(avg)	Q_MSE(min)	Q_MSE(max)	Q_RMSE(avg)	Q_RMSE(min)	Q_RMSE(max)	Q_ARV(avg)	Q_ARV(min)	Q_ARV(max)	E_TTR(avg)	E_TTR(min)	E_TTR(max)	E_TCL(avg)	E_TCL(min)	E_TCL(max)	E_TOT(avg)	E_TOT(min)	E_TOT(max)	E_MEM(avg)	E_MEM(min)	E_MEM(max)
SONN-3	97,28	94,62	100,00	2,72	0,00	5,38	6,08	0,00	15,00	0,02	0,00	0,05	0,13	0,00	0,22	NaN	NaN	1,08	79,70	69	125	0,03	0	15	89,30	74	149	349612,10	330369	365000
KNN(k=1)	100,00	100,00	100,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	NaN	NaN	NaN	0,20	0	2	1,01	0	48	318,50	309	376	95697,80	95426	95728
KNN(k=3)	85,82	84,49	87,97	14,18	12,03	15,51	15,90	14,44	17,89	0,05	0,05	0,06	0,22	0,21	0,24	1,20	1,18	1,23	0,10	0	1	1,12	1	19	353,90	346	393	95697,80	95426	95728
KNN(k=5)	82,40	80,06	86,39	17,60	13,61	19,94	23,22	20,84	26,56	0,07	0,07	0,08	0,27	0,26	0,29	1,27	1,24	1,31	0,10	0	1	1,36	1	25	430,00	418	451	95697,80	95426	95728
KNN(k=10)	78,32	74,68	80,70	21,68	19,30	25,32	31,76	28,92	37,65	0,10	0,09	0,12	0,32	0,30	0,35	1,36	1,31	1,44	0,00	0	0	1,89	1	16	595,70	586	625	95697,80	95426	95728
KNN(k=15)	75,25	70,89	77,85	24,75	22,15	29,11	36,40	33,26	42,80	0,12	0,11	0,14	0,34	0,32	0,37	1,43	1,37	1,48	0,00	0	0	2,30	2	8	726,90	716	735	95697,80	95426	95728
KNN(k=30)	69,39	67,41	71,52	30,61	28,48	32,59	46,54	43,67	52,49	0,15	0,14	0,17	0,38	0,37	0,41	1,62	1,58	1,65	0,00	0	0	4,00	3	26	1264,80	1244	1329	95697,80	95426	95728
SVML(c=1;e=0,1)	91,61	90,51	92,41	8,39	7,59	9,49	26,50	24,00	30,00	0,08	0,08	0,09	0,29	0,28	0,31	1,09	1,08	1,10	382,00	290	463	0,01	0	26	386,30	292	464	109889,50	109588	109923
SVML(c=10;e=0,1)	94,14	92,41	96,52	5,86	3,48	7,59	18,50	11,00	24,00	0,06	0,03	0,08	0,24	0,19	0,28	1,06	1,04	1,08	3303,30	2444	4128	0,01	0	1	3305,40	2445	4131	109889,50	109588	109923
SVML(c=50;e=0,1)	93,23	91,77	95,57	6,77	4,43	8,23	21,40	14,00	26,00	0,07	0,04	0,08	0,26	0,21	0,29	1,07	1,05	1,09	25488,20	14880	32895	0,01	0	1	25489,80	14882	32897	109889,50	109588	109923
SVML(c=100;e=0,1)	93,19	91,77	94,94	6,81	5,06	8,23	21,50	16,00	26,00	0,07	0,05	0,08	0,26	0,23	0,29	1,07	1,05	1,09	13756,80	8976	19194	0,01	0	14	13760,00	8978	19196	109889,50	109588	109923
SVML(c=20;e=0,1)	93,92	92,72	95,89	6,08	4,11	7,28	19,20	13,00	23,00	0,06	0,04	0,07	0,25	0,20	0,27	1,06	1,04	1,08	6131,60	3819	10444	0,01	0	1	6133,40	3820	10446	109889,50	109588	109923
SVML(c=20;e=0,01)	94,37	93,33	96,20	5,64	3,80	6,67	17,80	12,00	21,00	0,06	0,04	0,07	0,24	0,19	0,26	1,06	1,04	1,07	6290,70	3432	11129	0,01	0	1	6292,70	3434	11129	109889,50	109588	109923
SVML(c=20;e=0,001)	94,49	93,35	95,57	5,51	4,43	6,65	17,40	14,00	21,00	0,06	0,04	0,07	0,23	0,21	0,26	1,06	1,05	1,07	14904,70	5370	31463	0,01	0	1	14906,40	5372	31466	109889,50	109588	109923
SVMR(c=10;s=0,5;e=0,1)	100,00	100,00	100,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	NaN	NaN	NaN	3735,20	2425	4569	0,42	0	2	3866,40	2558	4697	109398,50	109097	109432
SVMR(c=10;s=1;e=0,1)	99,34	99,05	99,37	0,66	0,63	0,95	2,10	2,00	3,00	0,01	0,01	0,01	0,08	0,08	0,10	1,01	1,01	1,01	8005,00	5721	10718	0,23	0	3	8078,60	5795	10793	109398,50	109097	109432
SVMR(c=10;s=5;e=0,1)	93,99	93,35	94,62	6,01	5,38	6,65	19,00	17,00	21,00	0,06	0,05	0,07	0,25	0,23	0,26	1,06	1,06	1,07	3389,90	1608	5858	0,30	0	3	3484,50	1702	5949	109398,50	109097	109432
SVMR(c=10;s=10;e=0,1)	90,15	88,92	91,46	9,85	8,54	11,08	31,10	27,00	35,00	0,10	0,09	0,11	0,31	0,29	0,33	1,11	1,09	1,12	2060,60	953	4279	0,39	0	2	2184,30	1077	4404	109398,50	109097	109432
SVMP(c=10;d=3;e=0,01)	98,92	98,10	99,68	1,08	0,32	1,90	3,40	1,00	6,00	0,01	0,00	0,02	0,10	0,06	0,14	1,01	1,00	1,02	645060,70	210423	1086039	0,13	0	2	645101,202	10461	1086082	109712,50	109411	109746
SVMP(c=10;d=4;e=0,01)	78,03	63,29	81,33	21,97	18,67	36,71	89,40	59,00	316,00	0,28	0,19	1,00	0,51	0,43	1,00	∞	1,23	∞	14064,70	13172	15338	0,06	0	3	14084,80	13192	15359	109712,50	109411	109746
SVMP(c=10;d=6;e=0,01)	61,09	34,92	65,19	38,91	34,81	65,08	315,90	315,00	316,00	1,00	1,00	1,00	1,00	1,00	1,00	∞	∞	∞	15306,20	14991	15758	0,06	0	1	15324,90	15010	15776	109712,50	109411	109746
RBFN(k=5;f=2;c=1)	78,60	69,94	82,91	21,40	17,09	30,06	67,60	54,00	95,00	0,21	0,17	0,30	0,46	0,41	0,55	1,28	1,21	1,43	2,90	1	16	0,01	0	1	5,70	2	20	93918,60	93654	93948
RBFN(k=10;f=2;c=1)	89,43	87,03	91,46	10,57	8,54	12,97	33,40	27,00	41,00	0,11	0,09	0,13	0,32	0,29	0,36	1,12	1,09	1,15	2,90	2	3	0,02	0	1	8,30	7	9	94023,60	93759	94053
RBFN(k=15;f=2;c=1)	90,57	88,29	91,77	9,43	8,23	11,71	29,80	26,00	37,00	0,09	0,08	0,12	0,31	0,29	0,34	1,10	1,09	1,13	4,90	4	6	0,02	0	1	11,80	10	13	94128,60	93864	94158
RBFN(k=20;f=2;c=1)	91,93	90,19	93,04	8,07	6,96	9,81	25,50	22,00	31,00	0,08	0,07	0,10	0,28	0,26	0,31	1,09	1,07	1,11	6,80	6	7	0,03	0	1	16,40	14	18	94233,60	93969	94263
RBFN(k=5;f=1;c=2)	89,55	85,13	92,41	10,45	7,59	14,87	33,00	24,00	47,00	0,10	0,08	0,15	0,32	0,28	0,39	1,12	1,08	1,17	16,70	9	27	0,01	0	1	19,90	12	30	95344,60	95080	95374
RBFN(k=10;f=1;c=2)	91,55	90,51	92,70	8,45	7,30	9,49	26,70	23,00	30,00	0,08	0,07	0,09	0,29	0,27	0,31	1,09	1,08	1,10	31,90	25	55	0,02	0	1	36,90	29	61	96859,60	96595	96889
RBFN(k=15;f=1;c=2)	92,28	91,75	93,99	7,72	6,01	8,25	24,40	19,00	26,00	0,08	0,06	0,08	0,28	0,25	0,29	1,08	1,06	1,09	50,60	32	71	0,02	0	1	57,90	40	78	98374,60	98110	98404
RBFN(k=20;f=1;c=2)	92,50	91,14	93,35	7,50	6,65	8,86	23,70	21,00	28,00	0,08	0,07	0,09	0,27	0,26	0,30	1,08	1,07	1,10	54,60	36	85	0,03	0	1	64,60	45	96	99889,60	99625	99919
RBFN(k=5;f=2;c=2)	79,52	73,73	86,03	20,48	13,97	26,27	64,70	44,00	83,00	0,20	0,14	0,26	0,45	0,37	0,51	1,26	1,16	1,36	1,80	1	3	0,01	0	1	4,70	3	7	93940,60	93676	93970
RBFN(k=10;f=2;c=2)	87,84	83,54	91,46	12,16	8,54	16,46	38,40	27,00	52,00	0,12	0,09	0,16	0,35	0,29	0,41	1,14	1,09	1,20	3,00	3	3	0,02	0	1	8,30	7	9	94045,60	93781	94075
RBFN(k=15;f=2;c=2)	90,85	89,21	91,77	9,15	8,23	10,79	28,90	26,00	34,00	0,09	0,08	0,11	0,30	0,29	0,33	1,10	1,09	1,12	4,90	4	5	0,02	0	1	12,10	10	13	94150,60	93886	94180
RBFN(k=20;f=2;c=2)	91,42	90,51	92,09	8,58	7,91	9,49	27,10	25,00	30,00	0,09	0,08	0,09	0,29	0,28	0,31	1,09	1,09	1,10	6,80	6	7	0,03	0	1	16,60	14	18	94255,60	93991	94285
PNN(sigma=0,1)	100,00	100,00	100,00	0,00	0,00	0,00	0,07	0,00	0,09	0,00	0,00	0,00	0,01	0,00	0,02	1,01	1,01	1,02	0,00	0	0	0,98	0	3	309,60	300	322	96572,20	96304	96602
PNN(sigma=1)	78,92	77,22	80,38	21,08	19,62	22,78	48,65	46,80	50,15	0,15	0,15	0,16	0,39	0,38	0,40	6,38	5,96	6,81	0,00	0	0	0,99	0	7	311,60	300	345	96572,20	96304	96602
PNN(sigma=10)	65,37	64,24	66,35	34,63	33,65	35,76	78,53	78,31	78,59	0,25	0,25	0,25	0,50	0,50	0,50	3,84E+04	3,37E+04	4,22E+04	0,10	0	1	0,99	0	4	312,70	301	325	96572,20	96304	96602
PNN(sigma=100)	65,31	64,24	66,35	34,69	33,65	35,76	78,97	78,75	79,00	0,25	0,25	0,25	0,50	0,50	0,50	3,83E+08	3,37E+08	4,21E+08	0,00	0	0	0,99	0	4	311,30	300	324	96572,20	96304	96602

