Politechnika Wrocławska
Wydział Informatyki i Zarządzania
Instytut Informatyki

Rozprawa doktorska

Zespoły klasyfikatorów SVM dla danych niezbalansowanych

Maciej Zięba

Promotor: prof. dr hab. inż. Jerzy Świątek

Wrocław 2013

Podziękowania

Na wstępie chciałby podziękować swojemu promotorowi, prof. dr hab. inż. Jerzemu Świątkowi, za wsparcie merytoryczne w realizacji rozprawy doktorskiej, oraz opiekę naukową począwszy od wczesnych lat studiów. Swoje podziękowania kieruję również w stronę prof. dr hab. inż. Adama Grzecha, który umożliwił mi rozwój naukowy i służył wsparciem w zagadnieniach związanych z tematyką SOA.

Chciałbym również podziękować swojemu przyjacielowi, dr inż. Jakubowi Tomczakowi, za cenne uwagi dotyczące rozprawy i tysiące godzin spędzonych na dyskusjach naukowych. Ponadto chciałbym podziękować Adamowi Gonczarkowi za wartościowe uwagi dotyczące rozprawy, dr inż. Markowi Lubiczowi za pomoc w realizacji prac dotyczących analizy ryzyka operacyjnego, oraz dr inż. Agnieszce Prusiewicz, dr inż. Pawłowi Świątkowi i Pawłowi Stelmachowi za cenne uwagi dotyczące realizacji zagadnień związanych z paradygmatem SOA.

Szczególne podziękowania kieruję bliskiej mojemu sercu osobie, Katarzynie Pali, za wsparcie w realizacji rozprawy i liczne dyskusje interdyscyplinarne.

Pracę tą dedykuję swoim Rodzicom i Siostrze, bez wsparcia których niemożliwe byłoby poświęcenie się pracy naukowej i obranie obecnej drogi życiowej.

Część niniejszej pracy jest współfinansowana ze środków Unii Europejskiej poprzez Europejski Fundusz Rozwoju Regionalnego w ramach Programu Operacyjnego Innowacyjna Gospodarka na lata 2007-2013, numer projektu: POIG.01.03.01-00-008/08.

Część niniejszej pracy jest wykonana w ramach Grantu Plus współfinansowanego przez Unię Europejską w ramach Europejskiego Funduszu Społecznego.

Spis treści

Rozdział 1

Wstęp

1.1 Wprowadzenie

Postępująca cyfryzacja i informatyzacja spowodowała rozrost agregowanych w systemach informatycznych wolumenów danych. Rozrost danych idący w parze z ciągłym rozwojem technik uczenia maszynowego umożliwił automatyzację procesów decyzyjnych w systemach diagnostycznych, finansowych, bezpieczeństwa, oraz wielu innych, wykorzystujących rozwiązania z obszaru sztucznej inteligencji. Konieczność konstrukcji modeli decyzyjnych w sposób automatyczny zapoczątkowała rozwój gałęzi uczenia maszynowego poświęconej technikom niwelowania złej jakości surowych danych wykorzystywanych w procesie uczenia.

Problem niezbalansowania danych jest jednym z typowych problemów związanych ze złą jakością danych w zagadnieniach podejmowania decyzji formułowanych jako zadania klasyfikacji. Istotą problemu jest fakt, iż w zestawie danych wykorzystywanych w procesie uczenia obserwuje się przewagę liczności obiektów z jednej, bądź kilku klas. Zastosowanie typowych metody uczenia dla danych niezbalansowanych skutkuje obciążeniem konstruowanego modelu w kierunku klasy dominującej zbiór uczący. W konsekwencji konstruowany model decyzyjny ma tendencję do faworyzowania klasy przeważającej, co przekłada się na jego niską jakość predykcji.

Motywacją do podjęcia badań związanych z tematyką niezbalansowanych danych była niewystarczająca jakość, wysoka niestabilność, oraz brak uzasadnienia teoretycznego

dla opisanych w literaturze metod podejmujących problem. Większość z przedstawionych w piśmiennictwie rozwiązań wykorzystuje mechanizmy generowania syntetycznych obserwacji, techniki losowej eliminacji obiektów nadmiarowych, bądź też procedury związane z nadawaniem różnych kosztów błędnej klasyfikacji obiektom należącym do różnych klas. Metody te w większości przypadków obarczone są dużą losowością i koniecznością czasochłonnego wyznaczania optymalnych wartości dodatkowych parametrów uczenia. Tylko nieliczne z nich stanowią kompleksowe, teoretycznie uzasadnione rozwiązania niewymagające kalibracji wielu wartości parametrów mających kluczowe znaczenie dla jakości predykcji.

W rozprawie proponuje się zastosowanie zespołów klasyfikatorów SVM z metodą uczenia rozwiązującą problem danych niezbalansowanych w zbiorze uczącym. Algorytm konstrukcji kolejnych klasyfikatorów bazowych posiada silne uzasadnienie teoretyczne, gdyż sekwencyjnie minimalizuje ważoną, wykładniczą funkcję błędu uwzględniającą różnice w licznościach klas. Ponadto, konstrukcja każdego z klasyfikatorów bazowych SVM odbywa się poprzez minimalizację zmodyfikowanego kryterium uczenia, które eliminuje problem niezbalansowania nie tylko pomiędzy klasami, ale również w ramach każdej z klas. Dzięki połączeniu wielu technik uczenia maszynowego proponowane w rozprawie zespoły klasyfikatorów SVM charakteryzują się wysoką jakością klasyfikacji dla problemów o różnym stopniu niezbalansowania danych.

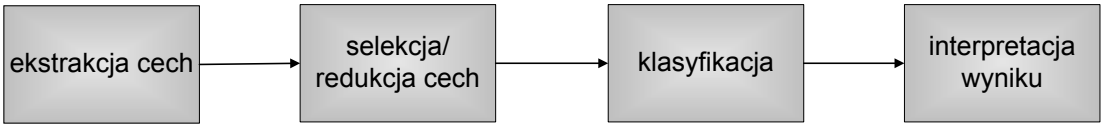
W niniejszym rozdziale opisano kluczowe dla zrozumienia tematyki pracy zagadnienia związane z rozpoznawaniem, klasyfikacją i uczeniem, a także sformułowano problem i przedstawiono tezę rozprawy.

1.2 Opis problemu

1.2.1 Rozpoznawanie obiektów

Rozpoznawanie obiektów (w polskim tłumaczeniu znane również jako rozpoznawanie obrazów, bądź też wzorców - *ang. pattern recognition*) jest jedną z podstawowych dyscyplin uczenia maszynowego. W pracy [121] definiuje się rozpoznawanie obiektów jako przydzielanie rozmaitego typu obiektów (lub zjawisk) do pewnych klas. Autor pracy [78] pisze o umiejętności rozpoznawania obiektów (zjawisk, procesów, sygnałów, sytuacji) jako

o zdolności do przypisania im konkretnego znaczenia (klasy) na podstawie pewnych charakterystycznych własności (cech). W ujęciu procesowym proponowanym w pozycji [32] rozpoznawanie obrazów zdefiniować można jako ciąg następujących po sobie operacji: ekstrakcji cech, selekcji (bądź też redukcji) cech, klasyfikacji, interpretacji wyniku klasyfikacji (Rysunek 1.1).



Rysunek 1.1: Podejście procesowe do rozpoznawania obiektów.

Operacja ekstrakcji cech polega na przekształceniu danych opisujących dany obiekt (zjawisko, sygnał, sytuację) do wektora cech charakteryzujących dany obiekt w zadanym kontekście rozpoznawania. Przykładowo, jeżeli rozważymy jedno z typowych zagadnień rozpoznawania obiektów jakim jest rozpoznawanie ręcznie napisanych znaków (liter, znaków specjalnych) [149], to operacja ekstrakcji cech będzie polegać na wyznaczeniu na podstawie trajektorii pisania takich cech charakterystycznych, które pozwolą na odróżnienie go od innych znaków. Przykładową cechą może być długość trajektorii pisania znaku, bądź też średni kąt pomiędzy kolejnymi odcinkami budującymi trajektorię. Innym przykładem ekstrakcji cech jest wydobywanie własności charakterystycznych dla sygnału EMG pozwalających określić stopień aktywności danego mięśnia [17].

Zasadniczym celem selekcji, bądź też redukcji cech, jest zwiększenie efektywności procesu rozpoznawania obiektu poprzez zmniejszenie wymiaru wektora cech. Zmniejszenie liczby cech w wektorze może odbyć się odrzucenie tych cech, które mają niewielki wpływ na wynik rozpoznawania (taką operację nazywa się selekcją cech), bądź też na utworzeniu nowego wektora cech o niższym niż wejściowy wektor wymiarze, składającego się z kombinacji wartości cech wektora wejściowego (wówczas mówmy o redukcji cech).

Kluczowym elementem procesu rozpoznawania obiektów jest operacja klasyfikacji. Operacja ta polega na przekształceniu wektora cech opisującego dany obiekt do wartości reprezentującej jedną z możliwych klas obiektu. Funkcję przekształcającą wektor cech do wartości charakteryzującej klasę nazywamy klasyfikatorem (*ang. classifier*), bądź też metodą klasyfikacji (*ang. classification method*). Klasyfikator może być podany przez eksperta, np.

w postaci zestawu reguł, bądź też jako drzewo decyzyjne, jednak w większości przypadków jest on konstruowany w procesie uczenia nadzorowanego z wykorzystaniem zbioru uczącego.

Ostatnim komponentem procesu rozpoznawania obiektów jest interpretacja wyniku klasyfikacji. Możliwe klasy dla wybranego problemu rozpoznawania mogą być zakodowane w postaci wektorów binarnych, zbioru liczb naturalnych, bądź też wartości nominalnych będącymi etykietami klas, dlatego konieczne jest przekształcenie ich na język naturalny w końcowym etapie procesu rozpoznawania.

Jakość procesu rozpoznawania obiektów uwarunkowana jest przede wszystkim doborem skutecznego klasyfikatora, dlatego niniejsza praca koncentruje się na zagadnieniach związanych z klasyfikacją.

1.2.2 Klasyfikacja

W poprzedniej sekcji przedstawiono klasyfikację jako operację nadawania klas obiektom. Klasy te mogą reprezentować np. poszczególne litery alfabetu, status kredytowy klienta systemu bankowego, bądź też rodzaj diagnozy medycznej. Formalnie, każdy obiekt opisać można D-wymiarowym wektorem cech (nazywany również wektorem atrybutów):

$$\mathbf{x} = [x_1 \dots x_d \dots x_D]^T, \tag{1.1}$$

gdzie element x_d wektora $\mathbf{x}$ reprezentuje d-tą cechę rozpatrywanego obiektu. Cechy opisujące obiekt mogą przyjmować wartości liczbowe (rzeczywiste, naturalne), bądź też nominalne, pochodzące ze zbioru wartości bez zdefiniowanej relacji następstwa. Przestrzeń możliwych wartości wektora cech oznaczać będziemy przez $\mathbb{X}$. Dla zadanego problemu klasyfikacji zdefiniować można zbiór możliwych etykiet klas:

$$\mathbb{Y} = \{C_1, \dots, C_y, \dots, C_Y\}, \tag{1.2}$$

gdzie Y oznacza liczbę możliwych etykiet klas. Najczęściej przyjmuje się, że etykietami klas są kolejne liczby naturalne, co oznacza, że $\mathbb{Y} = \{0, \dots, y, \dots, Y-1\}$.

Klasyfikator (metoda klasyfikacji) Ψ przypisuje każdemu wektorowi zmierzonych cech $\mathbf{x}$ z przestrzeni $\mathbb{X}$ etykietę klasy ze zbioru $\mathbb{Y}$. Innymi słowy, klasyfikator Ψ odwzorowuje przestrzeń cech w zbiór etykiet klas:

$$\Psi : \mathbb{X} \rightarrow \mathbb{Y}. \tag{1.3}$$

Równoważnie, klasyfikator Ψ generuje rozkład przestrzeni cech na tzw. obszary decyzyjne (*ang. decision regions*):

$$\mathbb{D}_{\mathbf{x}}^{(y)} = \{\mathbf{x} \in \mathbb{X} : \Psi(\mathbf{x}) = y\}. \tag{1.4}$$

Naturalnie, obszary decyzyjne wyznaczane przez klasyfikator Ψ dla każdej z klas ($y \in \mathbb{Y}$) są obszarami rozłącznymi i w sumie tworzą przestrzeń $\mathbb{X}$. Powierzchnie, które separują obszary decyzyjne, nazywamy powierzchniami decyzyjnymi (*ang. decision surfaces*).

1.2.3 Problem uczenia klasyfikatora

Jeżeli dana jest postać klasyfikatora, operacja klasyfikacji sprowadza się do wyznaczenia wartości funkcji Ψ dla zadanego wektora cech. Dla większości zadań klasyfikacyjnych postać klasyfikatora jest trudna do określenia bezpośrednio przez eksperta, dlatego konieczna jest jest jego konstrukcja w procesie uczenia nadzorowanego. Proces uczenia (nazywany również treningiem) wykorzystuje tzw. zbiór uczący (zbiór treningowy) zawierający wektory wartości cech i korespondujące etykiety klas obiektów, w przypadku których znany jest wynik klasyfikacji. Przykładowo, dla zadania rozpoznawania odręcznie napisanych liter zbiór uczący zawiera wektory wartości cech wyznaczone z trajektorii powstałych podczas pisania przez różnych pisarzy, wraz z etykietami klas reprezentującymi odpowiednie litery. Zbiór uczący oznacza się w następujący sposób:

$$\mathbb{S}_N = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n), \ldots, (\mathbf{x}_N, y_N)\}, \tag{1.5}$$

gdzie N oznacza liczbę elementów zbioru uczącego, $\mathbf{x}_n$ reprezentuje wektor cech charakteryzujący n-ty obiekt w zbiorze uczącym ($\mathbf{x}_n \in \mathbb{X}$), natomiast y_n oznacza etykietę klasy, do której należy n-ty obiekt.

W pozycji [10] wyróżnia się dwa podejścia do zagadnienia klasyfikacji:

- podejście polegające na bezpośredniej konstrukcji funkcji dyskryminujących,
- podejścia polegające na modelowaniu warunkowego prawdopodobieństwa *a posteriori*.

Pierwsze z podejść zakłada, że klasyfikator Ψ opisany jest zbiorem funkcji klasyfikujących nazywanymi funkcjami dyskryminującymi (nazywanych również funkcjami separującymi):

$$f_y : \mathbb{X} \to \mathbb{R}, \tag{1.6}$$

gdzie $y \in \mathbb{Y}$. W podejściu z konstrukcją funkcji dyskryminujących klasyfikacja odbywa się wedle następującej reguły:

$$\Psi(\mathbf{x}) = \max_{y \in \mathbb{Y}} f_y(\mathbf{x}). \tag{1.7}$$

Szczególnym przypadkiem klasyfikacji jest tzw. dychotomia, czyli klasyfikacja, w której rozpatruje się dwie klasy ($Y = 2$). Dla tego przypadku wystarczy określić jedną funkcję dyskryminującą postaci:

$$f(\mathbf{x}) = f_1(\mathbf{x}) - f_2(\mathbf{x}). \tag{1.8}$$

Obiekt opisany wektorem x klasyfikowany jest do pierwszej z klas gdy $f(\mathbf{x})$ przyjmuje wartości dodatnie, natomiast do drugiej klasy, gdy wartość $f(\mathbf{x})$ jest ujemna. Równanie postaci $f(\mathbf{x}) = 0$ jest równaniem powierzchni rozdzielającej (nazywanej również powierzchnią separującą) dwie klasy.

Najprostszą reprezentacją funkcji dyskryminującej jest funkcja liniowa następującej postaci:

$$f(\mathbf{x}) = \mathbf{a}^{\mathrm{T}}\mathbf{x} + b, \tag{1.9}$$

gdzie $\mathbf{a}$ jest wektorem wag, natomiast b jest wyrazem wolnym funkcji dyskryminującej. Proces uczenia klasyfikatora reprezentowanego przez dyskryminującą funkcję liniową sprowadza się do wyznaczenia wartości parametrów $\mathbf{a}$ oraz b.

Drugie z wymienionych podejść polega na modelowaniu warunkowego rozkładu *a posteriori*. W podejściu tym zakłada się, że wektor wartości cech opisujących rozpoznawany obiekt oraz etykieta klasy stanowią realizację pary zmiennych losowych $(\mathbf{X}, \mathbf{Y})$. Zmienna losowa $\mathbf{Y}$ przyjmuje wartości ze zbioru $\mathbb{Y}$, natomiast zmienna losowa $\mathbf{X}$ przyjmuje wartości z przestrzeni możliwych wartości wektora cech $\mathbb{X}$. Łączny rozkład pary zmiennych lo-

sowych $(\mathbf{X}, \mathbf{Y})$ reprezentowany jest przez prawdopodobieństwo łączne $p(\mathbf{X}, \mathbf{Y})$. Dla prawdopodobieństwa łącznego zachodzi następująca własność:

$$p(\mathbf{X}, \mathbf{Y}) = p(\mathbf{Y}|\mathbf{X})\ p(\mathbf{X}). \tag{1.10}$$

Własność tą nazywa się regułą iloczynu (*ang. product rule*). Korzystając z tej własności wyznaczyć można następującą zależność pomiędzy warunkowymi prawdopodobieństwami:

$$p(\mathbf{Y}|\mathbf{X}) = \frac{p(\mathbf{X}|\mathbf{Y})\ p(\mathbf{Y})}{p(\mathbf{X})}. \tag{1.11}$$

Powyższa własność nazywana jest regułą Bayesa (*ang. Bayes' theorem*) i stanowi podstawę probabilistycznego podejścia do rozpoznawania. Klasyfikacja w statystycznym podejściu odbywa się według następującej reguły:

$$\Psi(\mathbf{x}) = \max_{y \in \mathbb{Y}} p(\mathbf{Y} = y|\mathbf{X} = \mathbf{x}) = \max_{y \in \mathbb{Y}} p(y|\mathbf{x}). \tag{1.12}$$

Obiekt opisany wektorem cech $\mathbf{x}$ klasyfikowany jest do klasy, dla której wartość prawdopodobieństwa $p(y|\mathbf{x})$, nazywanego prawdopodobieństwem *a posteriori* jest najwyższa. Wartość prawdopodobieństwa $p(\mathbf{X})$ nie zależy od etykiety klasy, dlatego, zgodnie z regułą Bayesa (1.11) klasyfikator opisany równaniem (1.12) jest równoważny klasyfikatorowi następującej postaci:

$$\Psi(\mathbf{x}) = \max_{y \in \mathbb{Y}} p(\mathbf{X} = \mathbf{x}|\mathbf{Y} = y)\ p(\mathbf{Y} = y) = \max_{y \in \mathbb{Y}} p(\mathbf{x}|y)p(y). \tag{1.13}$$

Uczenie klasyfikatora w podejściu statystycznym sprowadza się do wyznaczenia odpowiednich rozkładów prawdopodobieństw warunkowych $p(y|\mathbf{x})$. Jeżeli prawdopodobieństwa warunkowe modelowane są bezpośrednio, na przykład jako model parametryczny, którego parametry wyznaczane są w procesie optymalizacji z wykorzystaniem zbioru uczącego, to takie podejście nazywa się podejściem dyskryminacyjnym (*ang. disriminative apprach*). Alternatywne podejście nazywane podejściem generującym (*ang. generative apprach*) polega na modelowaniu prawdopodobieństw $p(\mathbf{x}|y)$, oraz $p(y)$, co umożliwi późniejszą klasyfikację z wykorzystaniem reguły klasyfikacyjnej (1.13).

Operacja uczenia klasyfikatora jest szczególnym przypadkiem ekstrakcji wiedzy (*ang. knowledge extraction*) na potrzeby podejmowania decyzji dotyczących przydziału obiektu

do klasy. Przykładowo, w podejściu do klasyfikacji z wykorzystaniem funkcji dyskryminujących wiedza wydobyta w procesie uczenia zawarta jest w parametrach tych funkcji. Proces ekstrakcji wiedzy jest uzależniony od sposobu reprezentowania wiedzy w klasyfikatorze. Wiedza może być reprezentowania w postaci funkcyjnej (np. w postaci funkcji dyskryminujących, bądź też w postaci funkcji gęstości reprezentujących rozkłady prawdopodobieństw), ale również w postaci relacyjnej, logicznej [18], w postaci reguł, drzew [141], bądź sieci (grafów) decyzyjnych [121].

Kluczowym elementem w procesie uczenia klasyfikatorów są dane zawarte w zbiorze uczącym. W odróżnieniu od zbiorów danych dostępnych w repozytoriach uczenia maszynowego jakość danych rzeczywistych w większości przypadków nie pozwala na bezpośrednie wykorzystanie ich w procesie konstrukcji modeli klasyfikacyjnych. Zła jakość danych może być spowodowana brakującymi wartościami pewnych atrybutów (*ang. missing values problem*) [50], niezbalasowaniem danych (*ang. imbalanced data problem*) [56], czy też sekwencyjnym sposobem dostarczania danych (*ang. sequential data problem*) [33]. Niniejsza dysertacja poświęcona jest w większości problemowi niezbalansowania.

1.3 Sformułowanie problemu pracy

1.3.1 Problem niezbalanowania danych

W literaturze poświęconej zagadnieniom związanym z dysproporcjami pomiędzy klasami brak jednoznacznej definicji niebalansowania danych w obszarze uczenia maszynowego. Autorzy pozycji [56] stwierdzają, że każde zadanie klasyfikacji, w którym występują różne częstości pojawiania się obiektów obiektów należących do różnych klas, należy traktować jako problem niezbalansowany. Ze względu na możliwość dekompozycji wieloklasowych zadań klasyfikacji na zadania dwuklasowe problem dysproporcji w licznościach pomiędzy klasami rozpatruje się dla dychotomicznych zagadnień decyzyjnych, w których rozważa się dwie klasy: klasę pozytywną (*ang. positive class*), będącą klasą zdominowaną (*ang. minority class*), oraz klasę negatywną (*ang. negative class*) reprezentującą klasę dominującą (*ang. majority class*). Istotą problemu niezbalansowania jest fakt, że zastosowanie klasycznych mechanizmów uczenia na niezrównoważonym zbiorze danych może prowadzić do faworyzowania przez wyuczony klasyfikator klasy dominującej kosztem klasy zdo-

minowanej. Innymi słowy, typowe podejście może skutkować skonstruowaniem modelu równoważnemu klasyfikatorowi, który przydziela wszystkim obiektom klasę dominującą, niezależnie od wartości wektora cech. Ze względu na zdecydowanie wyższą częstość pojawiania się obiektów z klasy dominującej w stosunku do klasy zdominowanej metoda charakteryzująca się niskim błędem klasyfikacji może charakteryzować się niskim (bądź zerowym) stopniem wykrywalności obserwacji z klasy zdominowanej.

Klasyczny problem uczenia klasyfikatora można rozpatrywać jako zadanie optymalizacji, w którym poszukujemy takiego klasyfikatora Ψ, który minimalizuje jest błąd klasyfikacji E zadany równaniem:

$$E = \frac{1}{N} \sum_{n \in \mathbb{N}} I(\Psi(\mathbf{x}_n) \neq y_n), \tag{1.14}$$

gdzie operator $I(\cdot)$, nazywany indykatorem, przyjmuje wartość 1, jeżeli wyrażenie w argumencie jest prawdziwe, i 0 w przeciwnym wypadku. Alternatywnie, problem uczenia zdefiniować można jako zadanie maksymalizacji poprawności klasyfikacji Acc zadany wzorem:

$$Acc = 1 - E. \tag{1.15}$$

Minimalizacja funkcji celu postaci (1.14) przy mocno niezbalansowanych danych w przypadku trudno separowalnych problemów klasyfikacyjnych może prowadzić do całkowitej dyskryminacji jednej z klas na rzecz klasy dominującej, dlatego dla tego typu problemów konieczny jest wybór innego kryterium optymalizacji. Jednym z kryteriów optymalizacji w uczeniu niezbalansowanym jest błąd ważony E_{Imb} postaci:

$$E_{Imb} = \frac{1}{N_+} \sum_{n \in \mathbb{N}_+} I(\Psi(\mathbf{x}_n) \neq y_n) + \frac{1}{N_-} \sum_{n \in \mathbb{N}_-} I(\Psi(\mathbf{x}_n) \neq y_n), \tag{1.16}$$

gdzie N_+ i N_-, oraz $\mathbb{N}_+$ i $\mathbb{N}_-$ oznaczają kolejno liczności i zbiory indeksów obiektów należących do klasy pozytywnej i negatywnej. Problem uczenia dla danych niezbalansowanych, rozpatrywany jako zadanie optymalizacji, można również zdefiniować jako zadanie maksymalizacji wskaźnika średniej geometrycznej zadanego wzorem:

$$GMean = \sqrt{TP_{rate} \cdot TN_{rate}}, \tag{1.17}$$

gdzie TN_{rate} oznacza wskaźnik specyficzności (znamienności, *ang. specificity*), nazywany również wskaźnikiem TN (*ang. TN rate*), i definiuje się go w następujący sposób:

$$TN_{rate} = \frac{TN}{TN + FP}, \tag{1.18}$$

natomiast TP_{rate} nazywany jest w literaturze wskaźnikiem czułości (*ang. sensitivity*) , bądź też wskaźnikiem TP (*ang. TP rate*), i wyrażony jest wzorem:

$$TP_{rate} = \frac{TP}{TP + FN}. \tag{1.19}$$

	Zaklasyfikowany do klasy pozytywnej	**Zaklasyfikowany do klasy negatywnej**
Należy do klasy pozytywnej	TP (*True positive*)	FN (*False negative*)
Należy do klasy negatywnej	FP (*False positive*)	TN (*True negative*)

Tabela 1.1: Macierz konfuzji dla dychotomicznego zadania klasyfikacji.

Wartości TP (*ang. true positive*), FN (*ang. false negative*), FP (*ang. false positive*), TN (*ang. true negative*), stanowią elementy macierzy konfuzji (*ang. confusion matrix*, Tabela 1.1). Macierz konfuzji, nazywana również macierzą kontyngencji, określa, w jaki sposób klasyfikowane były obiekty z poszczególnych klas. Poszczególne pozycje macierzy definiuje się w następujący sposób:

$$TP = \sum_{n=1}^{N} I(\Psi(x_n) = +1)\ I(y_n = +1), \tag{1.20}$$

$$FN = \sum_{n=1}^{N} I(\Psi(x_n) = -1)\ I(y_n = +1), \tag{1.21}$$

$$FP = \sum_{n=1}^{N} I(\Psi(x_n) = +1)\ I(y_n = -1), \tag{1.22}$$

$$TN = \sum_{n=1}^{N} I(\Psi(x_n) = -1)\ I(y_n = -1), \tag{1.23}$$

gdzie $\mathbb{Y} = \{-1, +1\}$, etykieta $+1$ reprezentuje klasę zdominowaną (pozytywną), natomiast etykieta -1 klasę dominującą (negatywną). Problem uczenia z danych niezbalansowanych w pracy definiowany jest jako zagadnienie znalezienia klasyfikatora Ψ, który maksymalizuje kryterium $GMean$.