Tablica 6.24: Ionosphere, dane znormalizowane, 10-fold cross-validation, walidacja

Algorytm	Q_COR(avg)	Q_COR(min)	Q_COR(max)	Q_ERR(avg)	Q_ERR(min)	Q_ERR(max)	Q_SSE(avg)	Q_SSE(min)	Q_SSE(max)	Q_MSE(avg)	Q_MSE(min)	Q_MSE(max)	Q_RMSE(avg)	Q_RMSE(min)	Q_RMSE(max)	Q_ARV(avg)	Q_ARV(min)	Q_ARV(max)	E_TTR(avg)	E_TTR(min)	E_TTR(max)	E_TCL(avg)	E_TCL(min)	E_TCL(max)	E_TOT(avg)	E_TOT(min)	E_TOT(max)	E_MEM(avg)	E_MEM(min)	E_MEM(max)
SONN-3	86,62	77,14	94,29	13,38	5,71	22,86	4,78	2,75	7,50	0,14	0,08	0,21	0,37	0,28	0,46	1,19	1,12	1,32	79,70	69	125	0,03	0	1	80,70	70	126	349612,10	330369	365000
KNN(k=1)	86,90	74,29	100,00	13,10	0,00	25,71	4,60	0,00	9,00	0,13	0,00	0,26	0,34	0,00	0,51	NaN	NaN	1,35	0,20	0	2	0,99	0	3	34,80	33	38	95697,80	95426	95728
KNN(k=3)	83,48	74,29	91,43	16,52	8,57	25,71	3,97	1,22	6,44	0,11	0,03	0,18	0,33	0,19	0,43	1,25	1,09	1,40	0,10	0	1	1,10	1	2	38,70	38	40	95697,80	95426	95728
KNN(k=5)	80,35	68,57	91,43	19,65	8,57	31,43	4,08	1,56	6,64	0,12	0,04	0,19	0,33	0,21	0,44	1,33	1,09	1,50	0,10	0	1	1,34	1	2	47,20	45	49	95697,80	95426	95728
KNN(k=10)	76,92	65,71	85,71	23,08	14,29	34,29	4,52	2,35	7,49	0,13	0,07	0,21	0,35	0,26	0,46	1,41	1,15	1,62	0,00	0	0	1,87	1	3	65,60	64	68	95697,80	95426	95728
KNN(k=15)	74,08	65,71	85,71	25,92	14,29	34,29	4,84	2,77	7,56	0,14	0,08	0,22	0,37	0,28	0,46	1,47	1,22	1,70	0,00	0	0	2,29	2	4	80,30	79	82	95697,80	95426	95728
KNN(k=30)	67,26	57,14	80,00	32,74	20,00	42,86	5,66	3,25	8,35	0,16	0,09	0,24	0,40	0,30	0,49	1,66	1,43	1,88	0,00	0	0	3,98	3	9	139,70	137	144	95697,80	95426	95728
SVML(c=1;e=0,1)	87,17	77,14	91,43	12,83	8,57	22,86	4,50	3,00	8,00	0,13	0,09	0,23	0,35	0,29	0,48	1,15	1,09	1,30	382,00	290	463	0,01	0	1	382,40	290	464	109889,50	109588	109923
SVML(c=10;e=0,1)	89,44	77,14	97,14	10,56	2,86	22,86	3,70	1,00	8,00	0,11	0,03	0,23	0,31	0,17	0,48	1,12	1,03	1,30	3303,30	2444	4128	0,01	0	1	3303,50	2445	4128	109889,50	109588	109923
SVML(c=50;e=0,1)	88,88	77,14	94,29	11,12	5,71	22,86	3,90	2,00	8,00	0,11	0,06	0,23	0,32	0,24	0,48	1,13	1,06	1,30	25488,20	14880	32895	0,01	0	1	25488,60	14881	32895	109889,50	109588	109923
SVML(c=100;e=0,1)	88,02	74,29	94,29	11,98	5,71	25,71	4,20	2,00	9,00	0,12	0,06	0,26	0,34	0,24	0,51	1,14	1,06	1,35	13756,80	8976	19194	0,00	0	1	13756,90	8976	19195	109889,50	109588	109923
SVML(c=20;e=0,1)	88,87	77,14	97,14	11,13	2,86	22,86	3,90	1,00	8,00	0,11	0,03	0,23	0,32	0,17	0,48	1,13	1,03	1,30	6131,60	3819	10444	0,01	0	1	6131,90	3819	10444	109889,50	109588	109923
SVML(c=20;e=0,01)	88,87	77,14	97,14	11,13	2,86	22,86	3,90	1,00	8,00	0,11	0,03	0,23	0,32	0,17	0,48	1,13	1,03	1,30	6290,70	3432	11129	0,00	0	0	6290,70	3432	11129	109889,50	109588	109923
SVML(c=20;e=0,001)	89,16	74,29	97,14	10,84	2,86	25,71	3,80	1,00	9,00	0,11	0,03	0,26	0,31	0,17	0,51	1,13	1,03	1,35	14904,70	5370	31463	0,00	0	1	14904,80	5370	31463	109889,50	109588	109923
SVMR(c=10;s=0,5;e=0,1)	93,44	88,57	97,14	6,56	2,86	11,43	2,30	1,00	4,00	0,07	0,03	0,11	0,25	0,17	0,34	1,07	1,03	1,13	3735,20	2425	4569	0,48	0	2	3752,10	2439	4583	109398,50	109097	109432
SVMR(c=10;s=1;e=0,1)	95,15	88,57	100,00	4,85	0,00	11,43	1,70	0,00	4,00	0,05	0,00	0,11	0,19	0,00	0,34	NaN	NaN	1,13	8005,00	5721	10718	0,24	0	1	8013,50	5729	10726	109398,50	109097	109432
SVMR(c=10;s=5;e=0,1)	90,87	82,86	97,14	9,13	2,86	17,14	3,20	1,00	6,00	0,09	0,03	0,17	0,29	0,17	0,41	1,10	1,03	1,21	3389,90	1608	5858	0,29	0	1	3400,10	1617	5868	109398,50	109097	109432
SVMR(c=10;s=10;e=0,1)	87,75	77,14	97,14	12,25	2,86	22,86	4,30	1,00	8,00	0,12	0,03	0,23	0,34	0,17	0,48	1,14	1,03	1,30	2060,60	953	4279	0,39	0	2	2074,40	966	4293	109398,50	109097	109432
SVMP(c=10;d=3;e=0,01)	86,91	68,57	91,43	13,09	8,57	31,43	4,60	3,00	11,00	0,13	0,09	0,31	0,35	0,29	0,56	1,16	1,09	1,46	645060,70	210423	1086039	0,13	0	1	645065,40	210427	1086044	109712,50	109411	109746
SVMP(c=10;d=4;e=0,01)	78,90	65,71	86,11	21,10	13,89	34,29	9,90	5,00	35,00	0,28	0,14	1,00	0,50	0,37	1,00	∞	1,16	∞	14064,70	13172	15338	0,06	0	1	14066,90	13174	15341	109712,50	109411	109746
SVMP(c=10;d=6;e=0,01)	63,02	44,44	74,29	36,98	25,71	55,56	35,10	35,00	36,00	1,00	1,00	1,00	1,00	1,00	1,00	∞	∞	∞	15306,20	14991	15758	0,05	0	1	15308,10	14993	15760	109712,50	109411	109746
RBFN(k=5;f=2;c=1)	74,63	60,00	91,43	25,37	8,57	40,00	8,90	3,00	14,00	0,25	0,09	0,40	0,49	0,29	0,63	1,37	1,09	1,67	2,90	1	16	0,01	0	1	3,20	1	16	93918,60	93654	93948
RBFN(k=10;f=2;c=1)	86,63	80,00	94,29	13,37	5,71	20,00	4,70	2,00	7,00	0,13	0,06	0,20	0,36	0,24	0,45	1,16	1,06	1,25	2,90	2	3	0,01	0	1	3,30	2	4	94023,60	93759	94053
RBFN(k=15;f=2;c=1)	89,73	80,00	97,14	10,27	2,86	20,00	3,60	1,00	7,00	0,10	0,03	0,20	0,31	0,17	0,45	1,12	1,03	1,25	4,90	4	6	0,03	0	1	6,10	5	8	94128,60	93864	94158
RBFN(k=20;f=2;c=1)	91,44	82,86	97,14	8,56	2,86	17,14	3,00	1,00	6,00	0,09	0,03	0,17	0,28	0,17	0,41	1,10	1,03	1,21	6,80	6	7	0,03	0	1	7,80	7	8	94233,60	93969	94263
RBFN(k=5;f=1;c=2)	88,88	82,86	94,29	11,12	5,71	17,14	3,90	2,00	6,00	0,11	0,06	0,17	0,33	0,24	0,41	1,13	1,06	1,21	16,70	9	27	0,01	0	1	17,10	9	28	95344,60	95080	95374
RBFN(k=10;f=1;c=2)	90,59	82,86	94,44	9,41	5,56	17,14	3,30	2,00	6,00	0,09	0,06	0,17	0,30	0,24	0,41	1,11	1,06	1,21	31,90	25	55	0,02	0	1	32,50	26	56	96859,60	96595	96889
RBFN(k=15;f=1;c=2)	92,02	82,86	97,14	7,98	2,86	17,14	2,80	1,00	6,00	0,08	0,03	0,17	0,27	0,17	0,41	1,09	1,03	1,21	50,60	32	71	0,02	0	1	51,30	32	72	98374,60	98110	98404
RBFN(k=20;f=1;c=2)	92,29	85,71	97,22	7,71	2,78	14,29	2,70	1,00	5,00	0,08	0,03	0,14	0,27	0,17	0,38	1,09	1,03	1,17	54,60	36	85	0,03	0	1	55,70	38	86	99889,60	99625	99919
RBFN(k=5;f=2;c=2)	77,76	60,00	94,29	22,24	5,71	40,00	7,80	2,00	14,00	0,22	0,06	0,40	0,46	0,24	0,63	1,30	1,06	1,67	1,80	1	3	0,00	0	1	1,90	1	3	93940,60	93676	93970
RBFN(k=10;f=2;c=2)	86,60	74,29	100,00	13,40	0,00	25,71	4,70	0,00	9,00	0,13	0,00	0,26	0,34	0,00	0,51	NaN	NaN	1,35	3,00	3	3	0,01	0	1	3,20	3	4	94045,60	93781	94075
RBFN(k=15;f=2;c=2)	90,87	82,86	97,14	9,13	2,86	17,14	3,20	1,00	6,00	0,09	0,03	0,17	0,29	0,17	0,41	1,10	1,03	1,21	4,90	4	5	0,03	0	1	5,90	5	7	94150,60	93886	94180
RBFN(k=20;f=2;c=2)	90,02	77,14	94,29	9,98	5,71	22,86	3,50	2,00	8,00	0,10	0,06	0,23	0,31	0,24	0,48	1,11	1,06	1,30	6,80	6	7	0,03	0	1	7,80	6	9	94255,60	93991	94285
PNN(sigma=0,1)	88,04	74,29	100,00	11,96	0,00	25,71	3,90	0,09	8,00	0,11	0,00	0,23	0,31	0,05	0,48	1,17	1,03	1,35	0,00	0	0	0,98	0	2	34,50	32	38	96572,20	96304	96602
PNN(sigma=1)	73,50	62,86	88,57	26,50	11,43	37,14	6,06	4,86	7,34	0,17	0,14	0,21	0,41	0,37	0,46	6,21	5,30	7,36	0,00	0	0	0,97	0	2	34,20	32	40	96572,20	96304	96602
PNN(sigma=10)	64,98	54,29	74,29	35,02	25,71	45,71	8,73	8,68	8,96	0,25	0,25	0,25	0,50	0,50	0,50	3,89E+04	3,00E+04	5,75E+04	0,10	0	1	0,97	0	2	34,00	32	37	96572,20	96304	96602
PNN(sigma=100)	64,98	54,29	74,29	35,02	25,71	45,71	8,77	8,75	9,00	0,25	0,25	0,25	0,50	0,50	0,50	3,88E+08	2,99E+08	5,74E+08	0,00	0	0	1,02	0	3	35,70	33	41	96572,20	96304	96602

Analogicznie jak dla wcześniej analizowanego zbioru Yeast, metoda SONN wykazuje wyjątkową skuteczność dla wzorców, które zostały wykorzystane w procesie treningu oraz najgorsze rezultaty dla wzorców nieznanych. Warto zauważyć, że wspólnymi cechami zbiorów Yeast oraz Arrhythmia są stosunkowo duża ilość klas problemu klasyfikacji oraz wielkość zbioru.

Z powodu dużej ilości klas dla zbioru Arrhythmia można zauważyć, że osiem z czternastu odmian instancji algorytmu SVM nie utworzyło klasyfikatora w dopuszczalnym czasie.

6.8. Porównania dla zbioru Arrhythmia

Tablica 6.25: Arrhythmia, dane znormalizowane, test na całym zbiorze treningowym

Algorytm	Q_COR	Q_ERR	Q_SSE	Q_MSE	Q_RMSE	Q_ARV	E_TTR	E_TCL	E_TOT	E_MEM
SONN-3	100,00	0,00	341,52	0,76	0,87	5201,95	3730	0,31	3870	2877487
KNN(k=1)	100,00	0,00	0,00	0,00	0,00	NaN	2	3,26	1477	1022801
KNN(k=3)	51,33	48,67	82,67	0,18	0,43	2,20	0	3,32	1500	1022801
KNN(k=5)	49,78	50,22	112,64	0,25	0,50	2,61	0	3,67	1660	1022801
KNN(k=10)	48,45	51,55	139,75	0,31	0,56	2,94	0	4,37	1975	1022801
KNN(k=15)	45,13	54,87	152,59	0,34	0,58	3,13	0	4,97	2247	1022801
KNN(k=30)	44,91	55,09	168,43	0,37	0,61	3,42	0	7,22	3262	1022801
SVML(c=1;e=0,1)	85,84	14,16	280,82	0,62	0,79	159,73	37023	0,07	37056	1193794
SVML(c=10;e=0,1)	94,47	5,53	291,21	0,64	0,80	180,57	577838	0,04	577858	1193794
SVML(c=50;e=0,1)	97,79	2,21	305,82	0,68	0,82	204,68	3835668	0,04	3835685	1193794
SVML(c=100;e=0,1)	97,79	2,21	300,50	0,66	0,82	203,98	3121304	0,04	3121322	1193794
SVML(c=20;e=0,1)	95,58	4,42	297,01	0,66	0,81	150,96	1643353	0,04	1643373	1193794
SVML(c=20;e=0,01)	95,80	4,20	285,71	0,63	0,80	139,82	598151	0,05	598173	1193794
SVML(c=20;e=0,001)	96,46	3,54	291,17	0,64	0,80	164,99	1580671	0,05	1580694	1193794
SVMR(c=10;s=0,5;e=0,1)	100,00	0,00	57,70	0,13	0,36	29,72	824972	16,02	832212	1164754
SVMR(c=10;s=1;e=0,1)	100,00	0,00	114,87	0,25	0,50	13,35	289301	11,19	294360	1164754
SVMR(c=10;s=5;e=0,1)	82,96	17,04	260,05	0,58	0,76	88,33	165593	5,91	168266	1164754
SVMR(c=10;s=10;e=0,1)	73,89	26,11	258,78	0,57	0,76	52,20	22848	5,42	25297	1164754
SVMP(c=10;d=3;e=0,01)	nieakceptowalny czas treningu									
SVMP(c=10;d=4;e=0,01)	nieakceptowalny czas treningu									
SVMP(c=10;d=6;e=0,01)	nieakceptowalny czas treningu									
RBFN(k=5;f=2;c=1)	58,63	41,37	229,19	0,51	0,71	11,70	75	0,32	218	1020681
RBFN(k=10;f=2;c=1)	61,73	38,27	235,60	0,52	0,72	18,13	25	0,67	327	1021226
RBFN(k=15;f=2;c=1)	67,26	32,74	226,20	0,50	0,71	18,10	39	0,96	474	1021771
RBFN(k=20;f=2;c=1)	67,92	32,08	225,54	0,50	0,71	19,19	53	1,29	634	1022316
RBFN(k=5;f=1;c=2)	57,08	42,92	229,11	0,51	0,71	11,53	228	0,29	361	1031907
RBFN(k=10;f=1;c=2)	63,05	36,95	236,10	0,52	0,72	17,78	267	0,59	532	1043662
RBFN(k=15;f=1;c=2)	63,72	36,28	232,80	0,52	0,72	19,39	715	0,90	1122	1055417
RBFN(k=20;f=1;c=2)	65,93	34,07	230,11	0,51	0,71	21,06	620	1,23	1176	1067172
RBFN(k=5;f=2;c=2)	59,29	40,71	231,96	0,51	0,72	13,15	13	0,32	157	1020703
RBFN(k=10;f=2;c=2)	60,62	39,38	237,97	0,53	0,73	17,28	25	0,67	329	1021248
RBFN(k=15;f=2;c=2)	67,04	32,96	224,57	0,50	0,70	17,54	41	0,98	484	1021793
RBFN(k=20;f=2;c=2)	66,37	33,63	227,64	0,50	0,71	19,53	54	1,31	646	1022338
PNN(sigma=0,1)	100,00	0,00	0,00	0,00	0,00	1,00	0	8,32	3761	1025513
PNN(sigma=1)	74,34	25,66	275,27	0,61	0,78	14,60	0	8,16	3687	1025513
PNN(sigma=10)	54,42	45,58	384,19	0,85	0,92	4,17E+06	0	8,39	3790	1025513
PNN(sigma=100)	53,98	46,02	385,13	0,85	0,92	4,24E+10	0	8,11	3664	1025513

Tablica 6.26: Arrhythmia, dane znormalizowane, 10-fold cross-validation, trening

Algorytm	Q_COR(avg)	Q_COR(min)	Q_COR(max)	Q_ERR(avg)	Q_ERR(min)	Q_ERR(max)	Q_SSE(avg)	Q_SSE(min)	Q_SSE(max)	Q_MSE(avg)	Q_MSE(min)	Q_MSE(max)	Q_RMSE(avg)	Q_RMSE(min)	Q_RMSE(max)	Q_ARV(avg)	Q_ARV(min)	Q_ARV(max)	E_TTR(avg)	E_TTR(min)	E_TTR(max)	E_TCL(avg)	E_TCL(min)	E_TCL(max)	E_TOT(avg)	E_TOT(min)	E_TOT(max)	E_MEM(avg)	E_MEM(min)	E_MEM(max)
SONN-3	100,00	100,00	100,00	0,00	0,00	0,00	312,96	305,32	324,31	0,77	0,75	0,80	0,88	0,87	0,89	8049,71	1855,20	23176,96	3698,20	3155	4268	0,30	0	26	3819,40	3283	4379	2441421,00	2337196	2526346
KNN(k=1)	100,00	100,00	100,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	NaN	NaN	NaN	0,20	0	2	3,20	2	44	1301,30	1259	1361	920550,50	918741	921007
KNN(k=3)	50,49	48,16	51,84	49,51	48,16	51,84	74,90	72,00	78,56	0,18	0,18	0,19	0,43	0,42	0,44	2,23	2,13	2,36	0,00	0	0	3,40	2	20	1384,40	1348	1428	920550,50	918741	921007
KNN(k=5)	49,78	47,67	51,84	50,22	48,16	52,33	101,67	99,88	105,20	0,25	0,25	0,26	0,50	0,50	0,51	2,63	2,47	2,81	0,00	0	0	3,94	3	30	1603,10	1481	1779	920550,50	918741	921007
KNN(k=10)	47,20	43,98	49,88	52,80	50,12	56,02	126,60	123,87	129,97	0,31	0,30	0,32	0,56	0,55	0,57	2,97	2,74	3,21	0,00	0	0	4,36	3	106	1774,70	1664	2145	920550,50	918741	921007
KNN(k=15)	44,86	41,03	48,16	55,14	51,84	58,97	138,49	134,69	142,66	0,34	0,33	0,35	0,58	0,58	0,59	3,16	2,95	3,43	0,00	0	0	4,64	4	8	1889,20	1860	1926	920550,50	918741	921007
KNN(k=30)	44,08	39,31	47,67	55,92	52,33	60,69	152,62	148,76	156,75	0,38	0,37	0,39	0,61	0,60	0,62	3,46	3,22	3,81	0,00	0	0	6,85	6	11	2785,90	2731	2820	920550,50	918741	921007
SVML(c=1;e=0,1)	86,16	85,01	86,98	13,84	13,02	14,99	253,62	245,20	264,73	0,62	0,60	0,65	0,79	0,78	0,81	179,84	135,55	252,49	23977,90	20344	31013	0,05	0	15	23997,90	20363	31032	1078127,20	1076080	1078639