Kryterium średniej geometrycznej nie jest jedynym wskaźnikiem niezbalansowania. Alternatywnie do $GMean$ w literaturze poświęconej zagadnieniom niezabalansowania danych rozważa się kryterium AUC (*ang. area under curve*) [11, 63] które reprezentuje pole powierzchni pod krzywą ROC (*ang. Receiver operating characteristic*). AUC definiuje się w następujący sposób:

$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2}, \tag{1.24}$$

gdzie wskaźnik FP_{rate} (*ang. FP rate*) definiuje się następująco:

$$FP_{rate} = \frac{FP}{FP + TN}. \tag{1.25}$$

Pomimo iż problem niezbalansowania danych jest zagadnieniem znanym w literaturze [50], analiza piśmiennictwa pozwala stwierdzić, że problem ten nie jest całkowicie rozwiązany. Dlatego zachodzi konieczność opracowywania nowych, charakteryzujących się wyższą jakością klasyfikacji, metod. W ramach rozprawy proponuje się zastosowanie zespołów klasyfikatorów SVM dedykowanych do rozwiązania problemu dysproporcji w danych. Od proponowanej metody wymaga się by:

1. Charakteryzowała się wyższą niż inne metody klasyfikacji jakością predykcji wyrażoną wskaźnikami $GMean$ (1.17), oraz AUC (1.24).

2. W sposób formalny dało się wykazać zasadność jej stosowania do problemu niezbalansowania.

Opracowana metoda zostanie ponadto zastosowana do rozwiązania rzeczywistego problemu predykcji przeżywalności pooperacyjnej, oraz wybranych zagadnień związanych z podejmowaniem decyzji w systemach opartych na paradygmacie SOA.

1.4 Cel i teza pracy

Celem pracy jest osiągnięcie wysokiej jakości klasyfikacji dla problemu niezbalansowanych danych poprzez zastosowanie zespołów klasyfikatorów SVM. Teza pracy jest następująca:

„Zastosowanie zespołów klasyfikatorów SVM zwiększa skuteczność klasyfikacji w zadaniach o niezbalansowanym zbiorze uczącym."

1.5 Zakres pracy

Zakres pracy obejmuje:

1. Opracowanie wzmacnianego klasyfikatora SVM dla danych niezbalansowanych.
2. Opracowanie dwóch modyfikacji algorytmu wykorzystujących metody selekcji obserwacji informacyjnych.
3. Analizę ilościową opracowanych metod z wykorzystaniem zestawu zbiorów benchmarkowych.
4. Zastosowanie opracowanych metod do zadania predykcji przeżywalności pooperacyjnej.
5. Opracowanie zorientowanej na usługi architektury projektowania usług eksploracji danych celem udostępnienia opracowanych w ramach rozprawy rozwiązań.
6. Zastosowanie opracowanych algorytmów do problemów:
   (a) oceny ryzyka kredytowego;
   (b) detekcji anomalii w sieciach.

1.6 Plan pracy

Plan rozprawy doktorskiej jest następujący:

- **Rozdział 2** stanowi przegląd dostępnych typowych metod klasyfikacji, jak również rozwiązań dedykowanych do problemu niezbalansowania danych.
- **Rozdział 3** zawiera opis opracowanych algorytmów klasyfikacji eliminujących negatywne skutki niezbalansowania w zbiorze uczącym wraz z analizą ich własności.
- **Rozdział 4** prezentuje wyniki badań empirycznych przeprowadzonych celem analizy jakości opracowanych metod.
- **Rozdział 5** przedstawia rzeczywisty przykład zastosowania opracowanych metod dla problemu analizy ryzyka operacyjnego.
- **Rozdział 6** charakteryzuje zorientowaną na usługi architekturę udostępniania rozwiązań uczenia maszynowego opracowaną celem komercjalizacji metod i prezentuje dwa przykłady zastosowania.
- **Rozdział 7** zamyka rozprawę podsumowaniem i kierunkiem dalszych prac.

Rozdział 2

Metody klasyfikacji

W rozdziale dokonano przeglądu najważniejszych metod klasyfikacji. W pierwszej kolejności przedstawiono typowe modele klasyfikatorów nieuwzględniające w swoim działaniu problemu niezbalansowania danych. Druga część rozdziału poświęcona została złożonym metodom klasyfikacji, na których bazuje opracowany w ramach rozprawy zespół klasyfikatorów SVM. Ostatnia część stanowi syntetyczny opis metod dedykowanych do rozwiązania zagadnienia niezbalansowania danych.

2.1 Metody klasyfikacji dla danych zbalansowanych

Jeden z podstawowych podziałów metod klasyfikacji wyróżnia modele proste (*ang. individuals*) i złożone (*ang. compound classification models, complex classifcation models*). Autor pracy [78] definiuje modele złożone jako klasyfikatory, w przypadku których decyzja o przynależności obiektu do klasy nie jest operacją jednorazową, ale jest wynikiem mniej lub bardziej złożonego procesu decyzyjnego. Innymi słowy, jeżeli w danym modelu klasyfikatora, traktowanego jako system podejmowania decyzji, wyróżnić można podsystem realizujący odrębne, bądź to samo zadanie klasyfikacyjne, to wówczas taki model klasyfikacyjny jest modelem złożonym. Modele klasyfikacyjne, które nie spełniają tego warunku są nazywane modelami prostymi.

2.1.1 Proste modele klasyfikacyjne

W literaturze przedmiotu uczenia maszynowego proponuje się szereg prostych modeli klasyfikacyjnych. Do najpopularniejszych i najpowszechniej stosowanych należą m. in. :

- sieci neuronowe [31, 73, 108, 120, 149];
- maszyny wektorów wspierających (*ang. support vector machines, SVM*) [10, 73, 135];
- regresja logistyczna [10, 73];
- algorytm Naiwnego Bayesa;
- algorytm K najbliższych sąsiadów;
- reguły decyzyjne [18, 20, 29, 141];
- drzewa decyzyjne [15, 105, 106];

Sieci neuronowe, których budowa inspirowana jest budową mózgu, wykorzystują do klasyfikacji szczególną postać funkcji dyskryminującej nazywanej perceptronem. Pojedynczy perceptron może być stosowany jedynie do dychotomicznych, separowalnych liniowo problemów klasyfikacyjnych. Dla problemów bardziej złożonych neurony łączone są w rozmaite struktury sieciowe, o złożoności dostosowanej do problemów klasyfikacyjnych, które modelują. Literatura wyróżnia wiele modeli sieci neuronowych, które szeregowane są w postaci rozmaitych taksonomii [73, 108, 149]. Najpowszechniej stosowanym modelem sieci neuronowej jest jednokierunkowy, wielowarstwowy perceptron. Uczenie klasyfikatora reprezentowanego w postaci wielowarstwowego perceptronu odbywa się poprzez wyznaczenie wartości wag wszystkich neuronów znajdujących się w sieci. Jednym z najczęściej stosowanych algorytmów uczenia sieci neuronowej jest algorytm wstecznej propagacji błędów (*ang. backpropagation*) [73, 108, 149].

Rozwój sieci neuronowych wziął swój początek w zastosowaniach aplikacyjnych, które w konsekwencji doprowadziły do opracowania formalnych rozwiązań teoretycznych [73], natomiast rozwój maszyn wektorów wspierających przebiegał w kierunku przeciwnym. Koncepcja SVM wywodzi się od teorii statystycznego uczenia zaproponowanej po

raz pierwszy przez Vapnika i Chervonenkisa [135]. Klasyfikatory SVM zostały szczegółowo opisane w Rozdziale 3.

Regresja logistyczna jest typowym przykładem statystycznego dyskryminującego modelu klasyfikacyjnego. Jedną z zalet stosowania regresji logistycznej jest niewielka liczba parametrów, które muszą być oszacowane w procesie uczenia. Typową metodą stosowaną do estymacji parametrów jest metoda Newtona-Rapsona, nazywana iteracyjnie ważoną metodą najmniejszych kwadratów, która została opisana w pozycji [73].

Algorytm Naiwnego Bayesa (*ang. Naive Bayes*) należy do najpopularniejszych i najczęściej stosowanych algorytmów klasyfikacyjnych. Popularność tego klasyfikatora wynika z jego prostoty, probabilistycznych podstaw teoretycznych, niewrażliwości na problem brakujących wartości atrybutów [50] a także z możliwości aktualizacji w procesie uczenia przyrostowego [131]. Klasyfikator Naiwnego Bayesa jest typowym podejściem generującym, którego fundamentalną cechą jest założenie, że zmienne losowe charakteryzujące poszczególne cechy obiektu są niezależne.

Algorytm K Najbliższych Sąsiadów jest (*ang. K Nearest Neighbours,* K*-*NN*)* jednym z najpopularniejszych klasyfikatorów nieparametrycznych. Charakterystyczną cechą algorytmu K-NN jest brak wyodrębnionego procesu uczenia. Klasyfikator przechowuje cały zbiór danych na potrzeby procesu klasyfikacji w którym wyznaczana jest odległość pomiędzy klasyfikowanym obiektem, a wszystkimi obiektami znajdującymi się w zbiorze uczącym. Następnie analizowany jest rozkład klas K obiektów (nazywanych najbliższymi sąsiadami), których odległości od klasyfikowanego punktu są najmniejsze. Obiekt przydzielony zostanie do klasy, która ma największą liczbę przedstawicieli pośród K najbliższych sąsiadów.

Metoda K-NN cechuje się wysoką skutecznością klasyfikacji gdy dane są rozłożone gęsto i stanowią reprezentatywną próbę dla zadanego problemu [73]. Sytuacja taka jest niezwykle rzadka w przypadku rzeczywistych problemów klasyfikacji, dlatego algorytm K-NN stosuje się głównie jako metodę stanowiącą punkt odniesienia do oceny skuteczności innych metod, bądź też jako komponent klasyfikatorów bardziej złożonych.

Bardzo ważną grupę klasyfikatorów nieparametrycznych stanowią reguły decyzyjne. Główną cechą tej grupy algorytmów jest prosta i zrozumiała dla człowieka reprezentacja wiedzy, która zawarta jest w zbiorze kompletnych i niesprzecznych reguł. Regułowa reprezentacja wiedzy daje możliwość oceny klasyfikatora nie tylko poprzez eksperymentalne

badanie poprawności klasyfikacji i innych wskaźników jakości wykorzystywanych w uczeniu maszynowym, ale również poprzez analizę zrozumiałej dla człowieka wiedzy zawartej w zbiorze reguł decyzyjnych. Dzięki zrozumiałej dla człowieka reprezentacji wiedzy klasyfikator reprezentowany przez reguły decyzyjne może być modyfikowany przez człowieka poprzez wstawianie i eliminację reguł, czy też wykorzystany w procesie wnioskowania niezależnie od implementacji.

Każda z reguł decyzyjnych reprezentowana jest poprzez implikację, dla której strona implikująca stanowi koniunkcję co najmniej D formuł elementarnych reprezentujących podzbiory wartości poszczególnych atrybutów, natomiast strona implikowana reprezentowana jest przez formułę elementarną odnoszącą się do jednej z możliwych etykiet klas [18]. Proces klasyfikacji obiektu odbywa się poprzez analizę wartości logicznej strony implikującej dla każdej z reguł. Jeżeli wartość logiczna dla koniunkcji strony implikującej jest spełniona dla zadanych wartości atrybutów obiektu, to zostaje sklasyfikowany do klasy reprezentowanej przez implikowaną formułę elementarną. O takim obiekcie mówi się, że został pokryty przez daną regułę. Aby wynikiem procesu klasyfikacji dla dowolnego obiektu z przestrzeni $\mathbb{X}$ była dokładnie jedna etykieta klasy, to zbiór reguł musi być kompletny i niesprzeczny. Zbiór reguł decyzyjnych jest kompletny wtedy, i tylko wtedy, gdy dla każdego obiektu z przestrzeni $\mathbb{X}$ istnieje co najmniej jedna reguła, która pokrywa dany obiekt. Jeżeli w przestrzeni $\mathbb{X}$ nie istnieje obiekt, który jest pokryty przez dwie reguły reprezentujące różne klasy, to zbiór reguł jest niesprzeczny.

Zbiór reguł reprezentujących klasyfikator, ze względu na zrozumiałą dla człowieka reprezentację wiedzy, może być podany przez eksperta, jednak w większości przypadków jest on generowany w procesie uczenia klasyfikatora z wykorzystaniem zbioru uczącego. Jednym z podstawowych algorytmów wykorzystywanych do generowania reguł jest zaproponowany w pracy [20] algorytm PRISM. Rozwiązanie to bazuje na typowym podejściu generowania reguł nazywanym "separuj i zwyciężaj" (*ang. separate-and-conquer*) [141], którego główną ideą jest iteracyjne budowanie reguł, które pokrywają jak największą liczbę obserwacji należących do jednej klasy, i nie pokrywają obserwacji z innych klas. Obserwacje pokryte przez wygenerowaną regułę są eliminowane ze zbioru uczącego, a proces budowy klasyfikatora kończy się, gdy w zbiorze uczącym nie będzie już żadnych obiektów. Główną wadą takiego podejścia jest zbytnie dopasowywanie się generowanych reguł do obiektów znajdujących się zbiorze uczącym. Większość reguł generowanych jest z wyko-

rzystaniem jednego bądź kliku elementów ze zbioru uczącego, co prowadzi do budowy licznego zbioru zawierającego reguły o wysokiej szczegółowości. Propozycją rozwinięcia algorytmu PRISM, która wykorzystuje mechanizmy ucinania (*ang. pruning*) reguł celem generalizacji i redukcji ich liczby, jest algorytm RIPPER (*ang. Repeated Incremental Pruning to Produce Error Reduction*) [29]. W pierwszym kroku algorytmu zbiór uczący dzielony jest na dwa podzbiory: zbiór generujący (*ang. growing set*), oraz zbiór ucinający (*ang. pruning set*). W kolejnym kroku zbiór generujący jest wykorzystany do konstrukcji jednej reguły poprzez zastosowanie algorytmu PRISM. Wygenerowana reguła jest następnie generalizowana poprzez eliminację formuły elementarnej reprezentującej ostatnio dodany atrybut. Jeżeli proces eliminacji formuły elementarnej nie prowadzi do obniżenia się poprawności klasyfikacji wygenerowanej reguły na zbiorze ucinającym to proces eliminacji formuł elementarnych strony implikującej jest kontynuowany. W przeciwnym wypadku reguła zostaje dodana do zbioru reguł wynikowych, a wszystkie obiekty należące do zbioru generującego i ucinającego pokryte przez regułę są eliminowane. Przedstawiona procedura generowania jest powtarzana do momentu wyczerpania reguł ze zbioru generującego.

Alternatywną do klasyfikatorów regułowych grupę metod stanowią drzewa decyzyjne. Podobnie jak poprzednio scharakteryzowane reguły decyzyjne, również i te nieparametryczne klasyfikatory charakteryzują się zrozumiałą dla człowieka reprezentacją wiedzy. Drzewo decyzyjne składa się z wierzchołków które reprezentują cechy klasyfikowanych obiektów, oraz z krawędzi które reprezentują przedziały możliwych wartości cech. Aby drzewo decyzyjne skonstruowane zostało poprawnie, krawędzie wychodzące z każdego wierzchołka muszą być reprezentowane przez rozłączne warunki, które w sumie pokrywają całą przestrzeń możliwych wartości cechy. Innymi słowy, dla każdej możliwej wartości cechy może zostać wybrana dokładnie jedna krawędź wychodząca z wierzchołka, który ją reprezentuje. Każda ze ścieżek drzewa decyzyjnego zakończona jest liściem, który reprezentuje jedną z możliwych etykiet klas. Proces klasyfikacji odbywa się poprzez schodzenie wgłąb drzewa za każdym razem obierając tą krawędź, która spełnia wymaganie odnośnie wartości rozpatrywanej cechy obiektu. Założenie o kompletności i rozłączności krawędzi drzewa zapewnia, że dla dowolnego obiektu z przestrzeni $\mathbb{X}$ zostanie obrana dokładnie jedna ścieżka prowadząca od korzenia drzewa do liścia z etykietą.

Typowe podejścia do generowania reguł bazują na zasadzie "separuj i zwyciężaj". Analogiczna zasada, "dziel i zwyciężaj" (*ang. divide-and-conquer*), sformułowana została dla

algorytmów budowania drzew decyzyjnych. Typowym podejściem wykorzystującym wspomnianą zasadę jest zaproponowany przez Quinlana algorytm ID3 [105]. Procedura konstrukcji drzewa odbywa się rekurencyjnie. W pierwszym kroku wybierana jest cecha, która ma być umieszczona w korzeniu drzewa. Dla każdej wartości nominalnej cechy generowana jest wychodząca z korzenia krawędź. Zbiór uczący dzielony jest na podzbiory z których każdy zawiera elementy o różnych wartościach cechy umieszczonej w korzeniu. Proces wyboru cechy i dalszego podziału podzbiorów zbioru uczącego jest powtarzany do momentu, w którym aktualny podzbiór będzie zawierał jedynie obiekty jednej klasy. Procedura kończy się, gdy każda z możliwych ścieżek w drzewie zostanie zakończona etykietą klasy do której należą wszystkie elementy otrzymanego w wyniku podziału podzbioru uczącego. Jak kryterium wyboru cechy dla rozpatrywanego wierzchołka ID3 stosuje miarę entropii warunkowej.

Zasadniczą wadą algorytmu ID3 jest, podobnie jak w przypadku algorytmu PRISM, zbytnie dopasowanie się drzewa do zbioru uczącego. Ponadto, algorytm ID3 został zaprojektowany dla obiektów opisywanych jedynie atrybutami nominalnymi. Rozszerzeniem koncepcji ID3 jest algorytm C 4.5 [106], który daje możliwość budowania drzew decyzyjnych do klasyfikacji obiektów zawierających atrybuty numeryczne, posiada wbudowane mechanizmy obsługi brakujących wartości atrybutów, oraz wykorzystuje mechanizmy ucinania do generalizacji drzewa. Bardzo popularnym algorytmem jest również zaproponowane przez Breimana drzewo klasyfikacyjno-regresyjne (*ang. classification and regression tree, CART*) [15].

2.1.2 Złożone modele klasyfikacyjne

Złożone metody klasyfikacji stosuje się w przypadkach, kiedy opisane w poprzednim rozdziale proste metody charakteryzują się niską poprawnością klasyfikacji [153]. Wśród klasyfikatorów złożonych wyróżnia się: klasyfikatory wieloetapowe, wielozadaniowe, dwupoziomowe [78] oraz zespoły klasyfikatorów [73, 76]. Rozpoznawanie wieloetapowe realizuje następującą sekwencję czynności klasyfikacyjnych. W pierwszym kroku z wejściowego zbioru cech rozpatrywanych w zadanym problemie klasyfikacji wybierany jest podzbiór określonych cech, które stanowią podstawę do podjęcia decyzji klasyfikacyjnej na pierwszym etapie ograniczającej zbiór możliwych etykiet klas. Decyzja podjęta na pierw-

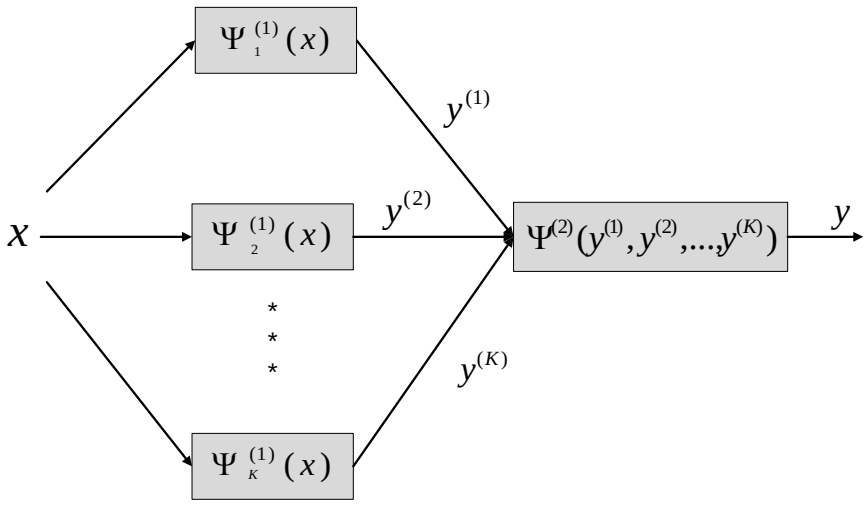
szym etapie determinuje wybór podzbioru cech dla kolejnego etapu decyzyjnego. Proces podejmowania decyzji lokalnych i wyboru podzbioru cech wykonywany jest do momentu podjęcia jednoznacznej decyzji będącej wynikiem klasyfikacji.

Charakterystyczną cechą wielozadaniowej klasyfikacji jest to, że rozpoznawany obiekt podlega wielokrotnej etykietyzacji, przy czym w każdym z zadań rozpatruje się odmienny zestaw klas, ich liczbę, oraz praktyczne znaczenie. Wynikiem klasyfikacji wielozadaniowej nie jest więc pojedyncza wartość klasy, a wektor etykiet wielowymiarowej przestrzeni klas. W przypadku klasyfikatorów dwupoziomowych decyzja podejmowana jest w następujący sposób. Na pierwszym poziomie K autonomicznych klasyfikatorów podejmuje decyzje rozwiązując lokalne zadania klasyfikacji. Uzyskane przez klasyfikatory etykiety klas przekazywane są jako wejście do klasyfikatora znajdującego się na drugim poziomie. Wektorem cech dla drugiego poziomu jest więc zestaw etykiet klas uzyskanych przez każdy z K klasyfikatorów.

Wymienione złożone modele dedykowane są dla zadań klasyfikacji, w przypadku których natura wymaga kompleksowych rozwiązań. W odróżnieniu od prezentowanych podejść, zespoły klasyfikatorów stosuje się do każdego problemu decyzyjnego, który może być zdefiniowany jako zadanie klasyfikacji. Celem stosowania zespołów klasyfikatorów nie jest modelowanie złożonych problemów decyzyjnych, a poprawa skuteczności podejmowanych decyzji poprzez stosowanie odpowiednich technik dywersyfikacji w procesie konstrukcji złożonego modelu [89]. Zespoły klasyfikatorów stosuje się również celem balansowania danych, w których występują dysproporcje pomiędzy klasami [49, 56, 119, 137], bądź też do rozwiązania problemu brakujących wartości atrybutów [50, 100].

Schemat modelu zespołu klasyfikatorów przedstawiony został na Rysunku 2.1. Na pierwszym etapie klasyfikacji wyróżnia się K klasyfikatorów $\Psi_k^{(1)}$, nazywanych klasyfikatorami bazowymi (*ang. base classifiers*), podejmujących autonomiczne decyzje klasyfikacyjne. Decyzje podjęte przez klasyfikatory bazowe przekazywane są do klasyfikatora łączącego $\Psi^{(2)}$ (*ang. combiner*), który dokonuje finalnej klasyfikacji.

Kluczowym elementem w zagadnieniach związanych z zespołami klasyfikatorów jest proces konstrukcji klasyfikatorów bazowych. Na ogół za modele bazowe przyjmuje się tzw. klasyfikatory słabe (*ang. week learners*), charakteryzujące się niską poprawnością klasyfikacji (nieznacznie wyższą niż 50%) i dużą wrażliwością na niewielkie zmiany w zbiorze uczącym. Zakładając, że klasyfikatory bazowe podejmują decyzje niezależnie, a prawdo-



Rysunek 2.1: Schemat modelu wzmacnianego klasyfikatora.

podobieństwo podjęcia trafnej decyzji przez każdy z klasyfikatorów bazowych jest równe p, gdzie $p > 0.5$, to prawdopodobieństwo podjęcia trafnych decyzji przez k z K klasyfikatorów jest realizacją rozkładu Bernoulliego (*ang. binomial distribution*) z parametrem p. Ponadto, prawdopodobieństwo, że większość z K klasyfikatorów podejmie trafne decyzje, t.j. $P(\mathbf{K} > \frac{K}{2}|p)$, gdzie $\mathbf{K}$ jest zmienną losową z rozkładu Bernoulliego, jest wyższe niż prawdopodobieństwo p określające sukces w decyzjach podejmowanych przez klasyfikatory indywidualnie.

W praktyce znalezienie klasyfikatorów podejmujących niezależne decyzje jest trudne, dlatego konieczne jest stosowanie technik różnicowania (dywersyfikacji, *ang. diversification techniques*) celem wymuszenia niezależności pomiędzy modelami. W zadaniu wyznaczania zróżnicowanych klasyfikatorów wyróżnia się dwa problemy:

- brak jest jednoznacznie zdefiniowanej miary dywersyfikacji,
- dostępność tylko jednego, niepodzielonego zbioru danych, który ma być wykorzystany w procesie uczenia zdywersyfikowanych klasyfikatorów bazowych.

Odnosząc się do pierwszego problemu Kuncheva w swojej pracy [77] dokonuje przegląd-

du miar stosowanych do badania zróżnicowania klasyfikatorów, podkreślając jednocześnie, że nie da się jednoznacznie stwierdzić, która z miar powinna stanowić kryterium w konstrukcji zespołów klasyfikatorów. Drugie z zagadnień wymusza zaproponowanie sposobu generowania zróżnicowanych zbiorów uczących ze zbioru wejściowego celem otrzymania niezależnych modeli składowych. Brak jednoznaczności w definiowaniu zróżnicowania klasyfikatorów bazowych doprowadził do powstania szeregu metod uczenia zespołów klasyfikatorów.

Metody dywersyfikacji klasyfikatorów bazowych

Zróżnicowanie klasyfikatorów bazowych może być osiągnięte poprzez:

- wprowadzenie losowości w procesie uczenia,
- zmianę parametrów uczenia,
- poprzez wprowadzenie modyfikacji w wejściowym zbiorem treningowym.

Pierwsze dwie z wymienionych technik dywersyfikacji odnoszą się do sytuacji, w której każdy z klasyfikatorów bazowych konstruowany jest z innymi wartościami parametrów jakie algorytm wykorzystuje w procesie uczenia. Przykładowo, dywersyfikacja może zostać osiągnięta poprzez dobór różnych konfiguracji neuronów jeżeli modelem bazowym jest sieć neuronowa [55], różnych liczb sąsiadów dla algorytmu KNN, czy też poprzez różne wartości parametru radialnej funkcji bazowej jądra (ang. radial basis kernel function) dla klasyfikatorów SVM [83].

Znacznie szerszą grupę stanowią techniki dywersyfikacji, w których dokonuje się zmian w wejściowym zbiorze uczącym S_N generując K różniących się od siebie zbiorów $\mathbb{S}_{N_1}^{(1)}$, $\ldots$, $\mathbb{S}_{N_K}^{(1)}$. Wygenerowane zbiory uczące wykorzystywane są w budowie kolejnych klasyfikatorów bazowych $\Psi_1^{(1)}, \ldots, \Psi_K^{(1)}$. Dywersyfikacja zbiorów $\mathbb{S}_{N_1}^{(1)}, \ldots, \mathbb{S}_{N_K}^{(1)}$ może być osiągnięta poprzez:

- losowanie elementów ze zbioru uczącego S_N [12, 47, 110],
- wykorzystanie różnych wag przyporządkowanych obserwacjom [42, 119]
- ograniczanie przestrzeni cech obiektów [14, 100],

- generowanie sztucznych obserwacji [25, 26, 53, 89],
- podmianę etykiet klas [13, 87, 152, 150].

W ramach pierwszej grupy technik zbiory uczące wykorzystywane do konstrukcji klasyfikatorów bazowych są generowane poprzez losowanie z zadanego rozkładu obiektów z wejściowego zbioru uczącego. Najprostszą i najbardziej popularną techniką generowania zbiorów bazowych jest N-krotne losowanie ze zwracaniem stosowane w algorytmie *bagging* (*ang. bootstrap sampling and aggregation*) [12] . Każdy ze zbiorów bazowych $\mathbb{S}_{N_k}^{(k)}$ jest generowany niezależnie z rozkładu jednostajnego (Algorytm 1). Model łączenia klasyfikatorów bazowych $\Psi^{(2)}$, w przypadku algorytmu *bagging*, sprowadza się do techniki klasycznego głosowania, czyli wyboru klasy najczęściej zwracanej przez klasyfikatory bazowe.

Algorithm 1: *Bagging*

Input : Zbiór uczący $\mathbb{S}_N$, zbiór możliwych klas $\mathbb{Y}$

Output: Klasyfikator wzmacniany *baggingiem*: $\Psi(\mathbf{x}) = \arg\max_{y \in \mathbb{Y}} \sum_{k=1}^{K} I(\Psi_k^{(1)}(\mathbf{x}) = y)$

1 **for** $k = 1 \rightarrow K$ **do**
2 Wyznacz zbiór $\mathbb{S}_N^{(k)}$ poprzez N-krotne losowanie ze zwracaniem obiektów ze zbioru $\mathbb{S}_N$;
3 Wyucz klasyfikator $\Psi_k^{(1)}$ na zbiorze uczącym $\mathbb{S}_N^{(k)}$;
4 **end**

Inną grupę algorytmów wykorzystujących losowanie obiektów jako technikę różnicowania zbiorów bazowych stanowią algorytmy wzmacniania (*ang. boosting*). W odróżnieniu od metody *baggingu*, w *boostingu* rozkład, z którego wykonywane jest losowanie podlega modyfikacji w kolejnych iteracjach procesu konstrukcji złożonego modelu, a sposób modyfikacji uwarunkowany jest poprawnością klasyfikacji już utworzonych klasyfikatorów bazowych. W kolejnych iteracjach z większym prawdopodobieństwem wybierane są te obserwacje z wejściowego zbioru danych, które były błędnie klasyfikowane przez wyuczone w poprzednich iteracjach modele bazowe. Jednym z najpopularniejszych i charakteryzujących się najwyższą poprawnością klasyfikacji algorytmem *boostingu* jest zaproponowany

w pracy [47] algorytm *AdaBoost.M1* (Algorytm 2). Rozkład, według którego są losowane obserwacje do kolejnych zbiorów bazowych, modyfikowany jest w ten sposób, że parametry rozkładu $D_k(n)$ dla obserwacji poprawnie sklasyfikowanych przemnażane są przez współczynnik β_k, gdzie $\beta_k \in [0, 1]$. Następnie dokonywana jest normalizacja parametrów celem otrzymania rozkładu (kroki 10 i 11 w Algorytmie 2). Współczynnik β_k zależny jest od ważonego błędu ϵ_k klasyfikatora bazowego $\Psi_k^{(1)}$ na zbiorze S_N i przyjmuje wartość 0, gdy $\epsilon_k = 0$, oraz 1, gdy $\epsilon_k = 0.5$. Ostateczna klasyfikacja odbywa się z wykorzystaniem ważonego głosowania, gdzie waga każdego z klasyfikatorów bazowych stanowi logarytm naturalny odwrotności współczynnika β_k. Charakterystyczną cechą tego algorytmu wzmacniania jest fakt, iż bardzo szybko minimalizuje błąd klasyfikacji, gdyż, jak wykazano w [48], procedura konstrukcji klasyfikatorów bazowych złożonego algorytmu sprowadza się do iteracyjnej minimalizacji wykładniczej funkcji błędu postaci:

$$E_{\exp} = \sum_{n=1}^{N} \exp\{-y_n g_k(\mathbf{x}_n)\}, \tag{2.1}$$

gdzie $g_k(\mathbf{x})$ stanowi kombinację liniową k klasyfikatorów bazowych wykorzystywaną do podjęcia ostatecznej decyzji:

$$g_k(\mathbf{x}) = \frac{1}{2} \sum_{l=1}^{k} c_l \Psi_l^{(1)}(\mathbf{x}), \tag{2.2}$$

c_l jest wagą klasyfikatora bazowego w zespole, a zbiór możliwych etykiet klas jest następujący, $\mathbb{Y} = \{-1, 1\}$.

Drugą grupę technik dywersyfikacji stanowią metody które wykorzystują zróżnicowanie wag (kosztów) przypisywanych obserwacjom ze zbioru uczącego. Metody wykorzystujące opisaną technikę różnicowania nazywa się metodami wrażliwymi na koszt (*ang. cost-sesitive methods*). Do najpopularniejszych metod zalicza się algorytm *AdaCost* [42], który stanowi rozwinięcie metod wzmacniania uwzględniające różne koszty błędnych klasyfikacji poszczególnych obserwacji ze zbioru uczącego. Metody wrażliwe na koszt stosuje się głównie w przypadkach, w których występują znaczne różnice w kosztach dotyczących błędnych decyzji w obrębie jednego problemu decyzyjnego, takich jak diagnostyka medyczna, czy wykrywanie SPAMu. Złożone metody klasyfikacji wykorzystujące wspomnianą technikę dywersyfikacji znajdują również zastosowanie w konstrukcji modeli decyzyjnych z niezbalansowanych danych [119].

Algorithm 2: Algorytm *AdaBoost.M1*

Input : Zbiór uczący $\mathbb{S}_N$, zbiór możliwych klas $\mathbb{Y}$

Output: Klasyfikator AdaBoost.M1: $\Psi(\mathbf{x}) = \arg\max_{y \in \mathbb{Y}} \sum_{k=1}^{K} \ln \frac{1}{\beta_k} I(\Psi_k^{(1)}(\mathbf{x}) = y)$

Zadaj rozkład początkowy D_1, w ten sposób, że $D_1(n) = \frac{1}{N}$ dla każdego $n \in \{1, \ldots, N\}$;

for $k = 1 \rightarrow K$ **do**

  Wyznacz zbiór $\mathbb{S}_N^{(k)}$ poprzez N-krotne losowanie ze zwracaniem obiektów ze zbioru $\mathbb{S}_N$ zgodnie z rozkładem D_k ;

  Wyucz klasyfikator $\Psi_k^{(1)}$ na zbiorze uczącym $\mathbb{S}_N^{(k)}$;

  Wyznacz błąd ważony klasyfikatora $\Psi_k^{(1)}$: $\epsilon_k \longleftarrow \sum_{n=1}^{N} D_k(n) I(\Psi_k^{(1)}(\mathbf{x_n}) \neq y_n)$;

  if $\epsilon_k > 0.5$ **then**

    $D_k(n) = \frac{1}{N}$ dla każdego $n \in \{1, \ldots, N\}$;

  else

    $\beta_k \longleftarrow \frac{\epsilon_k}{1-\epsilon_k}$;

    Aktualizuj D_k: $D_{k+1}(n) = D_k(n) \beta_k^{1 - I(\Psi_k^{(1)}(\mathbf{x_n}) \neq y_n)}$ dla każdego $n \in \{1, \ldots, N\}$;

    Normalizuj D_k dzieląc przez sumę $\sum_{n=1}^{N} D_k(n)$;

  end

end

W ramach trzeciej grupy technik dywersyfikacji każdy z klasyfikatorów bazowych konstruowany jest na zbiorze danych z zredukowanym wektorem cech. Najpopularniejszą metodą wykorzystującą wspomnianą technikę różnicowania jest zaproponowany przez Breimana algorytm Lasów Losowych (*ang. Random Forests*) [14]. W każdej iteracji budowania nowego klasyfikatora bazowego dla Lasu Losowego generowany jest zbiór uczący poprzez N-krotne losowanie ze zwracaniem ze zbioru wejściowego. W kolejnym kroku losowany jest d-elementowy podzbiór cech, gdzie $d << D$. Wygenerowany i zredukowany do d wylosowanych cech zbiór wykorzystany jest do budowy klasyfikatora bazowego, będącego modelem drzewa decyzyjnego bądź też algorytmem regułowym. Redukcję podzbioru cech stosuje się powszechnie również do rozwiązania problemu brakujących wartości atrybutów,

czego przykładem jest algorytm *Learn++.MF* zaproponowany w pracy [100].

Kolejną grupę metod różnicowania stanowią techniki polegające na generowaniu sztucznych obserwacji na podstawie wejściowego zbioru uczącego. Główną zaletą tej grupy metod jest fakt, że niwelują negatywne skutki wynikające z niezbalansowania zbioru uczącego. Jednym z najpopularniejszych algorytmów wzmacniania klasyfikatorów wykorzystującą technikę generowania syntetycznych próbek jest algorytm DECORATE (*ang. Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples*) [89]. Każdy ze zbiorów bazowych S_{N_k} składa się z obiektów należących do S_N poszerzonych o zestaw sztucznie wygenerowanych obserwacji z rozkładu o wartościach parametrów oszacowanych z danych wejściowych. Każdy nowo utworzony klasyfikator $\Psi_k^{(1)}$ włączany jest do zbioru klasyfikatorów bazowych jeżeli zmniejsza błąd klasyfikatora łącznego. Ostateczna decyzja w algorytmie DECORATE podejmowana jest z wykorzystaniem ważonego głosowania. W ramach tej grupy metod występują również algorytmy dedykowane do rozwiązania problemu niezbalansowania danych, takie jak *SMOTEBoost* [25], *RAMOBoost* [26], oraz *DataBoost-IM* [53].

Ostatnią z wymienionych grup stanowią techniki dywersyfikacji polegające na losowej zmianie etykiet klas elementów w zbiorze uczącym. Breiman w swojej pracy [13] proponuje, by zbiory bazowe generowane były poprzez zmianę etykiety klasy z określonym prawdopodobieństwem zależnym od proporcji pomiędzy klasami oraz od parametru podmiany klasy. Rozważania teoretyczne poparte wynikami analizy eksperymentalnej wykonanej na reprezentatywnej liczbie zbiorów danych wykazały, że proponowana przez Breimana metoda z podmianą klas daje wyniki zbliżone do algorytmów *baggingu* przy odpowiednim doborze prawdopodobieństwa zamiany etykiet zachowującego rozkład generujący obserwacje z rozpatrywanych klas. Autorzy pracy [87] analizując opracowaną przez Breimana metodę wykazali, że dla dużej liczby (ok. 1000) klasyfikatorów bazowych zachowanie rozkładu generującego nie jest konieczne, aby osiągnąć wysoką jakość klasyfikacji. Metody wykorzystujące jako technikę dywersyfikacji polegającą na podmianie etykiet klas stosowane były również z powodzeniem do rozwiązania problemu niezbalansowania [150, 152].

Metody łączenia klasyfikatorów bazowych

W poprzednim podrozdziale omówione zostały podstawowe metody dywersyfikacji klasyfikatorów bazowych. Inną istotną kwestią jest zagadanie ich łączenia celem podjęcia ostatecznej decyzji klasyfikacyjnej na podstawie składowych decyzji pochodzących od klasyfikatorów bazowych. Wyróżnia się dwie grupy metod łączenia klasyfikatorów, w zależności od charakteru decyzji zwracanych przez klasyfikatory bazowe [153]:

1. Metody łączenia na podstawie deterministycznych decyzji klasyfikatorów bazowych - klasyfikatory bazowe zwracają etykiety klas.

2. Metody łączenia na podstawie niepewnych decyzji klasyfikatorów bazowych - klasyfikatory bazowe zwracają wartości miary niepewności związane z przynależnością obiektu do każdej z rozpatrywanych klas.

W przypadku pierwszej grupy metod każdy z klasyfikatorów bazowych zwraca element ze zbioru $\mathbb{Y}$. Jedną z typowych metod łączenia jest algorytm głosowania, który wyraża się następującym wzorem:

$$\Psi^{(2)}(\mathbf{x}) = \arg\max_{y\in\mathbb{Y}} \sum_{k=1}^{K} I(\Psi_k^{(1)}(\mathbf{x}) = y). \tag{2.3}$$

Klasyfikator łączący $\Psi^{(2)}$ dokonuje wyboru tej klasy ze zbioru $\mathbb{Y}$, która została zwrócona przez największą liczbę klasyfikatorów bazowych. Algorytm głosowania stosuje się w klasycznej wersji algorytmu *baggingu*. Rozszerzeniem powyższej metody łączenia klasyfikatorów jest algorytm ważonego głosowania:

$$\Psi^{(2)}(\mathbf{x}) = \arg\max_{y\in\mathbb{Y}} \sum_{k=1}^{K} c_k I(\Psi_k^{(1)}(\mathbf{x}) = y). \tag{2.4}$$

Waga c_k utożsamiana jest z jakością klasyfikatora bazowego i stanowi ona funkcję odwrotności błędu klasyfikacji, podobnie jak w przypadku algorytmów *boostingu*, czy *DECORATE*, bądź też funkcję innych wskaźników jakości, takich jak wartość geometrycznej średniej poprawności klasyfikacji pierwszego i drugiego rodzaju [137].

Druga grupa metod łączenia klasyfikatorów bazowych zakłada, że wynikiem ich działania nie jest deterministyczna decyzja na temat przydziału danego obiektu do klasy, lecz

rozkład niepewności związany z przynależnością obiektu do każdej z klas. Znormalizowany opis niepewności utożsamia się z rozkładem *a posteriori* $p_k(y|\mathbf{x})$, gdzie zmienna losowa $\mathbf{Y}$ przyjmuje wartości ze zbioru $\mathbb{Y} = \{0, \ldots, Y-1\}$. Istnieje szereg metod łączenia klasyfikatorów bazowych zwracających wynik klasyfikacji w postaci probabilistycznej. Do najpopularniejszych metod zalicza się niedeterministyczny odpowiednik głosowania, w którym wybiera się klasę o najwyższej wartości sumy prawdopodobieństw:

$$\Psi^{(2)}(\mathbf{x}) = \arg\max_{y \in \mathbb{Y}} \sum_{k=1}^{K} p_k(y|\mathbf{x}). \tag{2.5}$$

Alternatywnie do sumy prawdopodobieństw stosuje się również wariant łączenia klasyfikatorów z iloczynem:

$$\Psi^{(2)}(\mathbf{x}) = \arg\max_{y \in \mathbb{Y}} \prod_{k=1}^{K} p_k(y|\mathbf{x}). \tag{2.6}$$

W pracy [76] wyróżniono alternatywne do sumy i iloczynu podejścia polegające na wyznaczaniu maksimum i minimum z prawdopodobieństw zawracanych przez klasyfikatory bazowe. Kolejnym podejściem łączenia klasyfikatorów w wariancie probabilistycznym jest podejście wykorzystujące generalizację stosową (*ang. stacked generalization*) [142]:

$$\Psi^{(2)}(\mathbf{x}) = \arg\max_{y \in \mathbb{Y}} \sum_{k=1}^{K} c_k^{(y)} p_k(y|\mathbf{x}). \tag{2.7}$$

Parametry $c_k^{(y)}$ wyznaczane są za pomocą metody najmniejszych kwadratów. Generalizacja stosowa, w odróżnieniu od innych metod łączenia klasyfikatorów, zakłada, że klasyfikator łączący $\Psi^{(2)}$ jest również budowany z wykorzystaniem zbioru uczącego. Badania empiryczne dotyczące stosowania różnych metod łączenia klasyfikatorów przeprowadzone przez autora rozprawy w pracy [153] wykazują, że stosowanie technik głosowania i ważonego głosowania daje zadowalające wyniki w konstrukcji modeli decyzyjnych, a stosowanie bardziej złożonych technik nie zwiększa poprawności klasyfikacji.

2.2 Metody przeciwdziałania niezbalansowanym danym

Opisane dotychczas metody klasyfikacji nie posiadały mechanizmów obsługi danych charakteryzujących się nierównym rozkładem klas. Sformułowany w rozprawie problem

danych niezbalansowania jest zagadnieniem częściowo wyjaśnionym. W literaturze wyróżnia się szereg technik stosowanych do rozwiązania tego problemu, które dzieli się na trzy grupy [49, 56]:

1. Podejścia działające na poziomie danych, nazywane zewnętrznymi (*ang. external approaches*) - obsługa danych niezrównoważonych odbywa się na poziomie przetwarzania danych, niezależnie od stosowanego algorytmu uczenia klasyfikatora.

2. Podejścia działające na poziomie algorytmu uczenia, nazywane wewnętrznymi (*ang. internal approaches*) - klasyczne algorytmy uczenia wzbogacane są o mechanizmy niwelujące negatywne skutki dysproporcji w danych.

3. Podejścia z uczeniem wrażliwym na koszt (*ang. cost-sensitive learning*) - techniki te stanowią kombinację zewnętrznego i wewnętrznego podejścia. Z jednej strony dane wejściowe modyfikowane są poprzez nadanie różnych wag (kosztów) poszczególnym obiektom, z drugiej strony algorytm uczenia wzbogacony jest o mechanizmy uwzględniające różne wagi nadane obserwacjom.

Zaprezentowany podział technik balansowania danych nie jest podziałem rozłącznym, gdyż niektóre z algorytmów zakładają jednoczesne wykorzystanie kilku technik.

2.2.1 Podejścia zewnętrzne

Zasadniczą cechą technik zewnętrznych jest fakt, że proces obsługi danych niezbalansowanych na etapie przetwarzania umożliwia stosowanie opisanych w tym rozdziale klasycznych algorytmów uczenia dedykowanych dla problemów zbalansowanych bez konieczności ich modyfikacji. Większość z technik wyodrębnianych w tej grupie wykorzystuje celem zbalansowania danych mechanizmy generowania nowych obserwacji (*ang. oversampling*) z klasy zdominowanej, bądź też techniki eliminacji obiektów (*ang. undersampling*) z klasy dominującej.

Podstawową metodą wykorzystującą technikę generowania nowych obiektów jest próbkowanie losowe (*ang. random oversampling*), które polega na duplikowaniu obserwacji z klasy zdominowanej poprzez ich losowanie ze zwracaniem z wejściowego zbioru uczącego. Analogicznie do próbkowania losowego wyróżnia się eliminację losową (*ang. random*

undersampling) obiektów z klasy dominującej. Metoda eliminacji losowej znajduje zastosowanie jedynie w przypadkach, w których usunięcie obserwacji nie spowoduje zmiany w rozkładzie klasy dominującej.

Celem zachowania rozkładu klasy dominującej stosuje się techniki eliminacji świadomej (*ang. informed undersampling*), polegające na inteligentnym wyborze obserwacji do usunięcia. Zestaw metod eliminacji świadomej wykorzystujących do wyboru obserwacji algorytm K-NN został opublikowany w pracy [86].

Proces próbkowania nowych obserwacji może również odbywać się w sposób *inteligentny*, poprzez generowanie nowych, syntetycznych obserwacji bazując na zdominowanych obserwacjach ze zbioru uczącego. Jedną z najpopularniejszych metod wykorzystujących próbkowanie syntetyczne jest algorytm *SMOTE* (*Synthetic Minority Over-sampling TEchnique*) [24]. Podejście to wykorzystuje algorytm K-NN w taki sposób, że syntetyczna obserwacja generowana jest na ścieżce łączącej dwóch sąsiadów z klasy zdominowanej. Główną wadą metody *SMOTE* jest to, że zakłada ona wygenerowanie nowych obserwacji dla każdego obiektu należącego do klasy zdominowanej co może prowadzić do zbudowania nadmiarowej liczby sztucznych obserwacji należących do tej klasy. Rozszerzeniem metody *SMOTE*, które eliminuje wspomniany problem, jest algorytm *Borderline-SMOTE* [64]. Metoda ta przeprowadza analizę wszystkich obserwacji z klasy zdominowanej i wybiera jedynie te, które znajdują się „blisko" płaszczyzny separującej klasy i mogą być błędnie zaklasyfikowane jako obiekty z klasy dominującej. Na wybranych obserwacjach następuje próbkowanie z wykorzystaniem klasycznego algorytmu *SMOTE*.

Inną grupę metod zewnętrznych stanowią algorytmy próbkowania z technikami czyszczenia danych (*ang. sampling with data clearing techniques*). Są to metody, w których wyodrębnia się dwa etapy: etap próbkowania, w którym wykorzystywane są podejścia bazujące na *SMOTE*, oraz etap usuwania obserwacji nadmiarowych. Jednym z typowych podejść związanych z czyszczeniem danych jest podejście wykorzystujące pojęcie wzajemnego sąsiedztwa obiektów należących do różnych klas, w literaturze nazywanym połączeniem *Tomek* (*ang. Tomek links*)[132]. Po wykonaniu etapu próbkowania ze zbioru uczącego usuwane są wszystkie obserwacje, które należą do połączenia *Tomek*. Ilościowa ocena jakości metody *SMOTE* z zastosowaniem czyszczenia danych metodą połączeń *Tomek* jest przedmiotem publikacji [8].

Inną grupę metod zewnętrznych stanowią podejścia próbkowania bazujące na grupowaniu (*ang. cluster-based sampling methods*). W ramach tej grupy metod w pierwszej kolejności następuje niezależne dla klasy dominującej i zdominowanej wyodrębnianie skupisk obiektów podobnych z wykorzystaniem klasycznych metod grupowania, a następnie wykonywane jest próbkowanie celem balansowania obiektów w ramach utworzonych skupisk elementów [66]. Cechą charakterystyczną tej grupy technik jest przeciwdziałanie negatywnym skutkom niezbalansowania nie tylko pomiędzy klasami (*ang. between-class imbalance*), ale również dysproporcjom występującym w ramach klas (*ang. within-class imbalance*) [56].

2.2.2 Podejścia wewnętrzne

Jedną z typowych grup reprezentujących rozwiązania wewnętrzne są techniki łączące klasyczne algorytmy konstrukcji zespołów klasyfikatorów z zastosowaniem metod próbkowania bądź eliminacji. Główną ideą tego typu rozwiązań jest konstrukcja klasyfikatorów bazowych na zbiorach danych poddanych modyfikacjom poprzez wykorzystanie różnych technik zewnętrznych. Typowym algorytmem wykorzystującym to podejście jest algorytm *SMOTEBoost* [25]. Metoda ta łączy ze sobą algorytm wzmacniania z próbkowaniem z wykorzystaniem *SMOTE*. W każdej iteracji konstrukcji klasyfikatora bazowego występuje proces uczenia ze zbioru danych poszerzonego o sztucznie wygenerowane próbki. Umożliwia to jednoczesne osiągnięcie dywersyfikacji klasyfikatorów bazowych i zbalansowania klasyfikatora łącznego. Rozszerzeniem algorytmu *SMOTEBoost* jest metoda *MSMOTEBoost* wykorzystująca zmodyfikowaną technikę próbkowania obserwacji syntetycznych opisaną w [62]. W ramach tej grupy technik wyróżnia się również algorytmy *RAMOBoost* [26], oraz *DataBoost-IM* [53], które w każdej iteracji *boostingu* generują syntetyczne obiekty wykorzystując te obserwacje ze zbioru uczącego, które były błędnie klasyfikowane przez zbudowane już klasyfikatory bazowe. Innym podejściem jest algorytm *RUSBoost* [113] w którym zbiory bazowe generowane są poprzez losową eliminację obiektów z klasy dominującej.

W literaturze [49] zaproponowano szereg metod eliminacji niezbalansowania wykorzystujących metodę *baggingu*, takich jak algorytmy wykorzystujące techniki generowania nowych obiektów (metoda *SMOTEBagging* [139]), oraz algorytmy wykorzystujące techniki eliminacji (metody *QuasiBagging* [23], *Asymetric Bagging* [127], *Roughly Balanced Bagging*

[58], *Partitioning* [21, 143], *Bagging Ensemble Variation* [80]).

Inną ciekawą grupę metod dedykowanych do rozwiązania problemu dysproporcji w zbiorze uczącym stanowią rozwiązania wykorzystujące obliczenia granularne (*ang. granual computing*) [122, 123, 124, 126]. Cechą charakterystyczną obilczeń granularnych jest zorientowana na wiedzę dekompozycja wyjściowego zagadnienia na mniejsze, dające się rozwiązać równolegle, problemy nazywane granulami informacyjnymi (*ang. information granules*). W literaturze istnieje wiele technik granulacji informacji wykorzystujących drzewa decyzyjne, zbiory rozmyte, algorytmy grupowania, czy też reguły asocjacyjne. W przypadku niezabalansowanych zagadnień decyzyjnych głównym celem dekompozycji jest zbudowanie granul informacyjnych o zbalansowanym charakterze.

Inne rozwiązania balansowania danych wykorzystują uczenie aktywne (*ang. active learning*) [38, 39], które pierwotnie stosowane było do iteracyjnego wyboru szczególnie istotnych z punktu widzenia zadania decyzyjnego obiektów ze zbioru uczącego celem ich zaetykietowania i wykorzystania w procesie uczenia. Głównym założeniem uzasadniającym zastosowanie uczenia aktywnego jest fakt, że dane wykorzystywane w procesie uczenia są znacznie bardziej zbalansowane w obszarze płaszczyzn separujących [38]. W procesie aktywnego uczenia wybierane są jedynie obserwacje najbardziej informacyjne, czyli te, które znajdują się w bliskim otoczeniu hiperpłaszczyzn oddzielających dwie klasy. W rezultacie klasyfikator uczony jest na zbalansowanym i zredukowanym do najistotniejszych obiektów zbiorze uczącym.

W przypadku dużego niezbalansowania wysoko wymiarowych danych stosuje się podejścia wykorzystujące uczenie jednoklasowe (*ang. one-class learning*), w tym szczególnie rozwiązania wykorzystujące jednoklasowy SVM (*ang. one-class SVM*) [85, 112, 146, 147].

2.2.3 Podejścia wrażliwe na koszt

Wśród podejść wrażliwych na koszt wyróżnia się szereg rozwiązań modyfikujących algorytmy wzmacniania. W większości przypadków modyfikacji podlega sposób aktualizacji wag w każdej iteracji generowania klasyfikatora bazowego (patrz Algorytm 2), poprzez uwzględnienie różnych kosztów wynikających z błędnej klasyfikacji obiektów z klasy dominującej i zdominowanej. W konsekwencji wyższe wartości wag zostaną przyporządkowane nie tylko obiektom błędnie klasyfikowanym, ale również tym należącym do klasy mniej

licznej. Do najpopularniejszych algorytmów należących do tej grupy zalicza się: *AdaCost* [42], *CSB1*, *CSB2* [130], *RareBoost* [67], *AdaC1*, *AdaC2*, *AdaC3* [119].

Ze względu na stosowane w procesie budowy drzew decyzyjnych mechanizmy ucinania, klasyfikatory tego typu są szczególnie wrażliwe na dysproporcje w danych [56]. Celem równoważenia decyzji podejmowanych z wykorzystaniem drzew decyzyjnych stosuje się specjalne techniki ucinania, takie jak ucinanie techniką Laplace'a (*ang. Laplace pruning technique*) [37], bądź też wrażliwe na koszt kryteria podziału przestrzeni cech [36].