SVML(c=10;e=0,1)	95,65	94,59	96,80	4,35	3,20	5,41	258,62	251,65	263,78	0,64	0,62	0,65	0,80	0,79	0,81	179,08	147,26	200,13	313864,40	218388	505393	0,04	0	1	313882,40	218407	505411	1078127,20	1076080	1078639
SVML(c=50;e=0,1)	nieakceptowalny czas treningu																													
SVML(c=100;e=0,1)	nieakceptowalny czas treningu																													
SVML(c=20;e=0,1)	nieakceptowalny czas treningu																													
SVML(c=20;e=0,01)	97,22	96,07	97,79	2,78	2,21	3,93	258,31	251,04	262,34	0,64	0,62	0,64	0,80	0,79	0,80	158,63	128,21	183,38	633590,20	493459	878283	0,05	0	1	633608,90	493477	878302	1078127,20	1076080	1078639
SVML(c=20;e=0,001)	nieakceptowalny czas treningu																													
SVMR(c=10;s=0,5;e=0,1)	nieakceptowalny czas treningu																													
SVMR(c=10;s=1;e=0,1)	100,00	100,00	100,00	0,00	0,00	0,00	100,00	95,76	108,05	0,25	0,24	0,27	0,50	0,49	0,52	13,65	11,37	23,20	206479,90	172008	230106	10,02	8	18	210556,80	175895	234224	1049087,20	1047040	1049599
SVMR(c=10;s=5;e=0,1)	83,46	82,31	84,48	16,54	15,52	17,69	234,01	227,38	240,22	0,58	0,56	0,59	0,76	0,75	0,77	88,98	70,84	109,70	141432,70	101893	179725	5,53	4	13	143680,80	104161	181986	1049087,20	1047040	1049599
SVMR(c=10;s=10;e=0,1)	73,85	71,74	76,11	26,15	23,89	28,26	228,51	218,60	237,23	0,56	0,54	0,58	0,75	0,73	0,76	45,38	34,92	61,45	28762,30	19839	47045	5,94	4	13	31177,30	22076	49635	1049087,20	1047040	1049599
SVMP(c=10;d=3;e=0,01)	nieakceptowalny czas treningu																													
SVMP(c=10;d=4;e=0,01)	nieakceptowalny czas treningu																													
SVMP(c=10;d=6;e=0,01)	nieakceptowalny czas treningu																													
RBFN(k=5;f=2;c=1)	59,00	57,39	60,69	41,00	39,31	42,61	208,39	203,99	214,16	0,51	0,50	0,53	0,72	0,71	0,73	13,89	11,37	16,79	15,80	12	30	0,40	0	4	179,80	167	194	918792,10	916989	919247
RBFN(k=10;f=2;c=1)	62,44	60,69	64,13	37,56	35,87	39,31	211,47	206,96	215,51	0,52	0,51	0,53	0,72	0,71	0,73	18,35	16,68	20,09	28,10	26	30	0,81	0	4	355,60	342	374	919337,10	917534	919792
RBFN(k=15;f=2;c=1)	65,39	63,79	68,72	34,61	31,28	36,21	208,54	202,96	214,81	0,51	0,50	0,53	0,72	0,71	0,73	20,22	17,72	21,54	42,90	41	44	1,18	0	4	523,80	502	543	919882,10	918079	920337
RBFN(k=20;f=2;c=1)	67,40	64,78	69,78	32,60	30,22	35,22	205,43	198,99	211,77	0,51	0,49	0,52	0,71	0,70	0,72	21,15	18,49	24,28	59,70	57	65	1,58	1	7	702,90	682	741	920427,10	918624	920882
RBFN(k=5;f=1;c=2)	59,41	57,39	60,93	40,59	39,07	42,61	210,61	204,30	217,37	0,52	0,50	0,53	0,72	0,71	0,73	13,00	11,30	14,54	249,00	46	384	0,38	0	3	401,80	200	538	930018,10	928215	930473
RBFN(k=10;f=1;c=2)	61,92	59,61	64,86	38,08	35,14	40,39	210,74	205,27	218,89	0,52	0,50	0,54	0,72	0,71	0,73	17,03	15,72	20,08	298,40	106	589	0,75	0	4	602,30	406	894	941773,10	939970	942228
RBFN(k=15;f=1;c=2)	64,13	61,18	66,58	35,87	33,42	38,82	208,89	203,81	212,26	0,51	0,50	0,52	0,72	0,71	0,72	18,45	17,19	20,85	506,70	119	848	1,13	0	33	967,50	582	1301	953528,10	951725	953983
RBFN(k=20;f=1;c=2)	65,71	62,65	67,81	34,29	32,19	37,35	206,72	198,28	210,38	0,51	0,49	0,52	0,71	0,70	0,72	19,65	17,35	21,96	930,90	541	1685	1,48	1	8	1534,10	1150	2250	965283,10	963480	965738
RBFN(k=5;f=2;c=2)	59,76	58,23	60,84	40,24	39,16	41,77	209,57	203,82	215,22	0,52	0,50	0,53	0,72	0,71	0,73	13,54	11,70	16,33	14,50	14	16	0,42	0	28	183,80	167	207	918814,10	917011	919269
RBFN(k=10;f=2;c=2)	62,54	58,97	64,53	37,46	35,47	41,03	210,35	201,95	217,02	0,52	0,50	0,53	0,72	0,70	0,73	17,46	14,18	20,10	28,10	26	30	0,80	0	4	353,20	334	367	919359,10	917556	919814
RBFN(k=15;f=2;c=2)	65,66	63,39	67,57	34,34	32,43	36,61	209,03	204,99	211,57	0,51	0,50	0,52	0,72	0,71	0,72	20,56	18,67	22,06	42,70	40	45	1,18	0	6	523,20	497	548	919904,10	918101	920359
RBFN(k=20;f=2;c=2)	67,99	66,09	70,69	32,01	29,31	33,91	204,16	198,40	210,78	0,50	0,49	0,52	0,71	0,70	0,72	20,63	17,97	23,62	58,30	56	59	1,57	1	8	698,00	660	724	920449,10	918646	920904
PNN(sigma=0,1)	100,00	100,00	100,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	0,00	0	0	9,25	7	29	3764,80	3686	3833	923443,30	921637	923899
PNN(sigma=1)	74,85	62,41	83,29	25,15	16,71	37,59	245,48	239,43	252,31	0,60	0,59	0,62	0,78	0,77	0,79	14,06	13,30	14,85	0,00	0	0	9,29	7	43	3777,20	3707	3842	923443,30	921637	923899
PNN(sigma=10)	51,50	33,66	57,25	48,50	42,75	66,34	345,76	345,06	346,00	0,85	0,85	0,85	0,92	0,92	0,92	3,97E+06	3,62E+06	4,30E+06	0,00	0	0	9,38	7	54	3816,70	3730	3915	923443,30	921637	923899
PNN(sigma=100)	51,28	33,17	57,25	48,72	42,75	66,83	346,61	345,93	346,78	0,85	0,85	0,85	0,92	0,92	0,92	4,05E+10	3,69E+10	4,38E+10	0,00	0	0	9,43	7	34	3837,90	3770	3948	923443,30	921637	923899

Tablica 6.27: Arrhythmia, dane znormalizowane, 10-fold cross-validation, walidacja

Algorytm	Q_COR(avg)	Q_COR(min)	Q_COR(max)	Q_ERR(avg)	Q_ERR(min)	Q_ERR(max)	Q_SSE(avg)	Q_SSE(min)	Q_SSE(max)	Q_MSE(avg)	Q_MSE(min)	Q_MSE(max)	Q_RMSE(avg)	Q_RMSE(min)	Q_RMSE(max)	Q_ARV(avg)	Q_ARV(min)	Q_ARV(max)	E_TTR(avg)	E_TTR(min)	E_TTR(max)	E_TCL(avg)	E_TCL(min)	E_TCL(max)	E_TOT(avg)	E_TOT(min)	E_TOT(max)	E_MEM(avg)	E_MEM(min)	E_MEM(max)
SONN-3	43,55	31,11	52,17	56,45	47,83	68,89	35,95	34,83	37,02	0,80	0,77	0,81	0,89	0,88	0,90	1608,94	1006,05	2649,99	3698,20	3155	4268	0,28	0	6	3710,80	3167	4279	2441421,00	2337196	2526346
KNN(k=1)	53,55	46,67	62,22	46,45	37,78	53,33	21,00	17,00	24,00	0,46	0,38	0,53	0,68	0,61	0,73	1,89	1,61	2,14	0,20	0	2	3,18	2	9	144,10	135	150	920550,50	918741	921007
KNN(k=3)	48,90	33,33	60,00	51,10	40,00	66,67	18,03	13,11	23,44	0,40	0,29	0,52	0,63	0,54	0,72	2,52	2,11	2,89	0,00	0	0	3,53	2	27	159,40	141	226	920550,50	918741	921007
KNN(k=5)	50,86	44,44	58,70	49,14	41,30	55,56	17,54	13,32	21,48	0,39	0,30	0,48	0,62	0,54	0,69	2,80	2,44	3,26	0,00	0	0	3,86	3	28	174,70	151	203	920550,50	918741	921007
KNN(k=10)	44,45	31,11	52,17	55,55	47,83	68,89	17,54	14,66	22,27	0,39	0,32	0,49	0,62	0,56	0,70	3,06	2,67	3,46	0,00	0	0	4,13	3	7	186,80	178	200	920550,50	918741	921007
KNN(k=15)	46,23	42,22	48,89	53,77	51,11	57,78	17,72	15,14	22,04	0,39	0,33	0,49	0,63	0,57	0,70	3,18	2,83	3,67	0,00	0	0	4,66	4	8	210,60	199	225	920550,50	918741	921007
KNN(k=30)	44,23	37,78	56,52	55,77	43,48	62,22	18,15	15,40	22,25	0,40	0,33	0,49	0,63	0,58	0,70	3,49	3,06	3,85	0,00	0	0	6,86	6	8	310,10	303	316	920550,50	918741	921007
SVML(c=1;e=0,1)	71,02	64,44	80,00	28,98	20,00	35,56	30,11	28,76	31,14	0,67	0,64	0,69	0,82	0,80	0,83	143,23	109,94	220,43	23977,90	20344	31013	0,04	0	1	23979,70	20346	31015	1078127,20	1076080	1078639
SVML(c=10;e=0,1)	64,84	55,56	77,78	35,16	22,22	44,44	31,19	28,68	32,41	0,69	0,64	0,72	0,83	0,80	0,85	125,55	65,85	194,54	313864,40	218388	505393	0,05	0	1	313866,60	218390	505396	1078127,20	1076080	1078639
SVML(c=50;e=0,1)	nieakceptowalny czas treningu																													
SVML(c=100;e=0,1)	nieakceptowalny czas treningu																													
SVML(c=20;e=0,1)	nieakceptowalny czas treningu																													
SVML(c=20;e=0,01)	63,52	50,00	75,56	36,48	24,44	50,00	31,55	29,76	32,94	0,70	0,66	0,72	0,84	0,81	0,85	111,77	76,70	154,41	633590,20	493459	878283	0,04	0	1	633592,20	493461	878285	1078127,20	1076080	1078639
SVML(c=20;e=0,001)	nieakceptowalny czas treningu																													
SVMR(c=10;s=0,5;e=0,1)	nieakceptowalny czas treningu																													
SVMR(c=10;s=1;e=0,1)	62,62	55,56	73,33	37,38	26,67	44,44	25,49	23,19	28,56	0,56	0,52	0,63	0,75	0,72	0,80	29,17	24,35	44,14	206479,90	172008	230106	10,00	8	14	206931,90	172470	230527	1049087,20	1047040	1049599
SVMR(c=10;s=5;e=0,1)	70,79	62,22	80,00	29,21	20,00	37,78	28,20	26,95	30,55	0,62	0,60	0,66	0,79	0,77	0,81	76,95	59,00	116,14	141432,70	101893	179725	5,52	4	12	141682,30	102125	179962	1049087,20	1047040	1049599
SVMR(c=10;s=10;e=0,1)	67,46	48,89	80,00	32,54	20,00	51,11	26,85	25,39	28,73	0,59	0,56	0,62	0,77	0,75	0,79	45,30	34,64	61,73	28762,30	19839	47045	5,95	4	12	29031,40	20077	47310	1049087,20	1047040	1049599
SVMP(c=10;d=3;e=0,01)	nieakceptowalny czas treningu																													
SVMP(c=10;d=4;e=0,01)	nieakceptowalny czas treningu																													
SVMP(c=10;d=6;e=0,01)	nieakceptowalny czas treningu																													
RBFN(k=5;f=2;c=1)	58,38	42,22	68,89	41,62	31,11	57,78	23,58	21,26	26,03	0,52	0,46	0,58	0,72	0,68	0,76	14,25	10,87	19,13	15,80	12	30	0,40	0	2	34,00	30	46	918792,10	916989	919247
RBFN(k=10;f=2;c=1)	60,61	44,44	68,89	39,39	31,11	55,56	24,36	21,76	26,62	0,54	0,48	0,59	0,73	0,70	0,77	19,60	14,79	23,92	28,10	26	30	0,80	0	2	64,20	59	68	919337,10	917534	919792
RBFN(k=15;f=2;c=1)	61,93	46,67	71,11	38,07	28,89	53,33	24,03	22,20	26,20	0,53	0,49	0,58	0,73	0,70	0,76	20,65	17,37	26,55	42,90	41	44	1,18	0	3	96,40	88	105	919882,10	918079	920337