Techniki wrażliwe na koszt stosowane są również w uczeniu sieci neuronowych. Wśród rozwiązań wyodrębnionych w ramach tej grupy wyróżnia się: podejście polegające na wrażliwej na koszt estymacji prawdopodobieństw na etapie klasyfikacji, rozwiązanie adaptujące zróżnicowany koszt w wyjściach sieci neuronowej, modyfikację parametru uczenia, czy też modyfikację minimalizowanej funkcji błędu [75].

Techniki wrażliwe na koszt są powszechnie stosowane również w przypadku klasyfikatorów typu SVM. Eliminacja problemu niezbalansowania odbywa się poprzez modyfikację kryterium uczenia, z wykorzystaniem różnych wartości parametru kosztu błędnej klasyfikacji dla klasy dominującej i zdominowanej [92, 136]. W pracy [137] wykorzystano dodatkowo procedurę wzmacniania wrażliwego na koszt klasyfikatora SVM.

Opisane w powyższym rozdziale metody przeciwdziałania negatywnym skutkom dysproporcji w zbiorze uczącym stanowią główne rozwiązania dostępne w literaturze lecz nie wyczerpują technik stosowanych do rozwiązania problemu. Dokładny przegląd metod balansowania danych w zadaniu klasyfikacji znajduje się w pozycji [56].

Rozdział 3

Złożone algorytmy SVM dla niezbalansowanych danych

W niniejszym rozdziale opisana została koncepcja wzmacnianych klasyfikatorów SVM dla danych niezbalansowanych. Zaproponowano autorską metodę uczenia tego typu klasyfikatora oraz zaproponowano dwie jego modyfikacje wykorzystujące algorytm eliminacji jednostronnej oraz uczenie aktywne.

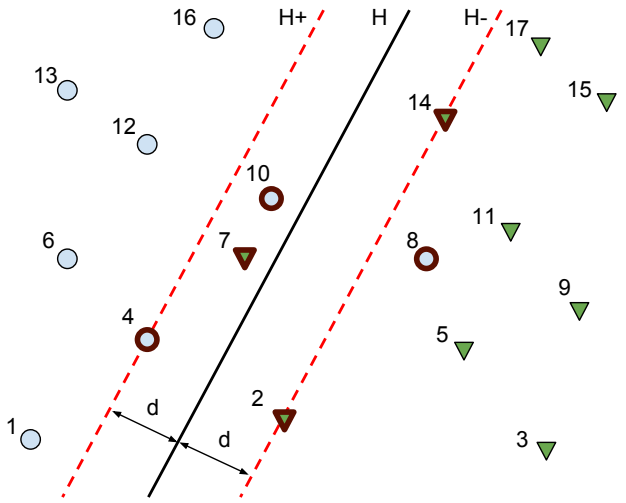
3.1 Zadanie uczenia klasyfikatora SVM dla niezbalansowanych danych

Zadanie uczenia liniowego klasyfikatora SVM, dla którego hiperpłaszczyzna klasyfikująca H (patrz Rysunek 1.1) zadana jest wzorem:

$$H : \mathbf{a}^T\mathbf{x} + b = 0, \tag{3.1}$$

sprowadza się maksymalizacji odległości d zadanej wzorem [1]:

[1] W rzeczywistości d przyjmuje postać $d = \frac{l}{\|\mathbf{a}\|}$, jednak bez straty ogólności można przyjąć $l = 1$ (przy jednoczesnym przeskalowaniu wag $\mathbf{a}$ oraz stałej b). Wówczas, hiperpłaszczyzny H_+, oraz H_- budujące margines klasyfikatora SVM (nazywane *hiperpłaszczyznami kanonicznymi*) opisane są kolejno równaniami $H+ : \mathbf{a}^T\mathbf{x} + b = +1$, oraz $H- : \mathbf{a}^T\mathbf{x} + b = -1$



Rysunek 3.1: Optymalna hiperpłaszczyzna otrzymana w wyniku wyuczenia klasyfikatora SVM na danych nieseparowalnych.

$$d = \frac{1}{\|\mathbf{a}\|}. \tag{3.2}$$

W rzeczywistości, uwzględniając nieseparowalność liniową klasyfikowanych danych problem uczenia definiuje się jako zadanie minimalizacji kryterium zadanego równaniem:

$$Q(\mathbf{a}) = \frac{1}{2}\mathbf{a}^T\mathbf{a} + C\sum_{n=1}^{N}\xi_n^i, \tag{3.3}$$

przy ograniczeniach:

$$y_n(\mathbf{a}^T\mathbf{x} + b) \geqslant 1 - \xi_n, \tag{3.4}$$

dla $n \in \{1, \ldots, N\}$. Wektor $\mathbf{a}$ oznacza zestaw parametrów hiperpłaszczyzny liniowego klasyfikatora SVM separującego dwie klasy (patrz Rysunek 1.1), C jest parametrem kosztu związanego z błędną klasyfikacją, natomiast ξ_n są zmiennymi [2], które przyjmują wartości dodatnie, gdy obserwacje znajdują się wewnątrz, bądź też po złej stronie marginesu wyznaczonego w procesie uczenia. Im wyższa jest wartość parametru C, tym wyższa jest kara za błędne zaklasyfikowanie danej obserwacji.

[2] W literaturze anglojęzycznej określa się je terminem *slack variables*, w literaturze polskiej mówi się o nich jako o zmiennych pomocniczych, bądź dodatkowych.

Uwzględnienie zmiennych ξ_n w zadaniu maksymalizacji odległości d umożliwia rozwiązanie zadania optymalizacji w przypadku, gdy obserwacje należące do dwóch klas nie są liniowo separowalne. Wówczas margines ograniczony hiperpłaszczyznami H_+, oraz H_- nazywany jest marginesem miękkim (*ang. soft margin*), gdyż dopuszcza on występowanie obserwacji błędnie klasyfikowanych. W zależności od położenia n-tej obserwacji względem marginesu wyróżnić można trzy przypadki:

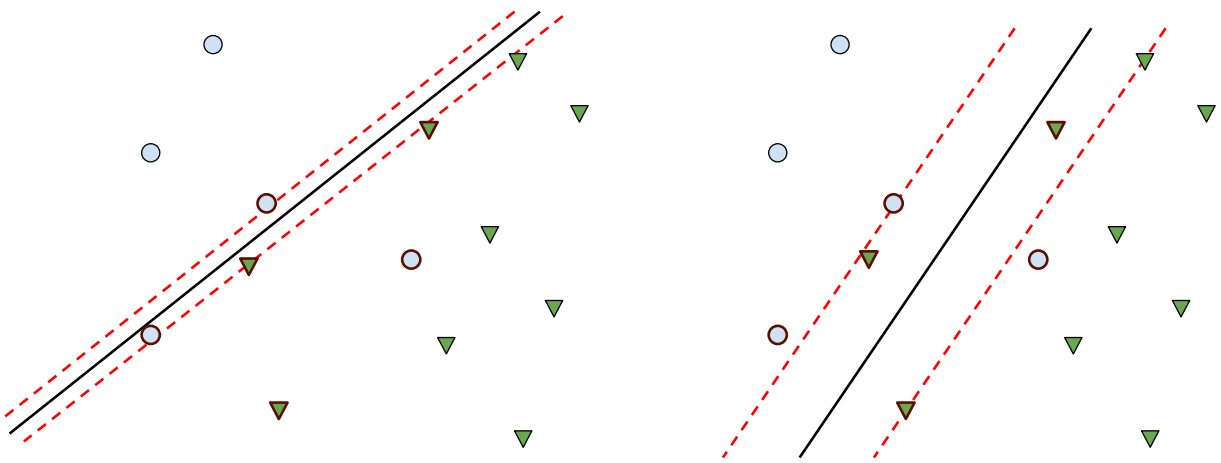
- Obserwacja jest poprawnie klasyfikowana i nie leży wewnątrz marginesu ograniczonego hiperpłaszczyznami $H+$, oraz $H-$, wówczas wartość zmiennej ξ_n wynosi 0 (obserwacje o indeksach $1, 4, 6, 12, 13, 16$ z pierwszej klasy, oraz obserwacje o indeksach z klasy drugiej $2, 3, 5, 9, 11, 14, 15, 17$ na Rysunku 3.1).
- Obserwacja jest poprawnie klasyfikowana, ale leży wewnątrz marginesu, wówczas $0 < \xi_n < 1$ (obserwacja o indeksie 10).
- Obserwacja nie jest poprawnie klasyfikowana, wówczas $\xi_n \geqslant 1$ (obserwacje o indeksach 7 i 8).

Obserwacje spełniające dwa ostatnie z wymienionych warunków, oraz obserwacje leżące na hiperpłaszczyznach H_+, oraz H_- (obserwacje o indeksach $2, 4$, oraz 14) nazywane są wektorami wspierającymi.

Zastosowanie kryterium (3.3) w zadaniu klasyfikacji z niezbalansowanymi danymi może prowadzić do zbytniego przesunięcia się marginesu SVM w kierunku klasy zdominowanej (Rysunek 3.2a). Celem balansowania algorytmu uczenia klasyfikatora SVM modyfikuje się kryterium optymalizacji do postaci zadanej wzorem (Rysunek 3.2b):

$$Q(\mathbf{a}) = \frac{1}{2}\mathbf{a}^T\mathbf{a} + C_+ \sum_{n_+ \in \mathbb{N}_+} \xi_{n_+}^i + C_- \sum_{n_- \in \mathbb{N}_-} \xi_{n_-}^i, \tag{3.5}$$

gdzie $\mathbb{N}_+ = \{n \in \{1, \ldots, N\} : y_n = +1\}$, oraz $\mathbb{N}_- = \{n \in \{1, \ldots, N\} : y_n = -1\}$. W literaturze stosuje się szereg podejść do rozwiązania problemu uczenia wrażliwego na koszt SVM, w których przyjmuje się zarówno normę $L_1(i = 1)$ dla zmiennych ξ_n [92], jak i normę $L_2(i = 2)$ [136]. O wartościach C_+, oraz C_- przyjmuje się dodatkowo, że powinny spełniać warunek:



(a) klasyczny SVM (b) SVM wrażliwy na koszt

Rysunek 3.2: Optymalna hiperpłaszczyzna otrzymana w wyniku wyuczenia klasyfikatora SVM na danych niezbalansowanych: a) poprzez minimalizację kryterium (3.3), b) poprzez minimalizację kryterium (3.5).

$$\frac{C_+}{C-} = \frac{N_-}{N_+}, \tag{3.6}$$

gdzie N_- oznacza liczność zbioru $\mathbb{N}_-$, natomiast N_+ liczność zbioru $\mathbb{N}_+$. Istotnym zagadnieniem w zadaniu konstrukcji SVM dla danych niezbalansowanych jest wybór normy dla zmiennych ξ_n. Wprowadzenie kryterium z normą L_2 zwiększa udział obserwacji błędnie klasyfikowanych ($\xi_n^2 \geqslant \xi_n$, dla $\xi_n \geqslant 1$), zmniejsza natomiast udział obserwacji znajdujących się wewnątrz marginesu ($\xi_n^2 < \xi_n$, dla $0 < \xi_n < 1$).

W odróżnieniu od rozwiązań opisanych w literaturze w niniejszej pracy rozważa się następujące kryterium optymalizacji algorytmu uczenia SVM dla danych niezbalansowanych:

$$Q(a) = \frac{1}{2}\mathbf{a}^T\mathbf{a} + C\sum_{n=1}^{N}\omega_n\xi_n. \tag{3.7}$$

Proponowane w pracy sformułowanie problemu uczenia klasyfikatora SVM dla danych niezbalansowanych sprowadza się minimalizacji kryterium (3.7) przy ograniczeniach (3.4). Wprowadzenie wag ω_n w sformułowaniu zadania optymalizacji daje możliwość dywersyfikacji kosztów błędnej klasyfikacji nie tylko pomiędzy klasami, ale również wewnątrz

danej klasy. Ponadto, uwzględnienie wag w zadaniu optymalizacji zwalania z konieczności wyboru normy dla zmiennych luźnych, gdyż sterowanie znaczeniem obserwacji błędnie klasyfikowanych odbywa się poprzez dobór odpowiednich wartości ω_n. Zmodyfikowane kryterium (3.7) stanowi również uogólnienie kryterium (3.5) dla normy L_1 przyjmując wartości wag spełniające warunki: $C_+ = C\omega_n$ dla $n \in \mathbb{N}_+$, oraz $C_- = C\omega_n$ dla $n \in \mathbb{N}_-$. Z drugiej strony sformułowanie problemu poprzez zadane kryterium umożliwia sterowanie generalizacją modelu poprzez wartość parametru C przy następującym założeniu że:

$$\sum_{n=1}^{N} \omega_n = N. \tag{3.8}$$

Sformułowanie problemu uczenia klasyfikatora SVM w postaci kryterium (3.7) daje też możliwość iteracyjnej aktualizacji wag w procesie konstrukcji klasyfikatorów bazowych algorytmów wzmacnianych. Proponowane sformułowanie problemu uczenia daje ponadto analogiczną do klasycznego problemu postać dualną zadania optymalizacji.

Celem rozwiązania przyjętego w pracy problemu uczenia dla kryterium (3.7) przy ograniczeniach (3.4) sformułowano zadanie dualne do rozpatrywanego problemu optymalizacji. W tym celu wyznaczono funkcję Lagrange'a:

$$L(\mathbf{a}, b, \xi, \boldsymbol{\lambda}, \boldsymbol{\gamma}) = \frac{1}{2}\mathbf{a}^T\mathbf{a} + C\sum_{n=1}^{N} \omega_n\xi_n - \sum_{n=1}^{N} \lambda_n(y_n(\mathbf{a}^T\mathbf{x}_n + b) - 1 + \xi_n) - \sum_{n=1}^{N} \gamma_n\xi_n, \tag{3.9}$$

gdzie $\boldsymbol{\lambda}$, $\boldsymbol{\gamma}$ oznaczają wektory mnożników Lagrange'a. W powyższym sformułowaniu problemu optymalizacji przyjmujemy, że wartości C oraz $\boldsymbol{\omega}$ są stałe. Wyznaczmy pochodne cząstkowe z funkcji Lagrange'a dla wszystkich parametrów będących przedmiotem zadania optymalizacji problemu prymalnego:

$$\frac{\partial L(\mathbf{a}, b, \xi, \boldsymbol{\lambda}, \boldsymbol{\gamma})}{\partial a_d} = a_d - \sum_{n=1}^{N} \lambda_n y_n x_n^{(d)}, \tag{3.10}$$

dla każdego $d \in \{1, \ldots, D\}$. Przyrównując pochodną pochodną cząstkową do 0 możemy wektorowo zapisać następujący układ równań:

$$\mathbf{a} = \sum_{n=1}^{N} \lambda_n y_n \mathbf{x}_n. \tag{3.11}$$

Postępując analogicznie z pozostałymi parametrami mamy:

$$\frac{\partial L(\mathbf{a}, b, \xi, \boldsymbol{\lambda}, \boldsymbol{\gamma})}{\partial b} = -\sum_{n=1}^{N} \lambda_n y_n. \tag{3.12}$$

Przyrównując pochodną do 0 mamy:

$$\sum_{n=1}^{N} \lambda_n y_n = 0. \tag{3.13}$$

Oraz dla wektora parametrów $\boldsymbol{\xi}$:

$$\frac{\partial L(\mathbf{a}, b, \xi, \boldsymbol{\lambda}, \boldsymbol{\gamma})}{\partial \xi_n} = C\omega_n - \lambda_n - \gamma_n, \tag{3.14}$$

$$C\boldsymbol{\omega} = \boldsymbol{\lambda} + \boldsymbol{\gamma}. \tag{3.15}$$

Wykorzystując warunki (3.11), (3.13), oraz (3.15) poprzez wstawienie do funkcji Lagrange'a (3.9) otrzymujemy funkcję celu dla zadania dualnego:

$$Q_D(\boldsymbol{\lambda}) = \sum_{n=1}^{N} \lambda_n - \frac{1}{2} \sum_{j=1}^{N} \sum_{i=1}^{N} \lambda_i \lambda_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j. \tag{3.16}$$

Ograniczenia dla dualnego zadania optymalizacji reprezentowane są przez równania (3.13), (3.15), oraz $\lambda_n, \gamma_n \geqslant 0$, dla każdego $n \in \{1, \ldots, N\}$. Wektor parametrów $\boldsymbol{\gamma}$ występuje jedynie w ograniczeniu (3.15), dlatego, wiedząc że $\gamma_n \geqslant 0$, warunek (3.15) zastępuje się warunkiem:

$$0 \leqslant \lambda_n \leqslant C\omega_n \tag{3.17}$$

Ostatecznie dualna forma zadania minimalizacji kryterium zadanego wzorem (3.7) przy ograniczeniach (3.4) sprowadza się do maksymalizacji kryterium (3.16) przy ograniczeniach (3.13), oraz (3.17). Wynikiem sformułowania problemu uczenia w postaci dualnej jest zwiększenie wymiaru parametrów z $D+1$ (wymiar wektora $\mathbf{a}$ poszerzonego o b) do N (wymiar wektora $\boldsymbol{\lambda}$). Zaletą formy dualnej jest to, że wyraża ona kryterium optymalizacji w terminach iloczynów skalarnych obserwacji $\mathbf{x}_n$, co jest istotne przy uogólnieniu klasyfikatora SVM na model nieliniowy [73]. Uogólniona postać dualna kryterium optymalizacji zadana jest wzorem:

$$Q_D(\boldsymbol{\lambda}) = \sum_{n=1}^{N} \lambda_n - \frac{1}{2} \sum_{j=1}^{N} \sum_{i=1}^{N} \lambda_i \lambda_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j), \tag{3.18}$$

gdzie $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ nazywana jest funkcją jądra [73]. Podejście polegające na zastąpieniu iloczynu skalarnego $\mathbf{x}_i \mathbf{x}_j$ funkcją jądra ma na celu przejście z przestrzeni $\mathbb{X}$, charakteryzującej się niskim wymiarem, na przestrzeń o wyższym (często nieskończonym) wymiarze.[3] Przejście to odbywa się poprzez zastosowanie nieliniowego przekształcenia $\phi(\cdot)$ i jego celem jest uzyskanie większej separowalnosci pomiędzy klasami. Dokładne uzasadnienie stosowalności funkcji jądra, wraz z przykładami znaleźć można w pozycjach [10, 73, 135].

Rozwiązanie zadania dualnego ma również jasną interpretację:

- Jeżeli $\lambda_n = 0$, wówczas korespondująca obserwacja $\mathbf{x}_n$ jest poprawnie klasyfikowana i leży poza marginesem wyznaczonym przez hiperpłaszczyzny H_+, oraz H_-.
- Jeżeli $0 < \lambda_n < C\omega_n$ wówczas obserwacja jest poprawnie klasyfikowana i leży na odpowiadającej klasie hiperpłaszczyźnie H_+ bądź H_-.
- Jeżeli $\lambda_n = C\omega_n$ wówczas obserwacja leży poza marginesem i jest błędnie klasyfikowana, bądź też leży wewnątrz marginesu.

Rozwiązanie zadania optymalizacji kwadratowego kryterium zadanego wzorem (3.18) przy ograniczeniach (3.13), oraz (3.17) jest rozwiązaniem optymalnym, wtedy, i tylko wtedy, gdy macierz o elementach $y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$ jest dodatnio pół-określona oraz dla każdego $n \in \{1, \ldots, N\}$ spełnione są następujące warunki *Kuhna-Tuckera*:

$$\begin{aligned} \lambda_n = 0 &\Rightarrow y_n y(\mathbf{x}_n) > 1, \\ 0 < \lambda_n < C\omega_n &\Rightarrow y_n y(\mathbf{x}_n) = 1, \\ \lambda_n = C\omega_n &\Rightarrow y_n y(\mathbf{x}_n) < 1, \end{aligned} \tag{3.19}$$

gdzie funkcję klasyfikującą dla uogólnionego na przypadek nieliniowy SVM definiuje się następująco:

$$y(\mathbf{x}_n) = \sum_{i=1}^{N} y_i \lambda_i K(\mathbf{x}_i, \mathbf{x}_n) + b. \tag{3.20}$$

[3] Procedurę przejścia do przestrzeni o wyższym wymiarze poprzez wykorzystanie funkcji jądra nazywa się *kernel trick*

W praktyce większość mnożników Lagrange'a przyjmuje wartość 0 co znacznie upraszcza procedurę uczenia. Zadanie uczenia klasycznego klasyfikatora SVM, dla którego problem uczenia zadany jest poprzez minimalizację kryterium (3.3) przy ograniczeniach (3.4), sprowadza się do rozwiązania zadania optymalizacji wypukłej z ograniczeniami liniowymi [46], jednak ze względu na wysokie wymagania obliczeniowe i pamięciowe tradycyjnych technik stosowanych do rozwiązania tego problemu w praktyce stosuje się rozwiązania polegające na dekompozycji wyjściowego zadania optymalizacji na zadania mniejsze [10, 73]. Do najpowszechniej stosowanych tego typu technik uczenia klasyfikatorów SVM zalicza się zaproponowany przez Vapnika *„chunking"* [135], algorytm dekompozycji Osuny (*ang. Osuna's algorithm*) [97], oraz zaproponowany w pracy [98] optymalizacji minimalnej (*ang. Sequential Minimal Optimization, SMO*), który, po wprowadzeniu zwiększających efektywność modyfikacji opisanych w pozycji [68], stanowi jedną z najpopularniejszych metod uczenia SVM.

3.2 Algorytm SMO dla przyjętego kryterium uczenia

W poprzedniej sekcji zdefiniowane zostało kryterium uczenia klasyfikatora SVM dla danych niezbalansowanych postaci (3.7) przy ograniczeniach (3.4). Wyprowadzona została postać dualna proponowanego w pracy kryterium postaci (3.16) przy ograniczeniach (3.13), oraz (3.17). Celem rozwiązania zadanego w postaci dualnej problemu uczenia zaproponowano modyfikację klasycznego algorytmu SMO, która została opisana w tej sekcji.

Cechą charakterystyczną algorytmu SMO jest to, że w każdej iteracji sekwencyjnego procesu optymalizacji wykorzystuje się jedynie dwa mnożniki Lagrange'a. Dzięki takiej redukcji zadania optymalizacyjnego w każdym kroku rozpatrywane jest najmniejsze z możliwych zadań (optymalizacja jedynie wektora dwóch zmiennych), przez co rozwiązanie ma prostą formę analityczną i nie wymaga przydziału dużych zasobów pamięciowych.

Załóżmy, że wybrane do optymalizacji mnożniki Lagrange'a oznaczone są kolejnymi liczbami naturalnymi, tj. jako λ_1, oraz λ_2. Rozważamy zadanie optymalizacji funkcji kwadratowej dwóch zmiennych przy zadanych ograniczeniach:

$$0 \leqslant \lambda_i \leqslant C\omega_i, i = 1, 2, \tag{3.21}$$

oraz:

$$y_1\lambda_1 + y_2\lambda_2 = const. \tag{3.22}$$

Drugie z ograniczeń jest konsekwencją przyjęcia stałych wartości pozostałych mnożników w ograniczeniu (3.11). Warunki (3.21) wymuszają położenie mnożników w prostokącie o wymiarach $C\omega_1$ na $C\omega_2$. Jednocześnie rozwiązania λ_1 i λ_2 muszą leżeć na prostej określonej równaniem $y_1\lambda_1 + y_2\lambda_2 = const$. Kierunek nachylenia prostej determinowany jest przez wartości etykiet klas y_i. Proces optymalizacji sprowadza się do wykonania dwóch kroków obliczeniowych (szczegółowy opis wyprowadzenia rozwiązania analitycznego opisany został w pozycji [98]). W pierwszym kroku aktualizowana jest wartość mnożnika λ_2 z wykorzystaniem procedury:

$$\lambda_{2,new} = \lambda_{2,old} - \frac{y_2(E_1^{(SMO)} - E_2^{(SMO)})}{\eta}, \tag{3.23}$$

gdzie $\lambda_{2,old}$ oznacza wartość mnożnika z poprzedniego kroku algorytmu optymalizacji, natomiast η, oraz $E_i^{(SMO)}$ definiuje się kolejno:

$$\eta = 2K(\mathbf{x}_1, \mathbf{x}_2) - K(\mathbf{x}_1, \mathbf{x}_1) - K(\mathbf{x}_2, \mathbf{x}_2), \tag{3.24}$$

oraz:

$$E_i^{(SMO)} = y_{old}(\mathbf{x}_i) - y_i, \tag{3.25}$$

gdzie $y_{old}(\mathbf{x}_i)$ stanowi aktualną wartość funkcji klasyfikacyjnej:

$$y_{old}(\mathbf{x}_i) = \sum_{n=1}^{N} y_n\lambda_{n,old}K(\mathbf{x}_n, \mathbf{x}_i) + b, \tag{3.26}$$

która dla przypadku liniowego przyjmuje postać:

$$y_{old}(\mathbf{x}_i) = \sum_{n=1}^{N} y_n\lambda_{n,old}\mathbf{x}_n^{\mathrm{T}}\mathbf{x}_i + b, \tag{3.27}$$

gdzie $\sum_{n=1}^{N} y_n\lambda_{n,old}\mathbf{x}_n$ jest estymatorem wektora parametrów $\mathbf{a}$ hiperpłaszczyzny separującej (3.1). W drugim kroku wykonywana jest korekta wyznaczonego mnożnika $\lambda_{2,new}$ tak, by spełniał ograniczenie (3.21):

$$\lambda_{2,opt,new} = \begin{cases} U & \text{dla } \lambda_{2,new} \geqslant U \\ \lambda_{2,new} & \text{dla } L < \lambda_{2,new} < U \\ L & \text{dla } \lambda_{2,new} \leqslant L. \end{cases} \tag{3.28}$$

Wartości L oraz U definiuje się następująco:

$$L = \begin{cases} \max\{0, (\lambda_{2,old} - \lambda_{1,old})\} & \text{dla } y_1 \neq y_2 \\ \max\{0, (\lambda_{2,old} + \lambda_{1,old} - C\omega_1)\} & \text{dla } y_1 = y_2, \end{cases} \tag{3.29}$$

oraz:

$$U = \begin{cases} \min\{C\omega_2, (C\omega_1 + \lambda_{2,old} - \lambda_{1,old})\} & \text{dla } y_1 \neq y_2 \\ \min\{C\omega_2, (\lambda_{2,old} + \lambda_{1,old})\} & \text{dla } y_1 = y_2, \end{cases} \tag{3.30}$$

Zasadniczą różnicą w stosunku do klasycznego algorytmu uczenia SMO jest uwzględnienie różnych kosztów $C\omega_1$ oraz $C\omega_2$ kojarzonych z obserwacjami $\mathbf{x}_1$ oraz $\mathbf{x}_2$. Wartość drugiego mnożnika wyznacza się korzystając z następującej postaci analitycznej:

$$\lambda_{1,opt,new} = \lambda_{1,new} + y_1 y_2 (\lambda_{2,old} - \lambda_{2,opt,new}). \tag{3.31}$$

Zależności (3.28), oraz (3.31) wyznaczają optymalne wartości mnożników Lagrange'a dla rozpatrywanego kroku sekwencyjnej optymalizacji. Efektywność algorytmu SMO osiągnięta została poprzez zaproponowaną w pracy [98] heurystykę wyboru dwóch mnożników Lagrange'a gwarantującą zmniejszenie wartości funkcji celu w każdym kroku optymalizacji. W ramach metody wyróżnia się dwie pętle przeszukujące dane, każdą odpowiedzialną za wybór jednego z mnożników: pętlę zewnętrzną oraz wewnętrzną.

W ramach pierwszej pętli przeszukiwany jest cały zbiór danych w poszukiwaniu obserwacji, która nie spełnia warunków *Kuhna-Tuckera* zadanych implikacjami (3.19). Jeżeli wykryto obserwację $\mathbf{x}_1$, która nie spełnia zadanych warunków to korespondujący z nią mnożnik λ_1 wybrany zostaje do procesu optymalizacji i automatycznie uruchamiana jest pętla wewnętrzna celem znalezienia drugiego z mnożników. Po jednym przejściu pętli zewnętrznej po całym zbiorze uczącym uruchamiana jest ona ponownie, jednak celem przyspieszenia działania algorytmu w dalszych przebiegach pętli rozważane są jedynie te obserwacje, dla których $0 < \lambda_j < C\omega_j$. Proces wyboru w pętli zewnętrznej wykonywany

jest aż do momentu, w którym, z zadaną dokładnością, każda z obserwacji będzie spełniać warunki *Kuhna-Tuckera*.

Po wybraniu w pętli zewnętrznej mnożnika λ_1 uruchamiana jest pętla wewnętrzna, w której dokonywana jest selekcja mnożnika λ_2. Pętla wewnętrzna przeszukuje wszystkie przypadki dla których $0 < \lambda_j < C\omega_j$ sprawdzając $|E_2^{(SMO)} - E_1^{(SMO)}|$ i wybiera ten mnożnik, dla którego moduł różnicy przyjmuje najwyższą wartość. Na wyselekcjonowanych przypadkach następuje optymalizacja mnożników zgodnie z procedurami (3.28) i (3.31).

W niniejszej pracy rozważamy zmodyfikowany algorytm SMO, jako algorytm uczenia SVM, który w rozprawie oznaczony będzie $SMO_{Imbl}(\mathbb{S}_N, \boldsymbol{\omega})$, gdzie $\boldsymbol{\omega}$ jest wektorem wag definiujących ograniczenia (3.13). W opisanych w tym rozdziale algorytmach możliwe jest jednak wykorzystanie innych metod uczenia klasyfikatorów SVM, które uwzględniają różne wartości ograniczeń $C\omega_n$.

3.3 Wyznaczanie wartości wag klasyfikatora SVM dla problemu niezbalansowania

Kluczowym elementem proponowanego klasyfikatora SVM dla danych niezbalansowanych jest wyznaczenie wektora wag $\boldsymbol{\omega}$ w taki sposób, aby wyuczony model osiągnął najwyższą jakość predykcji dla danych z nierównym rozkładem klas. Próbą rozwiązania tego problemu jest przyjęcie następujących wartości wag:

$$\omega_n = \begin{cases} \frac{N}{2N_+} & \text{dla } n \in \mathbb{N}_+ \\ \frac{N}{2N_-} & \text{dla } n \in \mathbb{N}_- \end{cases}, \tag{3.32}$$

gdzie wartości N_+, oraz N_- stanowią kolejno liczności zbiorów $\mathbb{N}_+$, oraz $\mathbb{N}_-$. Przyjęte wartości wag spełniają wówczas warunek:

$$\sum_{n \in \mathbb{N}_+} \omega_n = \sum_{n \in \mathbb{N}_-} \omega_n = \frac{N}{2}, \tag{3.33}$$

oraz warunek (3.8). Sumaryczny koszt błędnej klasyfikacji dla obserwacji zdominowanych jest więc równy sumarycznemu kosztowi błędnej klasyfikacji obiektów z klasy dominującej.

Zdefiniowanie wag w następujący sposób stanowi próbę rozwiązania problemu niezbalansowania pomiędzy klasami, jednak nie różnicuje ich w obrębie klas. Konieczne jest więc opracowanie metody dywersyfikującej wartości wag również w ramach klasy. W niniejszej pracy zaproponowano metody konstrukcji zespołów klasyfikatorów typu SVM, które rozwiązują wyżej postawiony problem poprzez nadanie wag obserwacjom w zależności od trudności ich sklasyfikowania przez dotychczas skonstruowane klasyfikatory bazowe.

3.4 Wzmacniany klasyfikator SVM dla niezbalansowanych danych

Algorytmy wzmacniania klasyfikatorów stosowane są na ogół celem podniesienia skuteczności klasyfikatorów słabych. W niniejszej rozprawie proponowane są zespoły klasyfikatorów typu SVM, które zaliczane są do klasyfikatorów silnych (ang. *strong learners*). W tym przypadku zastosowanie technik wzmacniania znajduje uzasadnienie w fakcie, iż celem ich zastosowania nie jest bezpośrednio poprawa skuteczności klasyfikacji, jednak niwelowanie negatywnych skutków jednoczesnego niebalansowania danych w ramach klasy, jak również pomiędzy klasami. W ramach podrozdziału scharakteryzowana została metoda konstrukcji wzmacnianego klasyfikatora SVM dedykowanego dla problemów klasyfikacji niezbalansowanej.

W pierwszej kolejności zaproponowano algorytm uczenia klasyfikatora wzmacnianego na danych niezbalansowanych *Boosting-IB* działający niezależnie od wyboru klasyfikatora bazowego. W ramach rozprawy, jako model bazowy dla klasyfikatora wzmacnianego, sugeruje się wykorzystanie metod typu SVM. W ogólności możliwe jest alternatywne zastosowanie innych klasyfikatorów bazowych, które minimalizują ważony błąd E_{Imb} postaci:

$$E_{Imb} = \frac{1}{N_+} \sum_{n \in \mathbb{N}_+} w_n^{(k)} I(\Psi_k^{(1)}(\mathbf{x}_n) \neq y_n) + \frac{1}{N_-} \sum_{n \in \mathbb{N}_-} w_n^{(k)} I(\Psi_k^{(1)}(\mathbf{x}_n) \neq y_n), \tag{3.34}$$

gdzie $w_n^{(k)}$ oznacza wagę n-tej obserwacji w k-tej iteracji pętli wzmacniania.

Procedurę uczenia wzmacnianego klasyfikatora dedykowanego do problemów niezbalansowanych opisuje Algorytm 3. W każdej iteracji algorytmu konstruowany jest klasyfikator bazowy w procesie minimalizacji funkcji błędu zadanego równaniem (3.34). W kolej-

Algorithm 3: Algorytm *Boosting-IB*

Input : Zbiór uczący $\mathbb{S}_N$, zbiór możliwych klas $\mathbb{Y}$

Output: Klasyfikator Boosting-IB: $\Psi(\mathbf{x}) = \arg\max_{y\in\mathbb{Y}} \sum_{k=1}^{K_{final}} c_k I(\Psi_k^{(1)}(\mathbf{x}) = y)$

Zadaj wartości początkowe wag $w_n^{(1)} = 1$ dla $n \in \{1, \ldots, N\}$;
$G \longleftarrow 0$;
$K_{final} \longleftarrow 1$;
for $k = 1 \rightarrow K$ **do**
    Wyucz klasyfikator bazowy $\Psi_k^{(1)}$ minimalizując funkcję błędu postaci (3.34);
    Wyznacz błąd postaci e_k korzystając z równania (3.35);
    if $e_k < 0.5$ **then**
        $c_k \longleftarrow \ln \frac{1-e_k}{e_k}$;
        Wyznacz wartość wskaźnika średniej geometrycznej G_k na danych $\mathbb{S}_N$ wykorzystując klasyfikator $\arg\max_{y\in\mathbb{Y}} \sum_{l=1}^{k} c_l I(\Psi_l^{(1)}(\mathbf{x}) = y)$;
        if $G_k > G$ **then**
            $K_{final} \longleftarrow k$;
            $G \longleftarrow G_k$;
        end
        Aktualizuj wagi: $w_n^{(k+1)} \longleftarrow w_n^{(k)} \exp\{c_k I(\Psi_k^{(1)}(\mathbf{x}_n) \neq y_n)\}$;
        Normalizuj wagi: $w_n^{(k+1)} \longleftarrow N \frac{w_n^{(k+1)}}{\sum_{n=1}^{N} w_n^{(k+1)}}$;
    else
        $c_k \longleftarrow 0$;
        Zadaj początkowe wartości dla wag $w_n^{(k+1)}$;
    end
end

nym kroku wyznaczania jest waga c_k reprezentująca udział danego klasyfikatora bazowego w klasyfikatorze złożonym. Wartość c_k zależy od znormalizowanej funkcji błędu e_k definiowanej następująco:

$$e_k = \frac{E_{Imb}}{\frac{1}{N_+}\sum\limits_{n\in\mathbb{N}_+} w_n^{(k)} + \frac{1}{N_-}\sum\limits_{n\in\mathbb{N}_-} w_n^{(k)}}. \tag{3.35}$$

Następnie na całym zbiorze $\mathbb{S}_N$ sprawdzana jest wartość kryterium $GMean$ dla wzmacnianego klasyfikatora składającego się z k klasyfikatorów bazowych. Jeżeli wartość rozpatrywanego wskaźnika po uwzględnieniu kolejnego (k-tego) klasyfikatora bazowego jest wyższa niż rozpatrywanego do tej pory jako najlepszego w sensie obranego kryterium (składającego się K_{final} klasyfikatorów bazowych) klasyfikatora wówczas aktualny klasyfikator złożony staje się najlepszym ($K_{final} = k$). Tego typu proces selekcji klasyfikatorów bazowych minimalizujący wskaźnik średniej geometrycznej został zaproponowany w pracy [137]. W ostatnim kroku wagi są aktualizowane w sposób analogiczny jak w klasycznych algorytmach wzmacniania:

$$w_n^{(k+1)} = w_n^{(k)} \exp\left\{c_k I(\Psi_k^{(1)}(\mathbf{x}_n) \neq y_n)\right\}. \tag{3.36}$$

Cechą charakterystyczną proponowanego w rozprawie algorytmu *Boosting-IB* jest to, że minimalizuje on wykładniczą funkcję błędu $E_{\exp,Imb}$ postaci:

$$E_{\exp,Imb} = \frac{1}{N_+}\sum_{n\in\mathbb{N}_+} \exp\left\{-y_n g_k(\mathbf{x}_n)\right\} + \frac{1}{N_-}\sum_{n\in\mathbb{N}_-} \exp\left\{-y_n g_k(\mathbf{x}_n)\right\}. \tag{3.37}$$

Dowód własności algorytmu *Boosting-IB* dotyczący minimalizacji funkcji wykładniczej funkcji błędu (3.37) przebiega następująco. Proces uczenia algorytmu *Boosting-IB* sprowadza się do wyznaczenia wartości współczynników $c_1, \ldots, c_k$, oraz klasyfikatorów $\Psi_1^{(1)}, \ldots, \Psi_k^{(1)}$ minimalizujących rozpatrywaną funkcję błędu. Proces minimalizacji przebiega sekwencyjnie, dlatego zakłada się ustalone wartości parametrów $c_1, \ldots, c_{k-1}$, oraz parametrów klasyfikatorów $\Psi_1^{(1)}, \ldots, \Psi_{k-1}^{(1)}$, a optymalizacji dokonuje się względem parametru c_k, oraz klasyfikatora $\Psi_k^{(1)}$. Wykładniczą funkcję błędu można rozpisać w następujący sposób:

$$
\begin{aligned}
E_{\exp,Imb} =& \frac{1}{N_+} \sum_{n\in\mathbb{N}_+} \exp\{-y_n g_{k-1}(\mathbf{x}_n) - \frac{1}{2} y_n c_k \Psi_k^{(1)}(\mathbf{x}_n)\} + \\
&+ \frac{1}{N_-} \sum_{n\in\mathbb{N}_-} \exp\{-y_n g_{k-1}(\mathbf{x}_n) - \frac{1}{2} y_n c_k \Psi_k^{(1)}(\mathbf{x}_n)\} \\
=& \frac{1}{N_+} \sum_{n\in\mathbb{N}_+} w_n^{(k)} \exp\{-\frac{1}{2} y_n c_k \Psi_k^{(1)}(\mathbf{x}_n)\} + \\
&+ \frac{1}{N_-} \sum_{n\in\mathbb{N}_-} w_n^{(k)} \exp\{-\frac{1}{2} y_n c_k \Psi_k^{(1)}(\mathbf{x}_n)\},
\end{aligned} \tag{3.38}
$$

gdzie przyjmuje stałą wartość $w_n^{(k)} = \exp\{-y_n g_{k-1}(\mathbf{x}_n)\}$, gdyż optymalizacja przebiega jedynie względem c_k, oraz $\Psi_k^{(1)}$. Załóżmy, że zbiór $\mathbb{T}_+$ oznacza zbiór indeksów obiektów należących do klasy pozytywnej klasyfikowanych poprawnie przez klasyfikator $\Psi_k^{(1)}(\mathbf{x})$, natomiast zbiór $\mathbb{F}_+$ zbiór indeksów pozostałych przykładów z klasy pozytywnej. Analogiczne oznaczenia, $\mathbb{T}_-$, oraz $\mathbb{F}_-$, przyjmijmy dla obiektów poprawnie i błędnie klasyfikowanych należących do klasy negatywnej. Wówczas funkcja błędu może być przedstawiona w następujący sposób:

$$
\begin{aligned}
E_{\exp,Imb} =& \frac{1}{N_+} \exp\{\frac{-c_k}{2}\} \sum_{n\in\mathbb{T}_+} w_n^{(k)} + \frac{1}{N_+} \exp\{\frac{c_k}{2}\} \sum_{n\in\mathbb{F}_+} w_n^{(k)} \\
&+ \frac{1}{N_-} \exp\{\frac{-c_k}{2}\} \sum_{n\in\mathbb{T}_-} w_n^{(k)} + \frac{1}{N_-} \exp\{\frac{c_k}{2}\} \sum_{n\in\mathbb{F}_-} w_n^{(k)}.
\end{aligned} \tag{3.39}
$$

Minimalizując funkcję błędu względem c_k otrzymujemy:

$$
\exp\{c_k^*\} = \frac{\frac{1}{N_+} \sum_{n\in\mathbb{T}_+} w_n^{(k)} + \frac{1}{N_-} \sum_{n\in\mathbb{T}_-} w_n^{(k)}}{\frac{1}{N_+} \sum_{n\in\mathbb{F}_+} w_n^{(k)} + \frac{1}{N_-} \sum_{n\in\mathbb{F}_-} w_n^{(k)}}. \tag{3.40}
$$

Korzystając z faktu:

$$
\begin{aligned}
\frac{1}{N_+} \sum_{n\in\mathbb{T}_+} w_n^{(k)} + \frac{1}{N_-} \sum_{n\in\mathbb{T}_-} w_n^{(k)} =& \frac{1}{N_+} (\sum_{n\in\mathbb{N}_+} w_n^{(k)} - \sum_{n\in\mathbb{F}_+,} w_n^{(k)}) \\
&+ \frac{1}{N_-} (\sum_{n\in\mathbb{N}_-} w_n^{(k)} - \sum_{n\in\mathbb{F}_-,} w_n^{(k)}),
\end{aligned} \tag{3.41}
$$

oraz wstawiając e_k zadane wzorem (3.35) do formuły (3.40) otrzymujemy:

$$\exp\{c_k^*\} = \frac{1-e_k}{e_k}. \tag{3.42}$$

Po obustronnym zlogarytmowaniu równania otrzymujemy metodę przydziału wartości wag poszczególnym klasyfikatorom bazowym dla algorytmu uczenia złożonego klasyfikatora (krok 8, Algorytm 3). Zapisując wykładniczą funkcję błędu w następujący sposób mamy:

$$\begin{aligned} E_{\exp,Imb} =& (\exp\{\frac{-c_k}{2}\} + \exp\{\frac{c_k}{2}\})(\frac{1}{N_+}\sum_{n\in\mathbb{N}_+} w_n^{(k)} I(\Psi_k^{(1)}(\mathbf{x}_n) \neq y_n) \\ &+ \frac{1}{N_-}\sum_{n\in\mathbb{N}_-} w_n^{(k)} I(\Psi_k^{(1)}(\mathbf{x}_n) \neq y_n)) \\ &+ \exp\{\frac{-c_k}{2}\}(\frac{1}{N_+}\sum_{n\in\mathbb{N}_+} w_n^{(k)} + \frac{1}{N_-}\sum_{n\in\mathbb{N}_-} w_n^{(k)}). \end{aligned} \tag{3.43}$$

Zauważyć można, że minimalizacja funkcji wykładniczego błędu względem klasyfikatora $\Psi_k^{(1)}$ sprowadza się do minimalizacji funkcji błędu zadanej wzorem (3.34), gdyż drugi składnik sumy jest stały, natomiast wartość $(\exp\{\frac{-c_k}{2}\}+\exp\{\frac{c_k}{2}\})$ nie ma wpływu na położenie minimum. Poszukiwany klasyfikator bazowy $\Psi_k^{(1),*}$ jest wyznaczany w procesie uczenia polegającego na minimalizacji funkcji błędu (3.34). Korzystając z (3.37), oraz wstawiając otrzymane wartości $\Psi_k^{(1),*}$, oraz c_k^* procedurę aktualizacji wag można zapisać w następujący sposób:

$$w_n^{(k+1)} = w_n^{(k)} \exp\{-\frac{1}{2} y_n c_k^* \Psi_k^{(1),*}(\mathbf{x}_n)\}. \tag{3.44}$$

Wykorzystując następujący fakt:

$$y_n \Psi_k^{(1),*}(\mathbf{x}_n) = 1 - 2I(\Psi_k^{(1),*}(\mathbf{x}_n) \neq y_n). \tag{3.45}$$

Procedurę aktualizacji wag zapisać można następująco:

$$w_n^{(k+1)} = w_n^{(k)} \exp\{-\frac{1}{2} c_k^*\} \exp\{c_k^* I(\Psi_k^{(1),*}(\mathbf{x}_n) \neq y_n)\}. \tag{3.46}$$

Składnik $\exp\{-\frac{1}{2}c_k^*\}$ nie zależy od n i może być pominięty. Proces aktualizacji wag wykonywany jest więc zgodnie z równaniem zadanym wzorem (3.36), co kończy dowód twierdzenia dotyczącego sekwencyjnej optymalizacji błędu wykładniczego w procesie konstrukcji

klasyfikatora *Boosting-IB*. Równocześnie, algorytm uczenia maksymalizuje wartość współczynnika średniej geometrycznej $GMean$, co wynika bezpośrednio z warunku realizowanego w krokach 10-13 Algorytmu 3.

Kluczowym elementem w konstrukcji wzmacnianego klasyfikatora *Boosting-IB* jest procedura uczenia klasyfikatorów bazowych polegająca na minimalizacji ważonej funkcji błędu zadanej wzorem (3.34). Sformułowanie zadania uczenia klasyfikatora *Boosting-IB* przedstawione w Algorytmie 3. jest niezależne od wyboru modelu klasyfikatora bazowego, dlatego możliwe jest stosowanie dowolnej, wrażliwej na koszt, metody konstrukcji klasyfikatorów $\Psi_k^{(1)}$, dla której stosunek wartości kosztów błędnej klasyfikacji dla klasy pozytywnej i negatywnej spełnia zależność $\frac{N_-}{N_+}$. Dodatkowo, aktualizowane wartości wag mogą być uwzględnione w samym procesie uczenia klasyfikatora bazowego, bądź też znormalizowane do 1 i wykorzystane do próbkowania danych, w sposób analogiczny jak w klasycznych algorytmach wzmacniania.

W niniejszej pracy jako klasyfikator bazowy metody *Boosting-IB* wykorzystuje się opisany w tym rozdziale model *SVM* dla danych niezbalansowanych. Klasyfikator bazowy jest więc uczony poprzez maksymalizację kryterium (3.18) przy ograniczeniach (3.13), oraz (3.17) z odpowiednio dobranym wektorem wag $\boldsymbol{\omega}^{(k)}$, gdzie k oznacza indeks klasyfikatora składowego. Sformułowany problem uczenia może być rozwiązany z wykorzystaniem opisanego w rozdziale algorytmu $SMO_{Imbl}(\mathbb{S}_N, \boldsymbol{\omega}^{(k)})$, bądź też inną metodą optymalizacji.

Algorytm 4 przedstawia proces uczenia wzmacnianego klasyfikatora SVM dla danych niezbalansowanych. Krok 5 Algorytmu 3 dla ogólnej metody uczenia został zastąpiony krokami 5 i 6 Algorytmu 4 w ramach których następuje przypisanie wartości wag $\boldsymbol{\omega}^{(k)}$ zgodnie z procedurą:

$$\omega_n^{(k)} = \begin{cases} \frac{N}{2N_+} w_n^{(k)} & \text{dla } n \in \mathbb{N}_+ \\ \frac{N}{2N_-} w_n^{(k)} & \text{dla } n \in \mathbb{N}_- \end{cases} \tag{3.47}$$

i następne wykorzystanie ich w algorytmie $SMO_{Imbl}(\mathbb{S}_N, \boldsymbol{\omega}^{(k)})$ do konstrukcji klasyfikatora bazowego $\Psi_k^{(1)}$. Dodatkowo, w przypadku gdy znormalizowana wartość błędu e_k jest wyższa, bądź równa 0.5 następuje przeskalowanie parametru C bazowego klasyfikatora SVM zgodnie z procedurą opisaną w korku 20.. Parametr $\alpha \in [0, 1]$ nazywany jest współczynnikiem generalizacji i określa o ile procent zmniejszyć wartość parametru C. Mniejsza

Algorithm 4: Algorytm *BoostingSVM-IB*

Input : Zbiór uczący $\mathbb{S}_N$, zbiór możliwych klas $\mathbb{Y}$

Output: Klasyfikator Boosting-IB: $\Psi(\mathbf{x}) = \arg\max_{y\in\mathbb{Y}} \sum_{k=1}^{K_{final}} c_k I(\Psi_k^{(1)}(\mathbf{x}) = y)$

Zadaj wartości początkowe wag $w_n^{(1)} = 1$ dla $n \in \{1, \ldots, N\}$;
$G \longleftarrow 0$;
$K_{final} \longleftarrow 1$;
for $k = 1 \rightarrow K$ **do**
  Wyznacz wartości wektora $\boldsymbol{\omega}^{(k)}$ korzystając z równania 3.47 ;
  $\Psi_k^{(1)} \longleftarrow SMO_{Imbl}(\mathbb{S}_N, \boldsymbol{\omega}^{(k)})$;
  Wyznacz błąd postaci e_k korzystając z równania 3.35;
  if $e_k < 0.5$ **then**
    $c_k \longleftarrow \ln \frac{1-e_k}{e_k}$;
    Wyznacz wartość wskaźnika średniej geometrycznej G_k na danych $\mathbb{S}_N$ wykorzystując klasyfikator $\arg\max_{y\in\mathbb{Y}} \sum_{l=1}^{k} c_l I(\Psi_l^{(1)}(\mathbf{x}) = y)$;
    if $G_k > G$ **then**
      $K_{final} \longleftarrow k$;
      $G \longleftarrow G_k$;
    end
    Aktualizuj wagi: $w_n^{(k+1)} \longleftarrow w_n^{(k)} \exp\{c_k I(\Psi_k^{(1)}(\mathbf{x}_n) \neq y_n)\}$;
    Normalizuj wagi: $w_n^{(k+1)} \longleftarrow N \frac{w_n^{(k+1)}}{\sum_{n=1}^{N} w_n^{(k+1)}}$;
  else
    $c_k \longleftarrow 0$;
    Zadaj początkowe wartości dla wag $w_n^{(k+1)}$;
    $C \longleftarrow (1-\alpha)\cdot C$;
  end
end

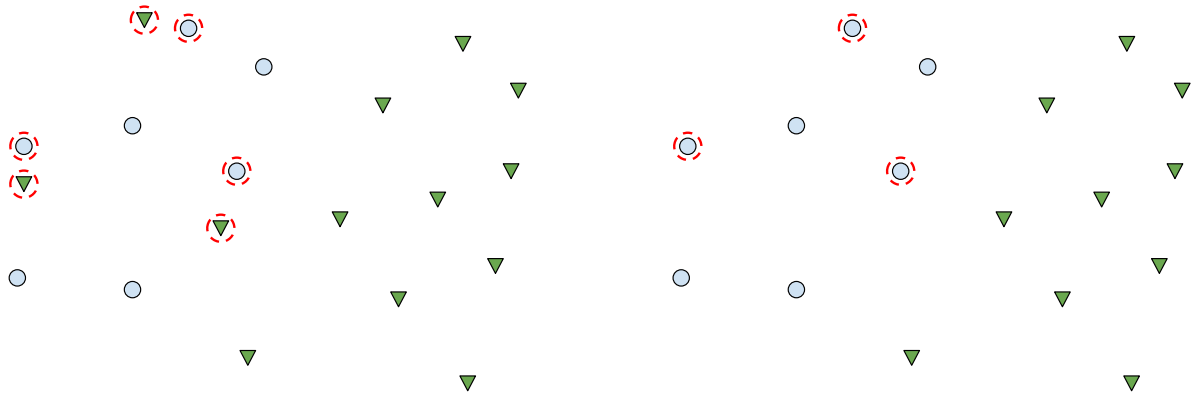
wartość C implikuje mniej restrykcyjne karanie błędnie klasyfikowanych obserwacji, przez co modele bazowe charakteryzują się wyższym stopniem generalizacji. Algorytm *Boosting-IB* wykorzystujący modele *SVM* jako klasyfikatory bazowe oznaczany będzie w pracy jako *BoostingSVM-IB*.

3.5 Algorytm *BoostingSVM-IB* z redukcją obserwacji nadmiarowych

Rzeczywiste dane wykorzystywane w procesie uczenia zawierają obserwacje wprowadzające szum informacyjny (ang. *"noise" examples*). Tego typu obserwacje pojawiają się w bliskim otoczeniu obiektów należących do innych klas, co może wpływać na zniekształcenie decyzji podejmowanych przez klasyfikator. Przykładowo, proces uczenia z wykorzystaniem klasyfikatora *BoostingSVM-IB* na danych zawierających tego typu obserwacje prowadzi do iteracyjnego zwiększania wartości ich wag w kolejnych iteracjach pętli wzmacniania. Przyczyną występowania obserwacji „zaszumiających" może być błędne pobranie wartości pewnych cech obiektu bądź też anomalne wystąpienie pewnego zjawiska. Rozwiązanie problemu pojawiających się obserwacji wprowadzających szum informacyjny jest niezwykle istotne w przypadku danych niezbalansowanych, gdzie koszt błędnego zaklasyfikowania obserwacji z klasy zdominowanej jest niejednokrotnie wyższy niż koszt błędnej klasyfikacji obiektu z klasy dominującej. Najistotniejszym elementem uczenia z danych niezbalansowanych jest detekcja i eliminacja obserwacji wprowadzających szum informacyjny należących do klasy dominującej.

Typowym podejściem do eliminacji problemu obserwacji wprowadzających szum informacyjny jest zastosowanie koncepcji połączeń wzajemnego sąsiedztwa obiektów należących do dwóch klas, w literaturze nazywanych połączeniami typu *Tomek* [132]. Dwie obserwacje, $\mathbf{x}_n$ oraz $\mathbf{x}_m$, należące do dwóch różnych klas ($y_n \neq y_m$) są związane relacją typu *Tomek* wtedy i tylko wtedy, gdy nie istnieje obserwacja $\mathbf{x}_l$ taka że $d(\mathbf{x}_m, \mathbf{x}_l) < d(\mathbf{x}_n, \mathbf{x}_m)$, bądź $d(\mathbf{x}_n, \mathbf{x}_l) < d(\mathbf{x}_n, \mathbf{x}_m)$. Miara $d(\cdot, \cdot)$ jest zadaną w przestrzeni miarą odległości pomiędzy obserwacjami.