RBFN(k=20;f=2;c=1)	64,37	51,11	71,11	35,63	28,89	48,89	23,94	21,97	25,77	0,53	0,49	0,57	0,73	0,70	0,76	19,95	17,31	24,26	59,70	57	65	1,59	1	4	131,60	123	139	920427,10	918624	920882
RBFN(k=5;f=1;c=2)	57,94	42,22	64,44	42,06	35,56	57,78	24,03	22,28	26,84	0,53	0,50	0,60	0,73	0,70	0,77	13,68	10,67	15,83	249,00	46	384	0,38	0	2	266,10	65	399	930018,10	928215	930473
RBFN(k=10;f=1;c=2)	60,39	44,44	68,89	39,61	31,11	55,56	24,23	21,82	26,06	0,54	0,48	0,58	0,73	0,70	0,76	18,04	14,62	23,07	298,40	106	589	0,75	0	2	332,10	141	620	941773,10	939970	942228
RBFN(k=15;f=1;c=2)	61,28	46,67	68,89	38,72	31,11	53,33	24,13	22,16	26,54	0,53	0,49	0,59	0,73	0,70	0,77	18,76	15,21	25,61	506,70	119	848	1,14	0	4	558,20	170	895	953528,10	951725	953983
RBFN(k=20;f=1;c=2)	62,15	46,67	68,89	37,85	31,11	53,33	24,28	22,43	25,98	0,54	0,50	0,58	0,73	0,71	0,76	19,70	16,17	27,08	930,90	541	1685	1,45	1	3	996,20	602	1750	965283,10	963480	965738
RBFN(k=5;f=2;c=2)	56,83	42,22	68,89	43,17	31,11	57,78	23,77	21,87	26,91	0,53	0,49	0,60	0,72	0,70	0,77	13,75	11,73	16,95	14,50	14	16	0,40	0	1	32,80	30	35	918814,10	917011	919269
RBFN(k=10;f=2;c=2)	61,48	46,67	69,57	38,52	30,43	53,33	24,06	22,15	26,06	0,53	0,49	0,58	0,73	0,70	0,76	17,39	14,39	20,30	28,10	26	30	0,80	0	2	64,10	60	68	919359,10	917556	919814
RBFN(k=15;f=2;c=2)	64,15	48,89	71,74	35,85	28,26	51,11	24,02	22,06	26,66	0,53	0,49	0,59	0,73	0,70	0,77	21,50	17,50	28,70	42,70	40	45	1,17	0	3	95,50	90	101	919904,10	918101	920359
RBFN(k=20;f=2;c=2)	62,37	46,67	77,78	37,63	22,22	53,33	24,10	21,91	26,33	0,53	0,49	0,59	0,73	0,70	0,76	21,10	17,94	26,59	58,30	56	59	1,59	1	5	130,10	120	138	920449,10	918646	920904
PNN(sigma=0,1)	51,78	46,67	62,22	48,22	37,78	53,33	21,31	16,41	24,01	0,47	0,36	0,53	0,69	0,60	0,73	2,00	1,61	2,17	0,00	0	0	9,32	7	35	421,10	398	444	923443,30	921637	923899
PNN(sigma=1)	47,58	40,00	53,33	52,42	46,67	60,00	32,51	31,78	33,72	0,72	0,71	0,74	0,85	0,84	0,86	165,73	40,69	420,59	0,00	0	0	9,32	7	15	421,40	409	437	923443,30	921637	923899
PNN(sigma=10)	49,76	37,78	60,87	50,24	39,13	62,22	38,43	38,25	39,11	0,85	0,85	0,85	0,92	0,92	0,92	4,29E+06	2,25E+06	7,31E+06	0,00	0	0	9,38	7	17	424,00	412	442	923443,30	921637	923899
PNN(sigma=100)	49,76	37,78	60,87	50,24	39,13	62,22	38,51	38,34	39,19	0,85	0,85	0,85	0,92	0,92	0,92	4,30E+10	2,24E+10	7,33E+10	0,00	0	0	9,33	8	12	421,90	407	439	923443,30	921637	923899

6.9. Skalowalność - porównanie wybranych wskaźników

W tym podrozdziale przenalizowane zostaną wartości wskaźników związanych ze skalowalnością algorytmów. Na wykresach 6.1 oraz 6.2 przedstawione są wielkości wskaźników E_TOT oraz E_MEM. Spośród testowanych algorytmów wybrano po jednym, konkretnie sparametryzowanym, który osiągnął najlepsze wartości Q_COR dla większości zbiorów danych. Wykaz algorytmów przedstawianych na poniższych wykresach znajduje się w tabeli 6.28. Omawiana analiza dotyczy wartości uzyskanych dla walidacji na całym zbiorze testowym.

Tablica 6.28: Algorytmy zastosowane podczas analizy skalowalności

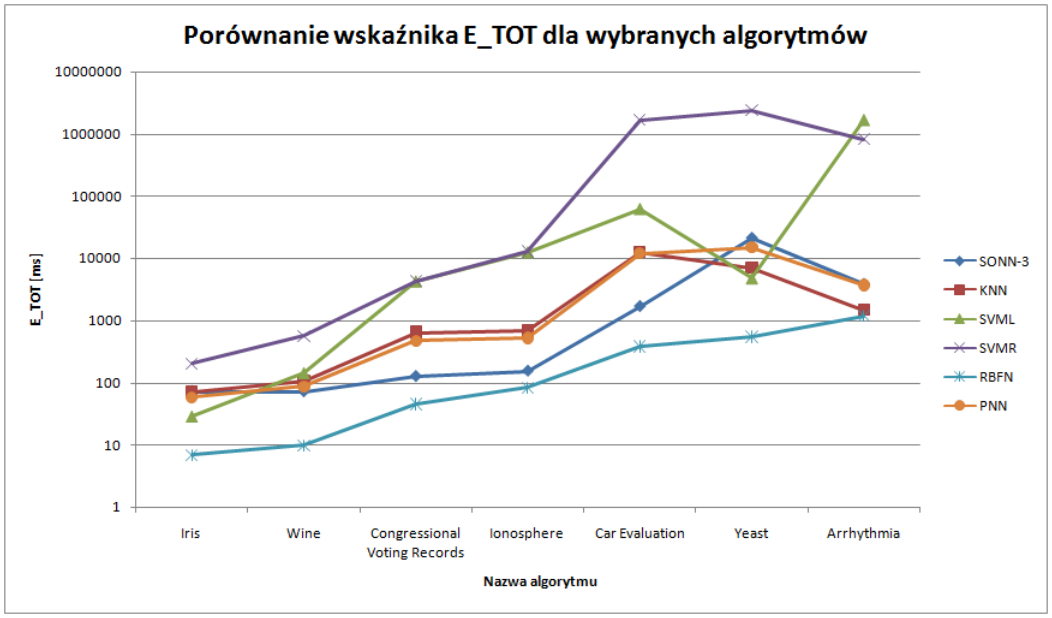
Oznaczenie	**Algorytm**	**Parametry**
SONN-3	Self Optimizing Neural Networks	
KNN	K nearest neighbours	$k = 3$
SVML	Support Vector Machines	klasyfikator liniowy, $cost = 20$, $eps = 0,01$
SVMR	Support Vector Machines	klasyfikator nieliniowy, $cost = 10$, $sigma = 0,5$, $eps = 0,1$
RBFN	Radial Basis Function Networks	funkcja radialna: wielokwadratowa, metoda wyboru centrów: K-mean clustering, $k = 20$

Na poziomych osiach obu wykresów widnieją algorytmy posortowane według indeksu, który obrazuje stopień trudności zbioru danych. Jest on obliczany jako:

$$\text{Stopień trudności} = \text{ilość cech} * \text{ilość wektorów treningowych} * \text{liczba klas}$$

Warto zauważyć, że tak obliczany indeks nie uwzględnia wszystkich czynników wpływających na rzeczywisty stopień trudności zbioru. Co więcej, stopień trudności mógłby być inaczej określany dla różnych algorytmów, ponieważ każdy ma inną specyfikę i wykorzystuje inne własności danych zbioru. Niemniej jednak, tak zdefiniowany indeks pozwolił na zadowalające oszacowanie stopnia trudności zbiorów danych dla analizowanych algorytmów i wskaźników.