Eliminacja obserwacji odbywa się poprzez wyszukanie wszystkich obiektów z klasy dominującej reprezentujących połączenie typu *Tomek* i ich usunięcie ze zbioru uczącego. Po-



(a) Obserwacje związane relacją typu *Tomek*.

(b) Dane po eliminacji obserwacji dominujących.

Rysunek 3.3: Wykorzystanie połączeń typu *Tomek* do eliminacji obserwacji nadmiarowych należących do klasy dominującej.

przez zastosowanie tej techniki eliminacji zostaną usunięte zarówno obiekty wprowadzające szum informacyjny, jak i elementy znajdujące się w bliskim sąsiedztwie hiperpłaszczyzny separującej. Istotnym elementem proponowanej metody eliminacji obserwacji jest wybór odpowiedniej miary odległości $d(\cdot,\cdot)$, gdyż powinna być ona zgodna z przyjętą miarą podobieństwa wykorzystywaną przez klasyfikator. Dla klasyfikatora SVM z zadaną funkcją jądra postaci $K(\mathbf{x}_n,\mathbf{x}_m) = \langle\phi(\mathbf{x}_n)\phi(\mathbf{x}_m)\rangle$, który stanowi klasyfikator liniowy w przestrzeni $\phi(X)$ proponowana w pracy jest następująca miara podobieństwa:

$$\begin{aligned} ||\phi(\mathbf{x}_n) - \phi(\mathbf{x}_m)||_2^2 &= ||\phi(\mathbf{x}_n)||_2^2 + ||\phi(\mathbf{x}_m)||_2^2 - 2\phi(\mathbf{x}_n)^T\phi(\mathbf{x}_m) \\ &= K(\mathbf{x}_n,\mathbf{x}_n) + K(\mathbf{x}_m,\mathbf{x}_m) - 2K(\mathbf{x}_n,\mathbf{x}_m). \end{aligned} \tag{3.48}$$

Eliminacja z wykorzystaniem połączeń *Tomek* odbywa się na etapie przetwarzania danych opisanych w tej sekcji dwóch modyfikacji algorytmu *BoostingSVM-IB*.

Przyjęcie ważonego kryterium postaci (3.34) zakłada, że poziom niebalansowania w pobliżu hiperpłaszczyzny separującej jest równy niezbalansowaniu całego zbioru danych. Innymi słowy zakłada się, że stosunek liczności klasy zdominowanej do dominującej $\frac{N_-}{N_+}$ dla całego zbioru uczącego jest bliski stosunkowi liczności obiektów znajdujących się w bezpośrednim sąsiedztwie hiperpłaszczyzny separującej i kluczowych z punktu widzenia zadania

klasyfikacji. W rzeczywistości, co pokazują wyniki badań empirycznych opublikowane m. in. w pracy [38], dane znajdujące się w bezpośrednim sąsiedztwie hiperpłaszczyzny separującej charakteryzują się niższym wskaźnikiem niezbalasnowania niż obserwacje w całym zbiorze treningowym.

Konieczne jest więc zastosowanie odpowiednich mechanizmów wyboru obserwacji kluczowych w kolejnych iteracjach konstrukcji klasyfikatorów bazowych dla opracowanego w ramach rozprawy złożonego algorytmu typu *SVM*. W niniejszej rozprawie proponuje się dwie metody wyznaczania obserwacji kluczowych wykorzystujących techniki wspomnianego we wstępie uczenia aktywnego [114]. Pierwotnie, uczenie aktywne było stosowane do rozwiązywania problemów charakteryzujących się wysokim kosztem nadania etykiet klas obiektom należącym do zbioru uczącego. Uczenie polegało na iteracyjnym wyborze najbardziej informacyjnych obserwacji z niezaetykietowanego zbioru danych i odkryciu ich rzeczywistych klas. Z czasem, podejścia dotyczące uczenia aktywnego zaczęto stosować również do problemu niezbalansowania danych [38, 39, 96].

W ramach rozprawy opracowano dwie modyfikacje metody *BoostingSVM-IB*:

- *BoostingSVM-IB.M1*. Modyfikacja zakłada wykonanie algorytmu selekcji jednostronnej na zbiorach danych wykorzystywanych do konstrukcji klasyfikatorów bazowych.
- *BoostingSVM-IB.M2*. Modyfikacja wykorzystuje procedurę selekcji obserwacji z wykorzystaniem poszerzonego marginesu SVM w każdej (począwszy od $k = 2$) iteracji pętli wzmacniania.

Każda z prezentowanych dwóch metod modyfikuje Algorytm 4 w kroku 5, gdzie zamiast całego zbioru $\mathbb{S}_N$ algorytm SMO wykorzystuje zbiór danych po zastosowaniu odpowiedniej metody selekcji obserwacji informacyjnych.

Pierwszy z opracowanych w ramach rozprawy mechanizmów wyboru obserwacji informacyjnych bazuje na algorytmie selekcji jednostronnej (*ang. one-sided selection*) opisanym w pracy [74]. W każdej iteracji pętli *boostingowej* wybierane są wszystkie obserwacje należące do klasy zdominowanej oraz wybrane obserwacje z klasy dominującej. Wybór obserwacji odbywa się zgodnie z Algorytmem 5. W pierwszym kroku generowany jest zbiór uczący $\mathbb{S}_{N_+}$ zawierający jedynie obserwacje z klasy zdominowanej. W następnym kroku losowana jest obserwacja z klasy dominującej, z rozkładu:

Algorithm 5: Algorytm selekcji jednostronnej

Input : Ważony zbiór uczący $\mathbb{S}_N = \{(\mathbf{x}_1, y_1, w_1^{(k)}), \ldots, (\mathbf{x}_N, y_N, w_N^{(k)})\}$

Output: Zredukowany zbiór uczący $\mathbb{S}_{N_2}$

$\mathbb{S}_{N_2} \longleftarrow \emptyset$;

Wyznacz zbiór $\mathbb{S}_{N_+} = \{(\mathbf{x}_n, y_n, w_n^{(k)}) \in \mathbb{S}_N : y_n = +1\}$;

Wyznacz zbiór $\mathbb{S}_{N_-} = \{(\mathbf{x}_n, y_n, w_n^{(k)}) \in \mathbb{S}_N : y_n = -1\}$;

Wylosuj obserwację $(\mathbf{x}_n, -1, w_n^{(k)})$ należącą do zbioru $\mathbb{S}_{N_-}$ z rozkładu $p(n)$ zadanego wzorem 3.49;

Wykonaj: $\mathbb{S}_{N_+} \longleftarrow \mathbb{S}_{N_+} \cup \{(\mathbf{x}_n, -1, w_n^{(k)})\}$;

for $(\mathbf{x}_m, y_m, w_m^{(k)}) \in \mathbb{S}_{N_-}$ **do**

  Wyznacz obserwację $(\mathbf{x}_l, y_l, w_l^{(k)}) \longleftarrow \underset{(\mathbf{x}_l, y_l, w_l^{(k)}) \in \mathbb{S}_{N_+}}{\operatorname{argmin}} d(\mathbf{x}_m, \mathbf{x}_l)$;

  if $y_l \neq y_m$ **then**

    Wykonaj: $\mathbb{S}_{N_2} \longleftarrow \mathbb{S}_{N_2} \cup \{(\mathbf{x}_l, y_l, w_l^{(k)})\}$;

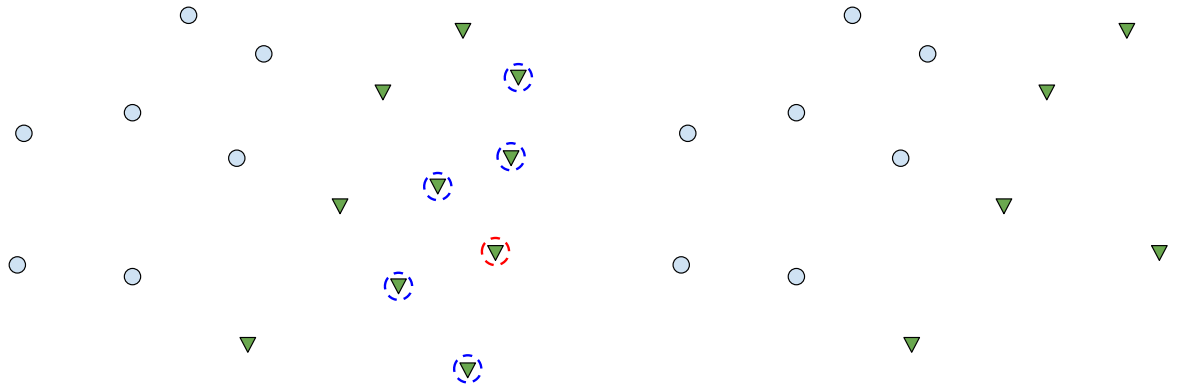
  end

end

$\mathbb{S}_{N_2} \longleftarrow \mathbb{S}_{N_2} \cup \mathbb{S}_{N_+}$;

$$p(n) = \frac{\frac{1}{w_n^{(k)}}}{\sum_{n \in \mathbb{N}_-} \frac{1}{w_n^{(k)}}}, \tag{3.49}$$

gdzie wartości wag $w_n^{(k)}$ aktualizowane są w kolejnych iteracjach konstrukcji klasyfikatorów bazowych, zgodnie z procedurą (3.36). Wybór takiego rozkładu dla losowania próbki oznacza, że z większym prawdopodobieństwem będą losowane te obserwacje, które były poprawnie klasyfikowane przez dotychczas skonstruowane klasyfikatory bazowe. Na Rysunku 3.4a zaznaczono na czerwono obserwację, która została wybrana w wyniku losowania. W dalszych krokach metody identyfikowane są wszystkie te obserwacje należące do klasy dominującej, które znajdują się bliżej wylosowanej obserwacji, niż jakiejkolwiek innej należącej do klasy zdominowanej. Wybierane są więc te obserwacje z klasy dominującej, które są poprawnie klasyfikowane przez klasyfikator najbliższego sąsiada wyuczony



(a) Obserwacja wybrana w wyniku losowania (kolor czerwony) i obserwacje wybrane do eliminacji (kolor niebieski).

(b) Dane po wykonaniu selekcji jednostronnej.

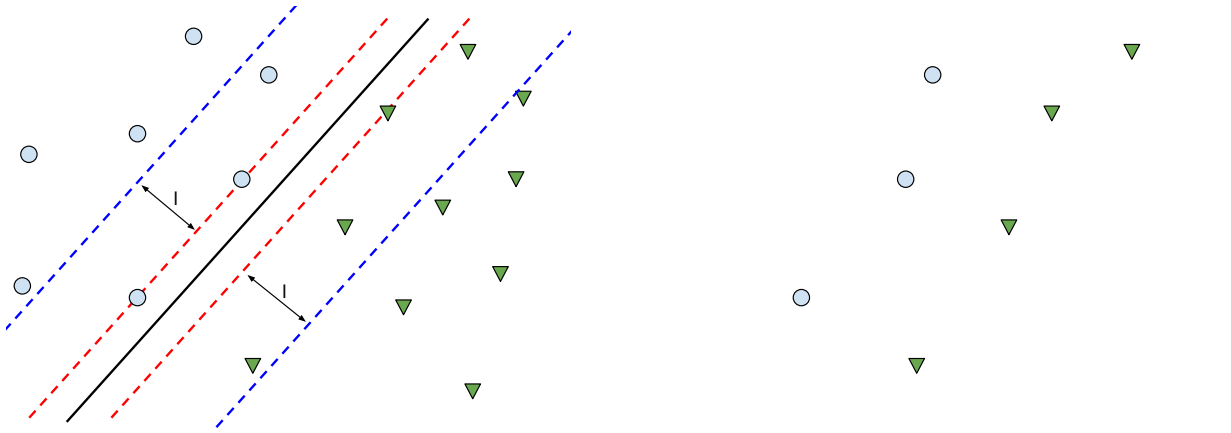
Rysunek 3.4: Wybór obserwacji informacyjnych z wykorzystaniem algorytmu selekcji jednostronnej.

na zbiorze $\mathbb{S}_{N_+}$ powiększonym o wylosowaną obserwację. Na Rysunku 3.4a wybrane obserwacje zostały oznaczone na niebiesko. Zbiór po eliminacji wyszczególnionych obserwacji został przedstawiony na Rysunku 3.4b.

Druga z proponowanych metod aktywnej selekcji obserwacji wykorzystuje pojęcie szerokiego marginesu (*ang. wide margin*) klasyfikatora SVM. W każdej kolejnej iteracji (począwszy od $k = 2$ - w pierwszym kroku procedury wzmacniania zbiór $\mathbb{S}_{N_1}$ powinien zostać wygenerowany poprzez zastosowanie selekcji jednostronnej) zbiór uczący $\mathbb{S}_{N_k}$ wykorzystywany do budowy k-tego klasyfikatora bazowego konstruowany jest zgodnie z regułą:

$$\mathbb{S}_{N_k} = \{(\mathbf{x}_n, y_n) \in \mathbb{S}_N : y_n y_{k-1}(\mathbf{x}_n) \leqslant 1 + l\}, \tag{3.50}$$

gdzie $y_{k-1}(\mathbf{x}_n)$ stanowi wyjście $k-1$ bazowego klasyfikatora SVM, natomiast l ($l \geqslant 0$) jest parametrem algorytmu selekcji obserwacji informacyjnych, reprezentującym odległość szerokiego marginesu od rzeczywistego marginesu SVM dla przeskalowanej przestrzeni danych. W przeskalowanej przestrzeni szerokość marginesu wynosi 2, natomiast odległości H_+ od H, oraz H_- od H wynoszą 1, parametr l określa więc procentowo o ile poszerzony został margines separujący w stosunku do marginesu wyznaczonego w procesie uczenia



(a) Zastosowanie szerokiego marginesu SVM do znalezienia obserwacji informacyjnych.

(b) Dane po eliminacji obserwacji znajdujących się poza marginesem.

Rysunek 3.5: Wybór obserwacji informacyjnych z wykorzystaniem szerokiego marginesu SVM.

klasyfikatora SVM. Im wyższa wartość parametru l tym większa liczba obserwacji zebrana w procesie aktywnej selekcji. Rysunek 3.5a przedstawia przykładowy szeroki margines dla wybranego zestawu danych. Rysunek 3.5b przedstawia zestaw danych po eliminacji obserwacji z wykorzystaniem szerokiego marginesu. W odróżnieniu od metody selekcji jednostronnej wybór obserwacji poprzez zastosowanie szerokiego marginesu SVM odbywa się również względem klasy zdominowanej.

3.6 Przypadek wieloklasowy

Proponowane w rozprawie algorytmy odnoszą się do dychotomicznych zadań klasyfikacji. W literaturze spotyka się szereg metod, które dają możliwość wykorzystania klasyfikatorów dwuklasowych do zadań o większej liczbie możliwych klas. Typowym podejściem stosowanym w literaturze jest metoda *jeden-przeciw-reszcie* (*ang. one-versus-rest*) [10]. Metoda ta polega na wyuczeniu Y dychotomicznych klasyfikatorów w ten sposób, że każdy z klasyfikatorów konstruowany jest na tym samym zbiorze danych, jednak za każdym razem rozważa się tylko dwie klasy: aktualną klasę y oraz klasę powstałą w wyniku połączenia klas pozostałych. Istotną wadą scharakteryzowanego podejścia jest to, że dla klasycz-

nych metod uczenia klasyfikatorów występuje problem niebalansowania danych. Zakładając równomierny rozkład obserwacji dla Y-klasowego zadania stosunek liczności klasy dominującej do zdominowanej jest rzędu $Y-1$ dla każdego zbioru wykorzystywanego do budowy klasyfikatorów w metodzie *jeden-przeciw-reszcie*. Problem ten eliminowany jest zupełnie w przypadku zastosowania opracowanych metod, ponieważ uwzględniają one zarówno niezbalansowanie wynikające z samego charakteru danych, jak i to wynikające z przyjętej metodyki konstrukcji klasyfikatorów w ramach podejścia *jeden-przeciw-reszcie*. Opracowane w ramach rozprawy metody klasyfikacji dla danych niezbalansowanych umożliwiają bezproblemowe stosowanie podejścia *jeden-przeciw-reszcie* dla przypadków wieloklasowych.

Alternatywnym podejściem do klasyfikacji wieloklasowej z wykorzystaniem dychotomicznych klasyfikatorów jest metoda *jeden-przeciw-jednemu* (*ang. one-versus-one*) [10]. Podejście to zakłada zbudowanie $\frac{Y(Y-1)}{2}$ dwuklasowych klasyfikatorów reprezentujących wszystkie możliwe pary klas. Klasyfikacja z wykorzystaniem tego podejścia odbywa się poprzez sprawdzenie wyników klasyfikacji dla wszystkich zbudowanych klasyfikatorów dychotomicznych i wyborze tej klasy, która została zwracana najczęściej. Zasadniczą wadą podejścia jest konieczność zbudowania dużej liczby klasyfikatorów. Pewne modyfikacje metody redukujące liczbę potrzebnych modeli zostały opisane w pracach [2, 99].

3.7 Uwagi

W ramach rozprawy zaproponowano uogólnione kryterium uczenia klasyfikatora *SVM* dla danych niezbalansowanych postaci (3.7). Przedstawiono modyfikację algorytmu uczenia SVM metodą SMO dla przyjętego kryterium optymalizacji. Niezależnie od opracowanych rozwiązań, zaproponowano algorytm uczenia klasyfikatora złożonego *Boosting-IB* minimalizujący sekwencyjnie funkcję błędu postaci (3.37). Pokazano w jaki sposób wykorzystać opracowany model klasyfikatora *SVM* jako klasyfikator bazowy w metodzie *Boosting-IB*, czego wynikiem było opracowanie klasyfikatora *BoostingSVM-IB*. W dalszej kolejności zaproponowano dwie modyfikacje algorytmu *BoostingSVM-IB* poprzez zastosowanie metod aktywnego uczenia z wykorzystaniem eliminacji jednostronnej (*BoostingSVM-IB.M1*) i szerokiego marginesu SVM (*BoostingSVM-IB.M2*).

Modyfikacja algorytmu SMO nie wpływa na zmianę jego złożoności obliczeniowej, która w najlepszym wypadku wynosi $O(N_{svm} \cdot N)$ [34], gdzie N oznacza liczbę obiektów należących do zbioru uczącego natomiast, N_{svm} oznacza liczbę wektorów wspierających. Dla wzmocnionego *boostingiem* klasyfikatora SVM złożoność obliczeniowa wynosi $O(K \cdot N_{svm} \cdot N)$. Przy założeniu, że liczba klasyfikatorów bazowych jest zdecydowanie mniejsza niż liczba wektorów wspierających ($K << N_{svm}$) rząd złożoności obliczeniowej dla algorytmu *BoostingSVM-IB* jest taki sam jak w przypadku klasycznego algorytmu SMO. Zastosowanie mechanizmów uczenia aktywnego redukuje złożoność do rzędu $O(N_{svm,active} \cdot N_{active})$, gdzie $N_{active})$, oraz $N_{svm,active}$ stanowią kolejno liczność zbioru i liczbę zidentyfikowanych wektorów wspierających dla tego zbioru bazowego, dla którego iloczyn wspomnianych dwóch wartości jest najwyższy.

Opracowane w ramach doktoratu metody stanowią rozwinięcie koncepcji dotyczącej zastosowania klasyfikatorów SVM dla danych niezbalansowanych [92, 136], w szczególności zastosowania wzmacnianych klasyfikatorów SVM [137]. Zarówno podejście opisane w pracy [137] jak i metody opracowane w ramach rozprawy wykorzystują koncepcję wzmacniania klasyfikatorów, jednak algorytmy te różni sposób podejścia do rozwiązania problemu niezbalasnowania. Autorzy [137] proponują przyjęcie kryterium uczenia postaci (3.5), gdzie zamiast modyfikacji wartości zmiennych ξ_n w kolejnych iteracjach pętli *boostingowej* proponuje się modyfikację wartości wag obserwacji w procesie konstrukcji kolejnych klasyfikatorów bazowych. W niniejszej rozprawie rozważa się uogólnione kryterium postaci (3.7), które umożliwia sterowanie złożonością modelu poprzez parametr C, z drugiej strony umożliwia naprzemienną optymalizację przyjętego kryterium jakości w sposób wewnętrzny poprzez optymalizację funkcji błędu względem dodatkowych zmiennych ξ_n, oraz zewnętrzny poprzez sekwencyjną optymalizację wag dla przyjętego wykładniczego kryterium niezbalansowania postaci (3.37). W pracy [137] brak jest ponadto wyszczególnienia propozycji metody uczenia klasyfikatorów bazowych. W ramach rozprawy został opracowany zmodyfikowany algorytm SMO na potrzeby konstrukcji bazowych klasyfikatorów SVM, dla przyjętego w pracy kryterium uczenia. Ponadto, w ramach pracy zaproponowano zastosowanie dwóch metod aktywnego uczenia, które są wykorzystane w konstrukcji klasyfikatorów bazowych celem zwiększenia jakości predykcji.

Rozdział 4

Badania empiryczne

W rozdziale opisano wyniki badań empirycznych przeprowadzonych celem porównania jakości opracowanych w rozprawie metod z najskuteczniejszymi rozwiązaniami opisanymi w literaturze. Badania wykonano z wykorzystaniem zestawu 44 danych *benchamarkowych* charakteryzujących się różnym stopniem niezbalansowania.

4.1 Cel badań

Celem badań była analiza jakości opracowanych metod *BoostingSVM-IB*, *BoostingSVM-IB.M1*, *BoostingSVM-IB.M2* w kontekście ich zastosowania do problemu niezbalansowanych danych. Do oceny jakości opracowanych metod wykorzystano następujące wskaźniki:[1]

- wskaźnik średniej geometrycznej ($GMean$) zadany wzorem (1.17),
- pole powierzchni pod krzywą ROC (AUC) zadane wzorem (1.24),
- poprawność klasyfikacji (Acc) zadaną wzorem (1.15),
- wskaźnik czułości (TP_{rate}) zadany wzorem (1.19),
- wskaźnik specyficzności (TN_{rate}) zadany wzorem (1.18).

[1] Celem zwiększenia czytelności podawanych wyników wartości wskaźników podawane są w ujęciu procentowym.

Kryteria $GMean$, oraz AUC charakteryzują jakość metod klasyfikacji w kontekście niezbalansowania zbioru uczącego i stanowią kluczowe metryki w przeprowadzonej analizie jakości.

Analiza jakości przeprowadzona została na zestawie 44 zbiorów *benchmarkowych* należących do różnych dziedzin i charakteryzujących się różnym stopniem niezbalansowania. Jakość opracowanych metod mierzona względem przyjętych kryteriów została porównana do jakości innych metod dedykowanych do rozwiązania problemu niezbalansowania danych.

4.2 Metodyka i narzędzia

W ramach badań zastosowano powszechnie stosowaną do oceny algorytmów klasyfikacji metodykę rozłożonej krzyżowej walidacji (*ang. stratified cross-validation*) [70]. Polega ona na podziale zbioru danych na zestaw K podzbiorów (*ang. folds*) zachowujących rozkład pomiędzy klasami. Proces oceny jakości klasyfikatora polega na K-krotnym wyuczeniu klasyfikatora na zestawie $K-1$ podzbiorów danych i walidacji na niewybranym zbiorze w taki sposób, że dla każdego powtórzenia wykorzystywany jest inny podzbiór danych do oceny. Metodyka krzyżowej walidacji jest powszechnie stosowana do oceny algorytmów klasyfikacji ze względu na fakt, iż w procesie testowania biorą udział wszystkie elementy zbioru treningowego, uczenie i klasyfikacja odbywają się w każdej iteracji na rozłącznych zbiorach danych, a czas potrzebny do przeprowadzenia eksperymentu nie jest aż tak duży jak w przypadku innych metod, takich jak podejście „pozostaw jedną” (*ang. leave-one-out*) [70].

Jako narzędzie do badań wykorzystano środowisko *KEEL Software* [1, 44]. Narzędzie to posiada moduł do analizy jakości klasyfikatorów dedykowanych do rozwiązania problemów niezbalansowania, jednocześnie umożliwia importowanie własnych algorytmów do środowiska. Wszystkie trzy metody opracowane w ramach rozprawy doktorskiej zostały zaimplementowane w języku *Java* z wykorzystaniem biblioteki *Weka* [54], następnie udostępnione w narzędziu *KEEL Software*.

4.3 Zbiory danych

Do oceny opracowanych algorytmów klasyfikacji wykorzystano 44 zbiory danych które są udostępnione zarówno w środowisku *KEEL* [49], jak i za pośrednictwem strony internetowej [2], na której znajduje się również dokładny opis dla każdego ze zbiorów. W przypadku zbiorów, w których wystąpiły więcej niż dwie klasy dokonano łączenia klas celem otrzymania zbiorów dychotomicznych. Charakterystyka zbiorów danych została opisana w Tabeli 4.5. Dla każdego zbioru danych podana została liczba obserwacji ($\#Obs.$), liczba atrybutów ($\#Atr.$), procent obserwacji należących do klasy pozytywnej ($\%P$), procent obserwacji należących do klasy negatywnej ($\%N$), oraz wskaźnik niezbalansowania (Imb_{rate}), liczony jako stosunek $\%N$ do $\%P$. Do eksperymentu wybrane zostały zbiory o różnym stopniu niezbalansowania począwszy od sytuacji, w której liczność klasy dominującej była prawie dwukrotnie wyższa ($Imb_{rate} = 1.82$), a skończywszy na sytuacji w której wskaźnik niezbalansowania wynosił 128.87.

4.4 Metody

Jakość działania opracowanych algorytmów *BoostingSVM-IB* (**BSI**), *BoostingSVM-IB.M1* (**BSI1**) oraz *BoostingSVM-IB.M2* (**BSI2**) została porównana z działaniem następujących algorytmów:

- *SVM* (**SVM**). Klasyfikator SVM uczony z wykorzystaniem algorytmu SMO.
- *SVM + SMOTE* (**SSVM**). Klasyfikator SVM wyuczony na danych, w przypadku których wykorzystano próbkowanie metodą *SMOTE*.
- *SMOTEBoostSVM* (**SBSVM**). Wzmacniany klasyfikator SVM, dla którego przed konstrukcją klasyfikatora bazowego wykorzystano algorytm *SMOTE* celem wygenerowania syntetycznych próbek należących do klasy dominującej.
- *C-SVM* (**CSVM**). Wrażliwy na koszt klasyfikator SVM szczegółowo opisany w pracach [125, 136].

[2] http://www.keel.es/dataset.php

- *AdaCost* (**AdaC**). Zespół wrażliwych na koszt klasyfikatorów SVM, dla których wagi obserwacji z klasy zdominowanej aktualizowane są o wyższe wartości niż wagi obiektów z klasy dominującej [42].

- *SMOTEBoost* (**SBO**). Zmodyfikowany algorytm wzmacniania, w ramach którego klasyfikatory bazowe konstruowane są z wykorzystaniem zbiorów poszerzonych o syntetyczne próbki wygenerowane metodą *SMOTE* [25].

- *RUSBoost* (**RUS**). Stanowi rozszerzenie algorytmu *SMOTEBoost*, polegające na zastosowaniu losowej eliminacji obiektów z klasy dominującej, za każdym razem, gdy budowany jest klasyfikator bazowy [113].

- *SMOTEBagging* (**SB**). Wykorzystuje algorytm *SMOTE* do konstrukcji klasyfikatorów bazowych w procesie wzmacniania [139].

- *UnderBagging* (**UB**). Wykorzystuje mechanizmy eliminacji losowej obiektów w procesie konstrukcji klasyfikatorów bazowych [127].

Do analizy porównawczej algorytmów zaproponowane zostały dwie grupy metod: zestaw algorytmów bazujących na modelu *SVM*, oraz zestaw zespołów klasyfikatorów będących przedmiotem analizy przeprowadzonej w pracy [49]. W ramach pierwszej z grup wybrano szereg modyfikacji klasyfikatora *SVM* dedykowanych do problemu niezbalansowania, począwszy od klasycznego *SVM*, a skończywszy na metodzie wzmacniania z wykorzystaniem próbkowania metodą *SMOTE*. Drugą grupę metod stanowią złożone metody klasyfikacji wyselekcjonowane na podstawie wyników badań przedstawionych w [49].

Parametry metod wykorzystanych do badań zostały dobrane bazując na wynikach opisanych w literaturze [49, 125]. Parametry metod opracowanych w ramach rozprawy zostały dobrane metodą walidacji krzyżowej.

4.5 Wyniki i dyskusja

Podstawowym kryterium stosowanym do oceny opracowanych metod klasyfikacji był wskaźnik $GMean$. Analizując wyniki badań opisane w Tabelach 4.6-4.10 widać wyraźnie, że klasyczny klasyfikator SVM pomimo wysokiej poprawności klasyfikacji charakteryzował

Pary metod	$\mathbf{R^+}$	$\mathbf{R^-}$	$\mathbf{p-wartość}$	**FWER**	**Hipoteza**
BSI vs. **SVM**	900.0	3.0	0.0000	0.0056	odrzucona na rzecz **BSI**
BSI vs. **CSVM**	835.0	26.0	0.0000	0.0063	odrzucona na rzecz **BSI**
BSI vs. **SBSVM**	819.0	42.0	0.0000	0.0071	odrzucona na rzecz **BSI**
BSI vs. **AdaC**	841.0	105.0	0.0000	0.0083	odrzucona na rzecz **BSI**
BSI vs. **SBO**	869.0	121.0	0.0000	0.0100	odrzucona na rzecz **BSI**
BSI vs. **SSVM**	715.0	146.0	0.0002	0.0125	odrzucona na rzecz **BSI**
BSI vs. **UB**	715.0	275.0	0.0102	0.0167	odrzucona na rzecz **BSI**
BSI vs. **RUS**	678.0	268.0	0.0133	0.0250	odrzucona na rzecz **BSI**
BSI vs. **SB**	651.0	295.0	0.0316	0.0500	odrzucona na rzecz **BSI**

Tabela 4.1: Wyniki testu Wilcoxona z procedurą Holma-Bonferroniego dotyczące porównania metody **BSI** z innymi algorytmami opisanymi w literaturze. W badaniu przyjęto poziom istotności równy $\alpha_{ist} = 0.05$.

Pary metod	$\mathbf{R^+}$	$\mathbf{R^-}$	$\mathbf{p-wartość}$	**FWER**	**Hipoteza**
BSI1 vs. **SVM**	942.0	4.0	0.0000	0.0056	odrzucona na rzecz **BSI1**
BSI1 vs. **SSVM**	855.0	91.0	0.0000	0.0063	odrzucona na rzecz **BSI1**
BSI1 vs. **CSVM**	837.0	109.0	0.0000	0.0071	odrzucona na rzecz **BSI1**
BSI1 vs. **SBSVM**	827.0	119.0	0.0000	0.0083	odrzucona na rzecz **BSI1**
BSI1 vs. **AdaC**	813.0	133.0	0.0000	0.0100	odrzucona na rzecz **BSI1**
BSI1 vs. **SBO**	789.0	157.0	0.0001	0.0125	odrzucona na rzecz **BSI1**
BSI1 vs. **UB**	663.0	240.0	0.0082	0.0167	odrzucona na rzecz **BSI1**
BSI1 vs. **SB**	651.0	252.0	0.0126	0.0250	odrzucona na rzecz **BSI1**
BSI1 vs. **RUS**	678.0	268.0	0.0133	0.0500	odrzucona na rzecz **BSI1**

Tabela 4.2: Wyniki testu Wilcoxona z procedurą Holma-Bonferroniego dotyczące porównania metody **BSI1** z innymi algorytmami opisanymi w literaturze. W badaniu przyjęto poziom istotności równy $\alpha_{ist} = 0.05$.

się zdecydowanie niższą wartością $GMean$ niż metody dedykowane do danych niezbalansowanych.

Pary metod	$\mathbf{R^+}$	$\mathbf{R^-}$	$\mathbf{p-wartości}$	**FWER**	**Hipoteza**
BSI2 vs. **SVM**	942.0	4.0	0.0000	0.0056	odrzucona na rzecz **BSI1**
BSI2 vs. **CSVM**	912.0	34.0	0.0000	0.0063	odrzucona na rzecz **BSI1**
BSI2 vs. **SBSVM**	900.0	46.0	0.0000	0.0071	odrzucona na rzecz **BSI1**
BSI2 vs. **SSSVM**	867.0	79.0	0.0000	0.0083	odrzucona na rzecz **BSI1**
BSI2 vs. **SBO**	858.0	88.0	0.0000	0.0100	odrzucona na rzecz **BSI1**
BSI2 vs. **AdaC**	852.0	94.0	0.0000	0.0125	odrzucona na rzecz **BSI1**
BSI2 vs. **RUS**	741.0	205.0	0.0012	0.0167	odrzucona na rzecz **BSI1**
BSI2 vs. **UB**	702.0	201.0	0.0017	0.0250	odrzucona na rzecz **BSI1**
BSI2 vs. **SB**	695.0	208.0	0.0023	0.0500	odrzucona na rzecz **BSI1**

Tabela 4.3: Wyniki testu Wilcoxona z procedurą Holma-Bonferroniego dotyczące porównania metody **BSI2** z innymi algorytmami opisanymi w literaturze. W badaniu przyjęto poziom istotności równy $\alpha_{ist} = 0.05$.

Pary metod	$\mathbf{R^+}$	$\mathbf{R^-}$	$\mathbf{p-wartość}$	**Hipoteza**
BSI vs. **BSI1**	490.0	456.0	0.8326	nie odrzucona
BSI vs. **BSI2**	355.0	591.0	0.1525	nie odrzucona
BSI1 vs. **BSI2**	361.0	405.0	0.2547	nie odrzucona

Tabela 4.4: Wyniki testu Wilcoxona przeprowadzonego pomiędzy metodami **BSI**, **BSI1**, oraz **BSI2**. W badaniu przyjęto poziom istotności równy $\alpha_{ist} = 0.05$.

Spośród wszystkich metod najwyższa średnia wartość wskaźnika $GMean$ liczona dla wszystkich zbiorów danych osiągnięta została przez metodę **BSI2** (Tabela 4.6). Kolejnymi metodami które osiągnęły najwyższą wartość wskaźnika były opracowane w ramach rozprawy algorytmy **BSI**, oraz **BSI1**. Najwyższą wartością $GMean$ spośród algorytmów klasyfikacji pochodzących z literatury charakteryzowały się kolejno metody **UB**, **RUS**, oraz **SSVM**.

Celem wykazania wysokiej jakości opracowanych w rozprawie metod wykorzystano procedurę Holma-Bonferroniego [59] pozwalającą na jednoczesne testowanie wielu hipotez. Procedurę tą opisać można w dwóch krokach:

- W pierwszym kroku został przeprowadzony szereg dwustronnych testów Wilcoxona

dla par obserwacji [51] mających na celu wyznaczenie p-wartości dla ciągu hipotez:

$$\mathcal{H}_i : \theta_{BSI} \neq \theta_i, \tag{4.1}$$

gdzie θ_{BSI} oznacza medianę wskaźnika $GMean$ liczoną na wszystkich zbiorach danych dla analizowanej metody, natomiast θ_i medianę osiągniętą przez i-tą metodę referencyjną rozpatrywaną w eksperymencie.

- W drugim kroku uszeregowano wyniki testów względem p-wartości i sprawdzono, czy dla uszeregowanego ciągu zachodzi nierówność:

$$pval_i \leqslant FWER_i, \tag{4.2}$$

gdzie $pval_i$ oznacza i-tą p-wartość w uszeregowaniu. Współczynnik $FWER_i$ (*ang. familywise error rate*) dla zadanego progu ufności α_{ist} $FWER_i$ definiowany jest w następujący sposób:

$$FWER_i = \frac{\alpha_{ist}}{M + 1 - i}, \tag{4.3}$$

gdzie M oznacza liczbę testowanych hipotez. Jeżeli dla hipotezy $\mathcal{H}_i$ zachodzi nierówność (4.2), wówczas jest odrzucana.

Analizę statystyczną przeprowadzono niezależnie dla metod **BSI**, **BSI1**, oraz **BS2**. Wyniki przeprowadzonych testów statystycznych przedstawiono w Tabelach 4.1-4.3. Wartości R^+, oraz R^- oznaczają sumy rang dwustronnego testu Wilcoxona kolejno dla badanej metody, oraz metod referencyjnych. Wyniki badań uszeregowano rosnąco względem p-wartości i wyznaczono odpowiadające im wartości wskaźnika $FWER$. Dla każdej z analizowanych metod wyniki testu dawały podstawę do odrzucenia każdej z hipotez dotyczących równości median. Rozpatrując z osobna każdą z metod **BSI**, **BSI1**, oraz **BS2** można stwierdzić z prawdopodobieństwem równym 0.95, że każda z nich jest lepsza niż rozważane w badaniu metody referencyjne opisane w literaturze. W ramach przeprowadzonych badań porównano również parami metody **BSI**, **BSI1**, oraz **BS2** (Tabela 4.4). Dla żadnej z par nie było podstaw do odrzucenia hipotezy co oznacza, że nie ma istotnej różnicy pomiędzy wynikami osiąganymi przez metody opisane w rozprawie.

Opracowane metody porównano również ze względu na kryterium AUC, które jest mniej wrażliwe na dysproporcje w wartościach TP_{rate}, oraz TN_{rate}. Wszystkie trzy metody

opracowane w ramach doktoratu dały wyższe wyniki niż algorytmy z literatury. W odróżnieniu od kryterium $GMean$, najwyższa wartość wskaźnika AUC została uzyskana dla metody **BSI** (Tabela 4.7).

Dla kryterium poprawności klasyfikacji najwyższe wartości osiągnięte zostały dla metod takich jak **SVM**, **SBSVM**, **SBO** które charakteryzowały się wysoką wartością wskaźnika TN_{rate} i niską wartością TP_{rate} (Tabele 4.8, 4.9, oraz 4.10). Najwyższa wartość wskaźnika TP_{rate} została osiągnięta dla metod **UB**, oraz **SSVM**, którez kolei osiągały najniższe wartości wskaźnika TN_{rate}.

Analiza wyników eksperymentu pozwala stwierdzić, że najwyższą jakością charakteryzowały się opisane w ramach rozprawy metody **BSI**, **BSI1**, oraz **BSI2**, czego wynikiem były osiągane przez nie najwyższe wartości wskaźnika $GMean$. Przeprowadzony test statystyczny nie wykazał istotnych różnic pomiędzy opracowanymi metodami. Zastosowanie mechanizmów aktywnego uczenia w metodzie **BSI2** celem wyboru jedynie istotnych obserwacji nie tylko skróciło czas obliczeń, ale i spowodowało nieznaczny wzrost jakości klasyfikacji mierzonej wskaźnikiem $GMean$. Z kolei metoda **BSI1**, wykorzystująca mechanizmy selekcji jednostronnej osiągnęła wynik nieznacznie gorszy niż metody **BSI**, oraz **BSI2**, jednak charakteryzowała się wyższą wartością wskaźnika TP_{rate} i i wyższymi wartościami $GMean$ w przypadku danych wysoce niezbalansowanych.

Zbiór danych	**#Obs.**	**#Atr.**	**%P**	**%N**	$\mathbf{Imb_{rate}}$
Glass1	214	9	35.51	64.49	1.82
Ecoli0vs1	220	7	35.00	65.00	1.86
Wisconsin	683	9	35.00	65.00	1.86
Pima	768	8	34.84	66.16	1.90
Iris0	150	4	33.33	66.67	2.00
Glass0	214	9	32.71	67.29	2.06
Yeast1	1484	8	28.91	71.09	2.46
Vehicle1	846	18	28.37	71.63	2.52
Vehicle2	846	18	28.37	71.63	2.52
Vehicle3	846	18	28.37	71.63	2.52
Haberman	306	3	27.42	73.58	2.68
Glass0123vs456	214	9	23.83	76.17	3.19
Vehicle0	846	18	23.64	76.36	3.23
Ecoli1	336	7	22.92	77.08	3.36
New-thyroid2	215	5	16.89	83.11	4.92
New-thyroid1	215	5	16.28	83.72	5.14
Ecoli2	336	7	15.48	84.52	5.46
Segment0	2308	19	14.26	85.74	6.01
Glass6	214	9	13.55	86.45	6.38
Yeast3	1484	8	10.98	89.02	8.11
Ecoli3	336	7	10.88	89.77	8.77
Page-blocks0	5472	10	10.23	89.77	8.77
Yeast2vs4	514	8	9.92	90.08	9.08
Yeast05679vs4	528	8	9.66	90.34	9.35
Vowel0	988	13	9.01	90.99	10.10
Glass016vs2	192	9	8.89	91.11	10.29
Glass2	214	9	8.78	91.22	10.39
Ecoli4	336	7	6.74	93.26	13.84
Yeast1vs7	459	8	6.72	93.28	13.87
Shuttle0vs4	1829	9	6.72	93.28	13.87
Glass4	214	9	6.07	93.93	15.47
Page-blocks13vs2	472	10	5.93	94.07	15.85
Abalone9vs18	731	8	5.65	94.25	16.68
Glass016vs5	184	9	4.89	95.11	19.44
Shuttle2vs4	129	9	4.65	95.35	20.5
Yeast1458vs7	693	8	4.33	96.67	22.10
Glass5	214	9	4.20	95.80	22.81
Yeast2vs8	482	8	4.15	95.85	23.10
Yeast4	1484	8	3.43	96.57	28.41
Yeast1289vs7	947	8	3.17	96.83	30.56
Yeast5	1484	8	2.96	97.04	32.78
Ecoli0137vs26	281	7	2.49	97.51	39.15
Yeast6	1484	8	2.49	97.51	39.15
Abalone9	4174	8	0.77	99.23	128.87

Tabela 4.5: Charakterystyka zbiorów danych wykorzystanych w badaniach. *Źródło [49]*

Zbiór danych	SVM	SSVM	SBSVM	CSVM	AdaC	SBO	RUS	SB	UB	BSI	BSI1	BSI2
Glass1	0.0	55.7	69.3	71.4	78.9	**80.1**	78.2	75.2	76.5	74.2	64.0	71.8
Ecoli0vs1	**98.7**	98.3	83.3	97.0	97.0	97.0	97.7	98.3	98.0	98.3	98.0	98.0
Wisconsin	96.9	**97.6**	95.7	94.6	97.2	96.3	95.9	96.4	96.3	97.3	97.2	96.9
Pima	69.6	75.3	74.4	73.2	71.6	74.4	73.3	**76.1**	76.0	74.6	74.8	74.6
Iris0	**100.0**	**100.0**	**100.0**	**100.0**	99.0	99.0	99.0	98.0	99.0	**100.0**	**100.0**	**100.0**
Glass0	48.1	70.7	74.8	77.4	81.5	81.5	**85.6**	82.7	82.9	77.8	72.2	79.9
Yeast1	45.2	70.6	70.3	71.6	64.6	70.7	70.6	72.9	72.2	72.5	**73.4**	72.7
Vehicle1	54.1	79.0	82.7	83.0	79.5	74.4	74.0	77.1	77.6	84.1	81.6	**84.5**
Vehicle2	93.8	95.0	**98.4**	97.4	98.1	97.7	97.6	97.0	95.9	98.1	96.6	97.5
Vehicle3	39.1	76.7	81.7	82.1	76.7	73.9	77.5	75.6	79.0	82.0	**82.2**	82.1
Haberman	0.0	55.3	62.0	62.4	56.0	63.0	62.6	65.6	66.2	64.2	64.6	**66.4**
Glass0123vs456	88.3	89.4	89.3	89.3	92.3	90.3	91.0	92.3	90.5	91.4	92.7	**93.4**
Vehicle0	95.0	96.5	96.5	**97.8**	97.7	96.3	96.0	96.4	95.3	97.1	97.5	97.4
Ecoli1	82.8	89.7	88.9	88.0	89.1	87.8	**91.2**	90.3	90.4	90.1	88.4	89.5
New-thyroid2	79.3	98.9	97.1	97.7	95.7	96.9	95.5	96.6	94.9	98.0	**99.7**	**99.7**
New-thyroid1	77.5	98.6	98.0	99.4	94.6	98.3	97.7	97.5	96.6	99.2	**99.7**	**99.7**
Ecoli2	77.2	91.1	92.4	91.9	88.1	90.4	88.4	88.0	89.5	92.2	**93.9**	92.7
Segment0	99.1	99.3	99.4	99.5	98.2	99.6	99.1	99.3	98.9	**99.8**	99.7	99.5
Glass6	84.4	89.5	86.9	88.8	88.7	83.5	91.3	**92.1**	89.7	88.6	90.3	87.1
Yeast3	76.5	91.8	89.8	90.7	89.2	89.3	91.6	**94.1**	93.1	91.9	91.9	92.5
Ecoli3	41.1	89.4	86.7	83.8	82.2	81.5	87.1	86.9	89.0	89.0	**90.4**	89.5
Page-blocks0	65.5	95.4	96.3	96.0	**99.8**	99.7	97.0	99.0	97.0	97.8	97.4	99.4
Yeast2vs4	74.0	89.4	87.1	88.3	91.9	87.7	91.3	90.2	**95.4**	89.2	89.6	89.6
Yeast05679vs4	0.0	79.5	75.1	74.2	78.1	77.3	**84.4**	79.7	79.1	79.1	80.6	79.6
Vowel0	97.1	98.8	**100.0**	**100.0**	97.0	99.1	95.8	98.6	94.8	**100.0**	98.5	99.7
Glass016vs2	0.0	56.2	57.5	61.9	55.6	60.6	59.8	66.0	73.3	**76.7**	74.9	74.9
Glass2	0.0	57.1	57.6	78.0	71.9	76.9	70.4	**83.6**	77.0	81.2	79.8	81.3
Ecoli4	80.6	92.4	88.0	88.6	92.7	88.0	92.6	92.9	88.7	92.6	93.2	**93.4**
Shuttle0vs4	99.6	99.6	99.6	99.6	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	99.6	**100.0**	**100.0**
Yeast1vs7	0.0	75.1	54.3	69.1	70.1	63.2	73.5	65.2	74.5	**79.4**	77.2	77.4
Glass4	39.2	90.7	82.2	86.6	88.1	91.9	92.7	88.0	85.7	92.9	89.1	**94.6**
Page-blocks13vs2	70.2	90.6	90.2	93.4	79.7	93.4	95.0	95.6	**96.0**	93.4	92.9	93.7
Abalone9vs18	0.0	87.1	72.1	86.0	69.0	78.3	78.5	78.0	77.3	89.9	**90.9**	89.6
Glass016vs5	0.0	95.0	87.4	81.2	86.4	92.9	**98.9**	85.4	94.1	98.3	93.2	97.7
Shuttle2vs4	90.9	99.6	91.3	91.3	91.3	**100.0**	**100.0**	**100.0**	**100.0**	91.3	**100.0**	**100.0**
Yeast1458vs7	0.0	63.8	66.6	57.3	42.1	43.8	61.9	54.6	64.2	66.4	65.8	**66.9**
Glass5	0.0	94.2	74.4	81.3	97.3	98.3	86.7	92.0	94.7	**99.0**	95.3	92.9
Yeast2vs8	74.1	76.7	74.1	61.0	49.8	73.7	77.1	**79.7**	76.2	79.6	79.1	77.0
Yeast4	0.0	81.2	62.0	77.3	69.5	66.0	82.2	74.7	84.8	81.4	82.2	82.2
Yeast1289vs7	0.0	69.7	18.2	62.6	57.7	59.5	**74.5**	58.1	71.5	73.3	71.2	**74.5**
Yeast5	21.3	96.6	84.6	94.0	87.5	90.9	96.0	96.3	95.8	94.8	**97.7**	97.0
Ecoli0137vs26	84.2	87.5	**96.7**	74.6	81.5	83.0	81.2	83.1	75.4	83.9	90.7	89.7
Yeast6	0.0	87.6	71.3	86.4	67.8	80.2	83.7	82.4	87.0	88.9	89.6	**90.1**
Abalone9	0.0	68.4	17.6	61.2	17.5	17.6	68.5	38.7	69.0	76.6	**78.1**	75.2
ŚREDNIA	51.0	85.0	80.0	83.8	80.9	82.8	86.0	84.8	86.3	87.9	87.6	**88.2**

Tabela 4.6: Szczegółowe wyniki testów jakości badanych metod wyrażone wskaźnikiem $GMean$.

Zbiór danych	SVM	SSVM	SBSVM	CSVM	AdaC	SBO	RUS	SB	UB	BSI	BSI1	BSI2
Glass1	49.6	60.9	72.1	71.4	79.3	**80.1**	78.2	75.5	76.5	74.3	74.2	72.0
Ecoli0vs1	**98.7**	98.4	84.3	97.0	97.0	97.0	97.7	98.4	98.0	98.6	98.4	98.0
Wisconsin	96.9	**97.6**	95.7	94.6	97.2	96.3	95.9	96.4	96.3	97.2	97.3	96.9
Pima	71.7	75.4	75.0	73.3	71.9	74.6	73.3	76.1	**76.2**	75.4	74.6	74.6
Iris0	**100.0**	**100.0**	**100.0**	**100.0**	99.0	99.0	99.0	98.0	99.0	**100.0**	**100.0**	**100.0**
Glass0	59.7	74.6	75.9	77.5	81.8	81.6	**85.6**	82.8	83.2	77.6	77.8	80.0
Yeast1	58.8	70.7	71.2	71.7	67.5	71.5	71.0	73.2	72.3	**73.9**	72.5	72.8
Vehicle1	63.3	79.3	82.7	83.1	80.1	74.4	74.0	77.1	78.1	**87.0**	84.3	84.6
Vehicle2	93.8	95.1	98.4	97.4	98.1	97.7	97.6	97.0	95.9	**98.7**	98.1	97.5
Vehicle3	57.0	77.0	81.7	82.3	77.0	74.1	77.5	75.6	79.5	**84.0**	82.1	82.1
Haberman	49.8	61.5	62.1	63.5	56.1	63.2	63.4	65.8	66.5	**69.3**	64.9	66.7
Glass0123vs456	88.7	89.5	89.4	89.4	92.3	90.4	91.1	92.3	90.6	92.1	91.4	**93.4**
Vehicle0	95.1	96.5	96.5	**97.8**	97.7	96.3	96.0	96.4	95.3	96.7	97.1	97.4
Ecoli1	83.5	89.9	88.9	88.0	89.2	87.9	**91.2**	90.5	90.5	87.2	90.3	89.7
New-thyroid2	81.4	98.9	97.1	97.7	95.8	96.9	95.5	96.6	95.0	98.6	98.0	**99.7**
New-thyroid1	80.0	98.6	98.0	99.4	94.6	98.3	97.7	97.5	96.6	98.6	99.2	**99.7**
Ecoli2	79.2	91.1	92.5	91.9	88.2	90.5	88.4	88.2	89.5	**94.9**	92.3	92.7
Segment0	99.1	99.3	99.4	99.5	98.3	99.6	99.1	99.3	98.9	99.7	**99.8**	99.5
Glass6	85.4	89.8	87.7	89.4	88.9	84.3	91.3	92.1	89.7	**96.3**	89.1	87.5
Yeast3	79.0	91.8	90.0	90.7	89.3	89.5	91.6	**94.1**	93.1	92.8	91.9	92.5
Ecoli3	57.7	89.4	87.0	83.9	82.2	82.2	87.1	86.9	89.2	89.3	89.0	**89.5**
Page-blocks0	71.4	95.5	96.3	96.1	**99.8**	99.7	97.1	99.0	97.1	98.9	97.8	99.4
Yeast2vs4	77.3	89.5	87.6	88.5	92.0	88.0	91.3	90.2	**95.5**	91.6	89.2	89.8
Yeast05679vs4	50.0	79.5	77.3	75.3	78.1	78.5	**84.5**	80.1	79.1	81.3	79.1	79.6
Vowel0	97.2	98.8	**100.0**	**100.0**	97.1	99.1	95.8	98.6	94.8	**100.0**	**100.0**	99.7
Glass016vs2	50.0	60.3	64.5	67.2	56.4	65.2	61.5	67.6	73.4	**82.3**	77.0	77.1
Glass2	50.0	61.0	64.6	79.3	73.3	78.0	70.7	83.6	**78.5**	**85.5**	81.4	81.3
Ecoli4	82.5	92.5	88.4	89.1	92.8	88.4	92.6	92.9	88.7	**94.9**	92.6	93.4
Shuttle0vs4	99.6	99.6	99.6	99.6	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	99.9	99.6	**100.0**
Yeast1vs7	50.0	75.1	64.2	69.7	70.5	67.8	74.3	67.5	74.5	78.9	**79.4**	77.8
Glass4	57.7	90.7	83.4	87.2	88.2	91.9	92.7	88.1	86.0	93.5	92.9	**94.7**
Page-blocks13vs2	74.3	90.6	90.5	93.4	81.5	93.5	95.0	95.6	**96.0**	94.1	93.4	93.7
Abalone9vs18	50.0	87.1	75.3	86.3	72.2	79.7	78.5	78.8	77.3	89.2	**89.9**	89.6
Glass016vs5	50.0	95.1	88.0	82.8	86.9	93.0	**98.9**	85.7	94.3	96.7	98.3	97.7
Shuttle2vs4	91.3	99.6	91.7	91.7	91.7	**100.0**	**100.0**	**100.0**	**100.0**	99.2	91.7	**100.0**
Yeast1458vs7	50.0	63.9	**70.9**	59.6	54.3	58.0	63.3	61.4	64.2	69.3	66.4	67.3
Glass5	50.0	94.4	77.5	82.8	97.3	98.3	87.2	92.0	94.9	98.1	**99.0**	93.0
Yeast2vs8	77.4	79.0	77.4	66.5	62.2	76.9	78.2	81.4	77.2	**96.1**	81.2	78.1
Yeast4	50.0	81.4	68.6	77.9	72.1	70.8	82.2	76.3	85.2	**86.3**	81.6	82.3
Yeast1289vs7	50.0	69.7	51.4	65.3	63.8	66.5	75.0	64.4	71.7	**76.5**	73.3	74.7
Yeast5	52.3	96.7	85.6	94.1	88.2	91.2	96.0	96.3	95.8	96.3	94.8	**97.0**
Ecoli0137vs26	85.3	87.6	96.7	77.3	82.2	83.9	81.9	84.1	75.5	**97.9**	85.0	89.8
Yeast6	50.0	87.6	75.2	86.7	71.6	81.8	83.8	83.3	87.0	**92.0**	88.9	90.2
Abalone9	50.0	68.7	51.1	64.7	50.7	51.1	68.5	55.7	70.0	**78.2**	76.6	75.3
ŚREDNIA	70.3	85.7	83.0	84.6	82.8	84.7	86.2	85.7	86.5	**90.0**	88.0	88.4

Tabela 4.7: Szczegółowe wyniki testów jakości badanych metod wyrażone wskaźnikiem AUC.