Na osiach pionowych przedstawiono wielkości dla każdego ze wskaźników. Warto zauważyć, że wartości na osi pionowej układają się w skali logarytmicznej.



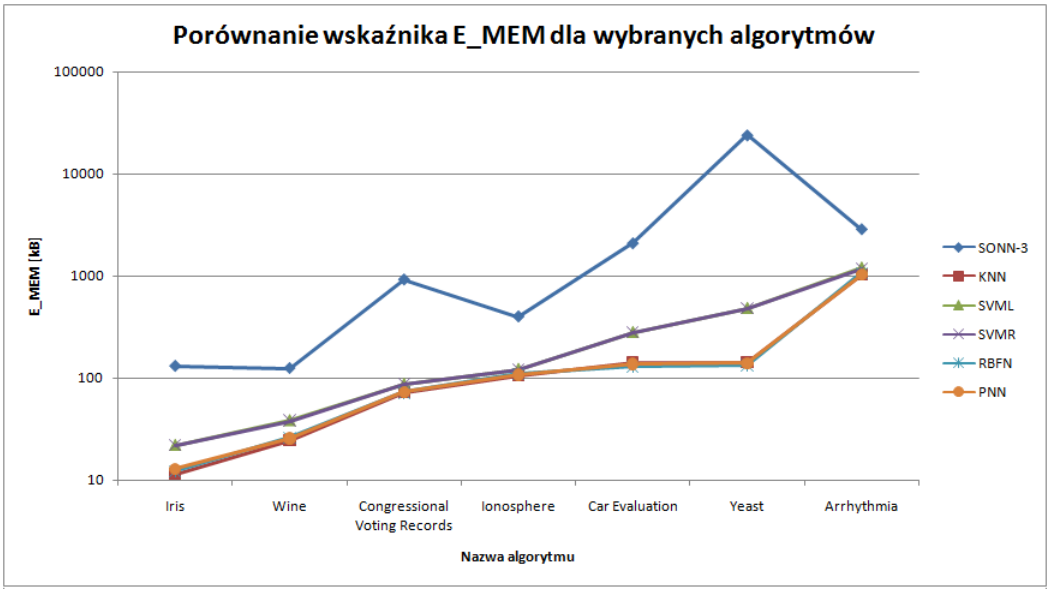
Rysunek 6.1: Porównanie wskaźnika E_TOT dla wybranych algorytmów

Na wykresie 6.1 przedstawiono wartości całkowitego czasu operacji E_TOT. Istotne zależności, które mogą zostać zaobserwowane na przedstawionym wykresie, to:

- Metodą, która w najkrótszym czasie potrafi utworzyć klasyfikator okazała się metoda sieci radialnych funkcji bazowych (RBFN).
- Sieci SONN potrafią w stosunkowo krótkim czasie utworzyć klasyfikator w porównaniu z innymi metodami.
- Popularny algorytm SVM (w dwóch odmianach SVML i SVMR) wymagał więcej czasu w stosunku do pozostałych metod dla każdego ze zbiorów.
- Zaproponowany sposób obliczania współczynnika stopnia trudności wydaje się uwzględniać wystarczającą ilość informacji o zbiorze danych, dzięki czemu poprawnie udało się uporządkować zbiory według stopnia trudności.

Kolejnym istotnym z punktu widzenia skalowalności wskaźnikiem jest ilość pamięci potrzebnej do utworzenia obiektu klasyfikatora. Wartości wskaźnika E_MEM znajdują się na wykresie 6.2. Analizując ten wykres można zauważyć, że:

- Algorytm SONN okazał się najbardziej wymagającym z punktu widzenia wymaganej pamięci.
- Zbiór Yeast jest szczególnie trudny dla sieci SONN (wskazują na to wysokie wartości wskaźników E_MEM i E_TOT).



Rysunek 6.2: Porównanie wskaźnika E_MEM dla wybranych algorytmów

- Wymagana pamięć przez algorytmy SVM zależy od ilości klas występujących dla danego problemu klasyfikacji.

7. Podsumowanie

W ramach pracy udało się przeprowadzić porównania wybranych metod klasyfikacji z sieciami SONN. Porównania 37 algorytmów dla 7 zbiorów danych, przy użyciu najpopularniejszych metod walidacji stanowią cenny zbiór informacji o sieciach SONN oraz analizowanych algorytmach. Dzięki tym informacjom udało się zauważyć kilka interesujących wniosków, które wymienione zostaną poniżej.

Oprócz samych porównań dużą wartością pracy jest aplikacja wspomagająca porównywanie metod klasyfikacji, dzięki której możliwa jest wygodna analiza algorytmów klasyfikacji dla dowolnych zbiorów danych, najbardziej popularnych metod walidacji i wstępnego przetwarzania danych. Obok aplikacji warto wymienić także bibliotekę algorytmów, która może z łatwością zostać zastosowana w dowolnym projekcie, co umożliwia stosowanie zaimplementowanych algorytmów w praktycznych zagadnieniach.

7.1. Wnioski

Dzięki uzyskanemu zestawowi porównań udało się wyciągnąć następujące wnioski:

- Sieci SONN są obiecującym i ciekawym narzędziem klasyfikacji. Dla większości zbiorów danych algorytm SONN w stosunkowo krótkim czasie potrafił skonstruować strukturę sieci, która na ogół potrafiła realizować zadanie klasyfikacji z porównywalnym lub niewiele gorszym wynikiem niż popularne metody klasyfikacji.

- Ciekawą własnością sieci SONN jest osiąganie stuprocentowej skuteczności dla walidacji na całym zbiorze danych treningowych, który nie posiada wzorców sprzecznych. Jest to unikatowa własność wśród badanych metod klasyfikacji (można było ją zaobserwować również dla algorytmu kNN, dla $k = 1$, który jednak stanowi raczej ciekawy, szczególny przypadek tej metody, niż stosowane narzędzie klasyfikacji).

- W porównaniu objętości pamięciowej, jaka potrzebna jest do utworzenia obiektu klasyfikatora sieci SONN wykazywały najgorsze właściwości spośród badanych algorytmów. Wynika to z pewnych założeń optymalizacyjnych algorytmu SONN, dzięki którym potrafią one w krótkim czasie utworzyć klasyfikator (zmniejszenie złożoności obliczeniowej

kosztem zwiększenia zużycia pamięci). Mimo możliwości aktualnie produkowanych komputerów i nieustannej poprawie wydajności podzespołów wymagania pamięciowe sieci SONN są na tyle wysokie, że powinny stać się przedmiotem dokładnej analizy i poprawy, jeżeli algorytm ten ma być stosowany do większych niż analizowane w pracy zbiory danych.

- Sieci SONN jako jedyne spośród analizowanych algorytmów wykazują odporność na znaczne różnice w wielkościach cech (np. zbiór Wine). Pozwala to na stosowanie tego algorytmu bez konieczności wstępnego przetwarzania danych (np. normalizacja).

- Realizując porównania metod dla różnych zbiorów danych, zauważono, że niektóre metody klasyfikacji przy nieumiejętnym dobraniu parametrów nie potrafią utworzyć klasyfikatora z powodu błędu (RBFN) lub zbyt długiego czasu treningu (SVM). Brak parametrów dla algorytmu SONN jest jedną z największych zalet tego algorytmu.

- Podczas porównań niektórych metod klasyfikacji (np. Knn, PNN) warto analizować oprócz czasu treningu także czas klasyfikacji, gdyż dla tych algorytmów główne obliczenia wykonywane są na etapie klasyfikacji.

- Popularny algorytm SVM, który jest powszechnie stosowany w praktycznych zastosowaniach osiąga najlepsze rezultaty w zakresie czasu treningu dla zbiorów danych o małej liczbie klas. Jest to związane z koniecznością tworzenia wielu klasyfikatorów binarnych dla problemów wieloklasowych, co rzadko opisywane jest w publikacjach i pracach badawczych.

- Najtrudniejszym zbiorem danych dla sieci SONN (z punktu widzenia wskaźników E_MEM i E_TOT oraz Q_COR) okazał sie zbiór Yeast. Cechą, która wyróżnia ten zbiór spośród innych, dla których przeprowadzono porównania, jest bardzo nierównomierny rozkład wzorców dla klas problemu. Podobną własność można zaobserwować dla zbioru Arrhythmia, dla którego sieci SONN również nie osiągnęły zadowalającej skuteczności.

- Dla zbioru Car Evaluation, który składa się z wartości symbolicznych (zamienionych na kolejne liczby naturalne) metoda SONN uzyskała stosunkowo dobre wskaźniki jakości uogólnienia. Powinno to zostać wzięte pod uwagę podczas tworzenia wersji SONN-4, która według planów autora metody, ma umożliwiać przeprowadzanie klasyfikacji dla zbiorów zawierających wartości symboliczne.

7.2. Możliwości rozwoju pracy

Celem pracy było porównanie sieci SONN z popularnymi metodami klasyfikacji. Dzięki przeprowadzeniu odpowiednich porównań udało się zaobserwować następujące aspekty działania algorytmu, które powinny stać się przedmiotem dokładniejszej analizy i próby poprawy:

- Duże wymagania pamięciowe metody - jest to najbardziej wyraźna wada sieci SONN w porównaniu z innymi metodami klasyfikacji. Ponad dziesięciokrotnie większe wymagania pamięciowe w stosunku do innych metod mogą uniemożliwić stosowanie metody SONN dla większych niż analizowane zbiorów danych.
- Niska skuteczność dla zbiorów Yeast oraz Arrhythmia - w pierwszej kolejności warto zbadać wpływ rozkładu wzorców według klas na zdolności generalizacyjne sieci SONN.
- Skuteczność algorytmu na ogół jest nieco gorsza od innych metod klasyfikacji - możliwe, że algorytm SONN starając się utworzyć strukturę, która ze stuprocentową skutecznością rozpoznaje wzorce treningowe traci zdolność do uogólnienia (przeuczenie).

Zrealizowana w ramach pracy aplikacja wspomagająca porównania okazała się bardzo pomocnym i funkcjonalnym narzędziem. W ramach jej rozwoju na pewno warto uwzględnić:

- Funkcjonalność wizualizacji utworzonych klasyfikatorów - aktualnie podawane są tylko informacje o strukturze klasyfikatora w formie tabelarycznej.
- Zwiększenie interaktywności z użytkownikiem - rozbudowa systemu pomocy oraz dodanie kilku udogodnień takich, jak szacowanie czasu przebiegu symulacji, czy pokazywanie pozostałego czasu symulacji.

Naturalnie, można również wzbogacić pracę o kolejne porównania uwzględniające nowe zbiory danych, inne lub inaczej sparametryzowane algorytmy, metody wstępnego przetwarzania danych, czy metody walidacji. Dzięki funkcjonalnej i ogólnodostępnej aplikacji, która została zaimplementowana i wykorzystana w ramach pracy możliwe jest dokonywanie nowych porównań metod klasyfikacji, dzięki którym - zgodnie z intencją autora - możliwy będzie rozwój metod klasyfikacji, a w szczególności algorytmu *Self Optimizing Neural Networks*.

A. Zawartość dołączonej płyty CD-ROM

Na dołączonej płycie CD-ROM znajdują się następujące pliki oraz foldery:

- plik LukaszDziedzia2009.pdf - dokument pracy magisterskiej zapisany w formacie .pdf.
- folder LukaszDziedzia2009 - źródła, które pozwalają na edycję oraz utworzenie dokumentu:
  - latex - pliki L̃atex, które stanowią treść niniejszego dokumentu.
  - symulacje - wyniki symulacji opisanych w pracy.
- Projekt - kod źródłowy zrealizowany jako część praktyczna pracy wraz z wszystkimi referencjami niezbędnymi do uruchomienia.

Spis rysunków

Spis tablic

Bibliografia

[1] praca zbiorowa, "How much information? 2003," 2003. http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/printable_report.pdf.

[2] "Przeglądarka internetowa clusty." http://Clusty.org/about.

[3] Łukasz Dziedzia, "Symulacja procesu uczenia sztucznej sieci neuronowej," 2006. (projekt zrealizowany w ramach przedmiotu „Programowanie systemów czasu rzeczywistego").

[4] D. Bridge, "Topics in Artificial Intelligence." http://www.cs.ucc.ie/~dgb/courses/tai/notes/handout4.pdf.

[5] C. J. Bugres, *A Tutorial on Support Vector Machines for Pattern Recognition*, p. 121–167. Data Mining and Knowledge Discovery, Kluwer Academic Publishers, 1998.

[6] "http://www.support-vector.net/icml-tutorial.pdf."

[7] A. W. Moore, "Support vector machines," 2001. http://www.autonlab.org/tutorials/svm15.pdf.

[8] J. S.-T. Nello Cristianini, "An introduction to support vector machines," 2000.

[9] J. C. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," 1998. http://home.mit.bme.hu/~gtakacs/download/platt1998.pdf.

[10] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," 2000.

[11] "http://omega.albany.edu:8008/machine-learning-dir/notes-dir/ker1/ker1-l.html."

[12] K.-B. Duan and S. S. Keerthi, "Which is the best multiclass SVM method? an empirical study," 2005.

[13] U. Halici, "Artificial neural networks, chapter IX."

[14] P. Schrater, "Radial basis function networks." http://gandalf.psych.umn.edu/users/schrater/schrater_lab/courses/PattRecog07/RegressionIII.pdf.

[15] M. Krętowska, "Sztuczne sieci neuronowe, ocena jakości sieci neuronowej, sieci RBF."

[16] S. Osowski, *Sieci Neuronowe w ujęciu algorytmicznym*. Warszawa: PWNT, 1996.

[17] N. Y. Nikolaev, "Probabilistic neural networks."

[18] F. Gorunescu, "Benchmarking probabilistic neural network algorithms." http://inf.ucv.ro/~aidc/proceedings/2006/1%20fgorunescu.pdf.

[19] D. F. Specht, "Probabilistic neural networks for classification, mapping or associative memory." http://www.inf.ufrgs.br/~engel/Common/CMP121/PNN.pdf.

[20] "Internetowy podręcznik statystyki." http://www.sixsigma.pl/textbook/stathome_stat.html?http%3A%2F%2Fwww.sixsigma.pl%2Ftextbook%2Fstneunet.html.

[21] A. Horzyk, "Self-optimizing neural network 3."

[22] A. Horzyk, "Self optimizing neural networks sonn-3 for classification tasks."

[23] A. Horzyk, "Self-optimizing neural networks sonn-2."

[24] A. Horzyk and R. Tadeusiewicz, "Comparison of plasticity of self-optimizing neural networks and natural neural networks."

[25] A. Horzyk, "Automatic discriminative lossy binary conversion of redundant real training data inputs for simplifying an input data space and data representation."

[26] A. Horzyk, "Nowe metody uczenia sieci neuronowych bez sprzężeń zwrotnych," 1999. (praca doktorska napisana pod kierunkiem prof. zw. dr hab. inż. Ryszarda Tadeusiewicza).

[27] J. Platt, "A resource-allocating network for function interpolation." Neural Computation, vol. 3, no. 2, pp. 213-225, (Summer 1991).

[28] N. Jankowski, "Ontogeniczne sieci neuronowe w zastosowaniu do klasyfikacji danych medycznych," 1999. (praca doktorska pod kierunkiem prof. Włodzisława Ducha).

[29] "Weka." http://www.cs.waikato.ac.nz/ml/weka/.

[30] "Rapid Miner." http://rapid-i.com/content/blogcategory/38/69/.

[31] "UC Irvine Machine Learning Repository." http://archive.ics.uci.edu/ml/index.html.

[32] L. I. Smith, "A tutorial on principal components analysis."

[33] M. M. Robert C. Martin, "Agile, Programowanie zwinne," 2008.

[34] J. Kerievsky, "Refaktoryzacja do wzorców projektowych," 2005.

[35] "ADO.NET." http://msdn.microsoft.com/en-us/library/e80y5yhx%28VS.80%29.aspx.

[36] "Microsoft Developer Network." http://msdn.microsoft.com/.

[37] A. Kusiak, K. Kernstine, J. Kern, K. McLaughlin, and T. Tseng, "Data mining: Medical and engineering case studies," 2000. http://www.icaen.uiowa.edu/~ankusiak/Res-in-Prog/IIE-0.pdf.

[38] T. Fawcett, "An introduction to ROC analysis," 2005.