Zbiór danych	SVM	SSVM	SBSVM	CSVM	AdaC	SBO	RUS	SB	UB	BSI	BSI1	BSI2
Glass1	64.0	53.7	66.4	71.5	77.1	**80.8**	78.0	77.6	76.2	74.3	63.6	73.4
Ecoli0vs1	**99.1**	98.6	96.4	97.3	96.8	96.8	97.7	98.6	98.2	98.6	98.2	98.2
Wisconsin	96.9	**97.4**	96.0	95.2	96.9	96.5	96.0	96.3	96.0	97.2	97.2	97.1
Pima	**77.0**	76.2	72.0	74.2	69.9	76.4	74.0	76.4	74.5	75.4	76.3	75.3
Iris0	**100.0**	**100.0**	**100.0**	**100.0**	99.3	99.3	99.3	98.7	99.3	**100.0**	**100.0**	**100.0**
Glass0	72.0	66.4	71.5	76.2	79.4	83.2	**85.0**	81.3	80.8	77.6	69.2	79.4
Yeast1	74.7	68.9	66.6	70.2	59.2	**76.1**	74.1	75.9	72.6	73.9	73.0	74.5
Vehicle1	79.3	75.8	83.0	85.5	75.5	75.7	73.5	77.0	73.9	**87.0**	83.2	86.5
Vehicle2	95.7	94.2	98.5	98.0	98.1	97.8	97.5	96.5	96.0	**98.7**	96.7	98.1
Vehicle3	77.7	73.5	81.6	**84.8**	73.5	77.0	78.0	74.5	74.9	84.0	82.0	83.8
Haberman	73.2	**74.2**	64.1	69.0	54.6	65.7	68.3	68.3	69.3	69.3	69.9	69.6
Glass0123vs456	93.0	91.1	92.1	92.1	**93.5**	92.5	92.5	**93.5**	89.7	92.1	93.0	93.0
Vehicle0	96.5	94.9	97.6	**98.2**	96.5	96.0	95.5	94.8	93.4	96.7	96.9	97.9
Ecoli1	89.3	86.6	89.3	87.8	86.9	**90.5**	89.3	88.1	87.5	87.2	84.2	86.9
New-thyroid2	94.0	98.1	99.1	98.1	96.7	96.7	96.3	96.3	93.5	98.6	**99.5**	**99.5**
New-thyroid1	93.5	97.7	98.6	99.1	94.9	97.2	98.1	97.7	96.3	98.6	**99.5**	**99.5**
Ecoli2	91.4	89.0	**95.2**	94.3	89.3	94.6	91.1	92.0	90.2	94.9	94.9	94.3
Segment0	99.7	**99.7**	99.7	**99.7**	99.2	**99.7**	99.2	99.4	98.8	99.7	99.7	99.7
Glass6	94.9	94.9	96.3	**96.7**	93.5	93.0	92.5	93.9	89.7	96.3	93.5	93.5
Yeast3	94.1	90.6	**95.1**	92.0	85.3	94.3	90.8	94.3	93.0	92.8	92.9	92.9
Ecoli3	89.9	87.8	**92.6**	86.9	79.5	90.8	86.0	87.8	85.1	89.3	92.0	90.2
Page-blocks0	96.6	91.5	99.4	98.9	**99.6**	99.4	94.5	98.1	94.5	98.9	98.3	98.9
Yeast2vs4	**95.3**	92.0	94.9	93.4	93.4	94.2	92.2	91.8	91.8	91.6	94.2	94.2
Yeast05679vs4	90.3	82.0	**92.2**	85.4	77.8	89.6	86.2	86.2	78.0	81.3	86.0	82.2
Vowel0	99.4	97.9	**100.0**	**100.0**	99.2	99.3	96.9	98.4	95.0	**100.0**	97.4	99.4
Glass016vs2	91.1	42.2	88.5	88.5	64.1	84.9	73.4	79.7	70.8	82.3	78.6	**92.2**
Glass2	**92.1**	43.0	89.3	91.6	85.5	89.3	75.7	84.6	65.4	85.5	82.7	80.4
Ecoli4	**97.9**	94.6	95.8	97.0	95.2	95.8	94.9	95.5	87.5	94.9	96.1	96.4
Shuttle0vs4	99.9	99.9	99.9	99.9	99.9	**100.0**	**100.0**	99.9	**100.0**	99.9	99.9	99.9
Yeast1vs7	93.5	76.7	**96.2**	78.2	76.7	89.1	83.9	82.8	75.6	78.9	80.8	84.5
Glass4	94.9	89.3	95.8	96.3	84.6	91.6	93.0	91.1	80.4	93.5	86.4	**96.7**
Page-blocks13vs2	93.9	92.1	**96.4**	94.4	67.7	95.7	95.1	95.0	95.2	94.1	93.4	94.1
Abalone9vs18	94.1	84.0	**94.5**	92.7	90.7	92.5	80.0	88.8	75.8	89.2	91.1	90.7
Glass016vs5	95.1	90.8	97.3	97.3	95.1	96.7	**97.8**	92.9	89.1	96.7	97.3	95.7
Shuttle2vs4	98.4	99.2	99.2	99.2	99.2	**100.0**	**100.0**	**100.0**	**100.0**	99.2	**100.0**	**100.0**
Yeast1458vs7	**95.7**	61.3	91.9	74.5	85.6	92.8	75.5	87.0	65.1	69.3	68.1	73.9
Glass5	95.8	89.3	97.7	97.7	94.9	96.7	95.8	94.9	90.2	**98.1**	91.1	96.7
Yeast2vs8	**97.9**	96.5	**97.9**	90.9	96.3	96.9	90.2	96.5	88.4	96.1	95.0	90.0
Yeast4	**96.6**	86.0	95.9	86.5	90.0	94.8	82.0	90.8	77.0	86.3	85.9	85.9
Yeast1289vs7	**96.8**	69.5	95.2	82.7	89.1	94.5	82.8	90.3	76.3	76.5	72.4	79.1
Yeast5	97.2	93.5	97.7	97.0	**98.5**	97.8	94.4	97.1	93.9	96.3	95.6	96.3
Ecoli0137vs26	**98.6**	89.3	96.8	96.4	92.5	95.7	91.8	96.1	79.4	97.9	95.7	93.6
Yeast6	97.6	89.5	**97.8**	93.0	93.6	97.1	87.5	94.6	88.2	92.0	93.4	94.5
Abalone9	**99.2**	74.8	98.3	85.3	97.6	98.3	71.4	95.1	58.8	78.2	78.1	78.7
ŚREDNIA	92.1	84.9	**92.3**	90.5	87.8	92.1	88.4	90.6	85.1	90.0	88.9	90.4

Tabela 4.8: Szczegółowe wyniki testów jakości badanych metod wyrażone poprawnością klasyfikacji.

Zbiór danych	SVM	SSVM	SBSVM	CSVM	AdaC	SBO	RUS	SB	UB	BSI	BSI1	BSI2
Glass1	0.0	85.5	**92.1**	71.1	86.8	77.6	78.9	68.4	77.6	73.7	65.8	67.1
Ecoli0vs1	**100.0**	99.3	71.4	97.9	96.5	96.5	97.9	99.3	98.6	99.3	98.6	98.6
Wisconsin	96.7	**98.3**	94.6	92.9	**98.3**	95.8	95.4	96.7	97.1	97.5	97.1	96.2
Pima	54.5	72.8	**85.1**	70.1	78.4	68.7	71.3	75.0	82.1	72.0	70.5	72.4
Iris0	**100.0**	**100.0**	**100.0**	**100.0**	98.0	98.0	98.0	96.0	98.0	**100.0**	**100.0**	**100.0**
Glass0	24.3	**98.6**	88.6	81.4	88.6	77.1	87.1	87.1	90.0	78.6	84.3	81.4
Yeast1	21.2	75.1	82.1	75.3	**87.2**	60.6	63.6	66.9	71.3	69.2	74.4	69.0
Vehicle1	30.4	86.6	82.0	78.3	**89.4**	71.9	75.1	77.4	86.6	78.8	78.3	80.6
Vehicle2	89.9	96.8	**98.2**	96.3	**98.2**	97.7	97.7	**98.2**	95.9	96.8	96.3	96.3
Vehicle3	15.6	84.0	82.1	77.4	84.0	68.4	76.4	77.8	**88.7**	78.3	82.5	78.8
Haberman	0.0	34.6	58.0	51.9	59.3	58.0	53.1	**60.5**	**60.5**	55.6	55.6	**60.5**
Glass0123vs456	80.4	86.3	84.3	84.3	90.2	86.3	88.2	90.2	92.2	90.2	92.2	**94.1**
Vehicle0	92.5	99.5	94.5	97.0	**100.0**	97.0	97.0	99.5	99.0	98.0	98.5	96.5
Ecoli1	72.7	96.1	88.3	88.3	93.5	83.1	94.8	94.8	96.1	96.1	**97.4**	94.8
New-thyroid2	62.9	**100.0**	94.3	97.1	94.3	97.1	94.3	97.1	97.1	97.1	**100.0**	**100.0**
New-thyroid1	60.0	**100.0**	97.1	**100.0**	94.3	**100.0**	97.1	97.1	97.1	**100.0**	**100.0**	**100.0**
Ecoli2	61.5	**94.2**	88.5	88.5	86.5	84.6	84.6	82.7	88.5	88.5	92.3	90.4
Segment0	98.2	98.8	99.1	99.1	97.0	99.4	99.1	99.1	99.1	**100.0**	99.7	99.4
Glass6	72.4	82.8	75.9	79.3	82.8	72.4	**89.7**	**89.7**	**89.7**	79.3	86.2	79.3
Yeast3	59.5	93.3	83.4	89.0	**94.5**	83.4	92.6	93.9	93.3	90.8	90.8	92.0
Ecoli3	17.1	91.4	80.0	80.0	85.7	71.4	88.6	85.7	**94.3**	88.6	88.6	88.6
Page-blocks0	42.9	**100.0**	92.9	92.9	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	96.4	96.4	**100.0**
Yeast2vs4	54.9	86.3	78.4	82.4	90.2	80.4	90.2	88.2	**100.0**	86.3	84.3	84.3
Yeast05679vs4	0.0	76.5	58.8	62.7	78.4	64.7	**82.4**	72.5	80.4	76.5	74.5	76.5
Vowel0	94.4	**100.0**	**100.0**	**100.0**	94.4	98.9	94.4	98.9	94.4	**100.0**	**100.0**	**100.0**
Glass016vs2	0.0	**82.4**	35.3	41.2	47.1	41.2	47.1	52.9	76.5	70.6	70.6	58.8
Glass2	0.0	82.4	35.3	64.7	58.8	64.7	64.7	82.4	**94.1**	76.5	76.5	82.4
Ecoli4	65.0	90.0	80.0	80.0	**90.0**	80.0	**90.0**	**90.0**	**90.0**	**90.0**	**90.0**	**90.0**
Shuttle0vs4	99.2	99.2	99.2	99.2	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	99.2	**100.0**	**100.0**
Yeast1vs7	0.0	73.3	30.0	60.0	63.3	43.3	63.3	50.0	73.3	**80.0**	73.3	70.0
Glass4	15.4	92.3	69.2	76.9	**92.3**	**92.3**	92.3	84.6	**92.3**	**92.3**	**92.3**	**92.3**
Page-blocks13vs2	49.7	88.7	83.0	92.3	**98.9**	90.7	94.8	96.4	97.0	92.5	92.3	93.2
Abalone9vs18	0.0	90.7	53.5	79.1	51.2	65.1	76.7	67.4	79.1	**90.7**	**90.7**	88.4
Glass016vs5	0.0	**100.0**	77.8	66.7	77.8	88.9	**100.0**	77.8	**100.0**	**100.0**	88.9	**100.0**
Shuttle2vs4	83.3	**100.0**	83.3	83.3	83.3	**100.0**	**100.0**	**100.0**	**100.0**	83.3	**100.0**	**100.0**
Yeast1458vs7	0.0	**66.7**	46.7	43.3	20.0	20.0	50.0	33.3	63.3	63.3	63.3	60.0
Glass5	0.0	**100.0**	55.6	66.7	**100.0**	**100.0**	77.8	88.9	**100.0**	**100.0**	**100.0**	88.9
Yeast2vs8	55.0	60.0	55.0	40.0	25.0	55.0	**65.0**	**65.0**	**65.0**	**65.0**	**65.0**	**65.0**
Yeast4	0.0	76.5	39.2	68.6	52.9	45.1	82.4	60.8	**94.1**	76.5	78.4	78.4
Yeast1289vs7	0.0	**70.0**	3.3	46.7	36.7	36.7	66.7	36.7	66.7	**70.0**	**70.0**	**70.0**
Yeast5	4.5	**100.0**	72.7	90.9	77.3	84.1	97.7	95.5	97.7	93.2	**100.0**	97.7
Ecoli0137vs26	71.4	85.7	**97.2**	57.1	71.4	71.4	71.4	71.4	71.4	71.4	85.7	85.7
Yeast6	0.0	**85.7**	51.4	80.0	48.6	65.7	80.0	71.4	**85.7**	**85.7**	**85.7**	**85.7**
Abalone9	0.0	62.5	3.1	43.8	3.1	3.1	65.6	15.6	**81.3**	75.0	78.1	71.9
ŚREDNIA	41.9	87.3	73.2	77.6	78.3	75.8	83.5	80.3	**88.5**	85.5	86.5	85.8

Tabela 4.9: Szczegółowe wyniki testów jakości badanych metod wyrażone wskaźnikiem TP_{rate}.

Zbiór danych	SVM	SSVM	SBSVM	CSVM	AdaC	SBO	RUS	SB	UB	BSI	BSI1	BSI2
Glass1	**99.3**	36.2	52.2	71.7	71.7	82.6	77.5	82.6	75.4	74.6	62.3	76.8
Ecoli0vs1	**97.4**	**97.4**	97.1	96.1	**97.4**	**97.4**	**97.4**	**97.4**	**97.4**	**97.4**	**97.4**	**97.4**
Wisconsin	97.1	96.8	96.8	96.4	96.2	96.8	96.4	96.2	95.5	97.1	97.3	**97.5**
Pima	**89.0**	78.0	65.0	76.4	65.4	80.6	75.4	77.2	70.4	77.2	79.4	76.8
Iris0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
Glass0	**95.1**	50.7	63.2	73.6	75.0	86.1	84.0	78.5	76.4	77.1	61.8	78.5
Yeast1	**96.4**	66.4	60.3	68.2	47.9	82.5	78.3	79.5	73.2	75.8	72.5	76.7
Vehicle1	**96.2**	72.0	83.3	87.9	70.7	76.9	73.0	76.8	69.5	89.8	84.9	88.6
Vehicle2	97.8	93.3	98.6	98.6	98.1	97.8	97.5	95.9	96.0	**99.4**	96.8	98.7
Vehicle3	**98.4**	70.0	81.4	87.2	70.0	79.8	78.5	73.3	70.3	86.0	81.9	85.5
Haberman	**99.6**	88.4	66.2	75.1	52.9	68.4	73.8	71.1	72.4	74.2	75.1	72.9
Glass0123vs456	**96.9**	92.6	94.5	94.5	94.5	94.5	93.9	94.5	89.0	92.6	93.3	92.6
Vehicle0	97.7	93.5	98.6	**98.6**	95.4	95.7	95.1	93.4	91.7	96.3	96.4	98.3
Ecoli1	**94.2**	83.8	89.6	87.6	84.9	92.7	87.6	86.1	84.9	84.6	80.3	84.6
New-thyroid2	**100.0**	97.8	**100.0**	98.3	97.2	96.7	96.7	96.1	92.8	98.9	99.4	99.4
New-thyroid1	**100.0**	97.2	98.9	98.9	95.0	96.7	98.3	97.8	96.1	98.3	99.4	99.4
Ecoli2	**96.8**	88.0	96.5	95.4	89.8	96.5	92.3	93.7	90.5	96.1	95.4	95.1
Segment0	**99.9**	**99.9**	99.8	99.8	99.5	99.8	99.2	99.5	98.7	99.7	99.7	99.7
Glass6	98.4	96.8	**99.5**	**99.5**	95.1	96.2	93.0	94.6	89.7	98.9	94.6	95.7
Yeast3	**98.4**	90.3	96.6	92.4	84.2	95.6	90.6	94.4	93.0	93.0	93.1	93.0
Ecoli3	**98.3**	87.4	94.0	87.7	78.7	93.0	85.7	88.0	84.1	89.4	92.4	90.4
Page-blocks0	**100.0**	91.0	99.8	99.3	99.5	99.3	94.1	98.0	94.1	99.1	98.4	98.9
Yeast2vs4	**99.8**	92.7	96.8	94.6	93.7	95.7	92.4	92.2	90.9	92.2	95.2	95.2
Yeast05679vs4	**100.0**	82.6	95.8	87.8	77.8	92.2	86.6	87.6	77.8	81.8	87.2	82.8
Vowel0	99.9	97.7	**100.0**	**100.0**	99.7	99.3	97.1	98.3	95.1	100.0	97.1	99.3
Glass016vs2	**100.0**	38.3	93.7	93.1	65.7	89.1	76.0	82.3	70.3	83.4	79.4	95.4
Glass2	**100.0**	39.6	93.9	93.9	87.8	91.4	76.6	84.8	62.9	86.3	83.2	80.2
Ecoli4	**100.0**	94.9	96.8	98.1	95.6	96.8	95.3	95.9	87.3	95.3	96.5	96.8
Shuttle0vs4	**100.0**	**100.0**	99.9	99.9	99.9	**100.0**	**100.0**	99.9	**100.0**	**100.0**	99.9	99.9
Yeast1vs7	**100.0**	76.9	98.4	79.5	77.6	92.3	85.3	85.1	75.8	78.8	81.4	85.5
Glass4	**100.0**	89.1	97.5	97.5	84.1	91.5	93.0	91.5	79.6	93.5	86.1	97.0
Page-blocks13vs2	**99.0**	92.4	97.9	94.6	64.2	96.2	95.2	94.9	95.0	94.3	93.6	94.2
Abalone9vs18	**100.0**	83.6	97.1	93.6	93.2	94.2	80.2	90.1	75.6	89.1	91.1	90.8
Glass016vs5	**100.0**	90.3	98.3	98.9	96.0	97.1	97.7	93.7	88.6	96.6	97.7	95.4
Shuttle2vs4	99.2	99.2	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
Yeast1458vs7	**100.0**	61.1	95.1	75.9	88.5	96.1	76.6	89.4	65.2	69.5	68.3	74.5
Glass5	**100.0**	88.8	99.5	99.0	94.6	96.6	96.6	95.1	89.8	98.0	90.7	97.1
Yeast2vs8	**99.8**	98.1	**99.8**	93.1	99.4	98.7	91.3	97.8	89.4	97.4	96.3	91.1
Yeast4	**100.0**	86.3	97.9	87.1	91.3	96.6	82.0	91.9	76.3	86.7	86.2	86.2
Yeast1289vs7	**100.0**	69.5	99.4	83.9	90.8	96.4	83.3	92.0	76.7	76.7	72.5	79.4
Yeast5	**100.0**	93.3	98.5	97.2	99.2	98.3	94.3	97.2	93.8	96.4	95.5	96.3
Ecoli0137vs26	**99.3**	89.4	96.1	97.4	93.1	96.4	92.3	96.7	79.6	98.5	96.0	93.8
Yeast6	**100.0**	89.6	98.9	93.3	94.7	97.9	87.6	95.2	88.3	92.1	93.6	94.7
Abalone9	**100.0**	74.9	99.1	85.6	98.4	99.0	71.4	95.7	58.7	78.2	78.1	78.7
ŚREDNIA	**98.7**	84.0	92.8	91.5	87.4	93.6	88.8	91.2	84.5	90.5	89.0	90.9

Tabela 4.10: Szczegółowe wyniki testów jakości badanych metod wyrażone wskaźnikiem TN_{rate}.

Rozdział 5

Zastosowanie metod w diagnostyce medycznej

Opracowane w ramach rozprawy metody klasyfikacji mogą być z powodzeniem stosowane do zadań klasyfikacyjnych z różnych dziedzin charakteryzujących się niezbalansowaniem danych. Niniejszy rozdział pokazuje, w jaki sposób problem podejmowania decyzji w obszarze analizy ryzyka operacyjnego może być rozwiązany z wykorzystaniem proponowanych w rozprawie metod klasyfikacji. Zakres prac dotyczących wymienionego zastosowania obejmuje:

- opracowanie algorytmu ekstrakcji reguł z proponowanych w rozprawie wzmacnianych algorytmów SVM,
- analizę jakości opracowanych metod na rzeczywistym zbiorze danych dotyczącym predykcji przeżywalności pooperacyjnej,
- analizę jakości reguł otrzymanych w procesie ekstrakcji,
- analizę jakości technik wstawiania brakujących wartości atrybutów przeprowadzoną dla metody *BoostingSVM-IB*.

5.1 Cel badań

W niniejszym rozdziale opisano wyniki badań dotyczące zastosowania opracowanych w ramach rozprawy algorytmów *BoostingSVM-IB*, *BoostingSVM-IB.M1* oraz *BoostingSVM-IB.M2* do problemu predykcji przeżywalności po operacji raka płuc, oraz zaproponowano metodę ekstrakcji reguł decyzyjnych, która w wyniku zastosowania modeluje zachowanie opracowanych metod klasyfikacji.

Obszar diagnostyki medycznej, ze względu na wysokie ryzyko błędnych decyzji oraz nieufności specjalistów wobec trudno zrozumiałych modeli decyzyjnych, jest obszarem niskiej stosowalności tego typu rozwiązań. Pomimo wyższej skuteczności trudno interpretowalnych modeli, takich jak opracowane w ramach rozprawy klasyfikatory SVM, konieczne jest zaproponowanie rozwiązań, które pozwolą skonstruować na bazie tzn. modelu „czarnej skrzynki" (*ang. black box models*) klasyfikator charakteryzujący się wysokim stopniem interpretowalności (*ang. interpretability*) (reguły i drzewa decyzyjne) i zadowalającą jakością klasyfikacji.

W rozdziale tym rozważa się problem predykcji przeżywalności pooperacyjnej wyrażony jako dychotomiczne zadanie klasyfikacji, w którym poszukiwana jest odpowiedź na pytanie czy pacjent przeżyje zakładany w badaniu okres 1 roku po operacji. Jakość klasycznych metod indukcji reguł stosowanych do rozwiązania tego problemu jest niezadowalająca, gdyż zagadnienie to charakteryzuje się silnym niebalansowaniem. Liczba pacjentów, którzy przeżyją rozpatrywany okres po operacji, jest znacząco wyższa niż liczba zgonów w przeciągu roku od obserwacji, co powoduje, że generowane reguły są silnie obciążone w kierunku klasy dominującej. W poprzednim rozdziale wykonane zostały badania empiryczne które potwierdzają wysoką skuteczność opracowanych w ramach rozprawy metod klasyfikacji dla problemów niezbalansowanych, dlatego proponuje się ich zastosowanie do rozpatrywanego zadania analizy ryzyka operacyjnego. Ze względu na fakt, że algorytmy te działają na zasadzie „czarnych skrzynek" i zrozumienie natury ich działania jest zadaniem trudnym, konieczne było opracowanie algorytmu indukcji reguł zachowującego wysoką jakość klasyfikacji.

5.2 Opis problemu predykcji pooperacyjnej i stosowanych metod

Jednym z typowych problemów decyzyjnych w dziedzinie torakochirurgii jest wybór pacjentów do operacji klatki piersiowej, w przypadku których przeprowadzony zabieg spowoduje rzeczywistą poprawę stanu zdrowia i wydłużenie czasu życia pacjenta. Dla każdego z pacjentów kluczowe jest więc określenie, czy dany pacjent będzie w stanie przeżyć zadany w analizie krytyczny okres czasu, który obejmuje 1 rok po operacji.

Do rozwiązania postawionego wyżej problemu stosuje się szereg klasycznych metod modelowania statystycznego, wykorzystujących krzywe Kaplana-Meiera, hierarchiczne modele statystyczne, regresję logistyczną, czy też regresję Coxa [3, 65, 115, 116]. Inne typowe podejścia wykorzystują modele postaci regresji logistycznej do konstrukcji *tablic scoringowych* (*ang. scoring tables*) [6, 41, 107]. Do rozwiązywania torakochirurgicznych problemów decyzyjnych coraz powszechniej stosowane są metody uczenia maszynowego, takie jak drzewa decyzyjne [35, 43], czy też sieci neuronowe [40, 109].

W większości z wymienionych prac nie rozpatruje się wpływu jakości dostępnych danych źródłowych na wyniki predykcji przeżywalności pooperacyjnej, szczególnie pomijając problemy brakujących wartości atrybutów [50] i poruszany w rozprawie problem niezbalansowania zbioru uczącego. Jedną z nielicznych prac dotyczących ryzyka operacyjnego i uwzględniających braki danych jest pozycja [43], w której proponuje się zastosowanie wielokrotnej imputacji opartej na algorytmie CART. W ramach prac [148, 151] przeprowadzono analizę podstawowych metod wstawiania niekompletnych wartości atrybutów w zadaniu predykcji przeżywalności pooperacyjnej. Otrzymane w pracy wyniki wykazały wysoką jakości technik imputacji wykorzystujących klasyfikator Naiwnego Bayesa i algorytm K-NN. Problem przeżywalności pooperacyjnej w kontekście danych niezbalansowanych był rozważany jedynie w pracy [84], w której autorzy zastosowali szereg podejść wykorzystujących zespoły klasyfikatorów dla danych o nierównym rozkładzie klas.

5.3 Indukcja reguł z modelu „czarnej skrzynki”

Z przeprowadzonych wcześniej rozważań wynika, że w zadaniu predykcji przeżywalności pooperacyjnej zachodzi konieczność zrealizowania dwóch zadań:

1. Konstrukcji modelu decyzyjnego (klasyfikatora) charakteryzującego się wysoką jakością klasyfikacji, i dużą odpornością na niedoskonałość danych wykorzystywanych w procesie uczenia.

2. Odkrycia i interpretacji wiedzy która wykorzystywana jest w procesie klasyfikacji celem zrozumienia natury zjawiska.

Pierwsze z zadań zrealizowano poprzez zaproponowanie algorytmów *BoostingSVM-IB*, *BoostingSVM-IB.M1* oraz *BoostingSVM-IB.M2*, których wysoka jakość i odporność na niezbalansowanie danych została wykazana w poprzednim rozdziale. Drugie z zadań realizowane jest poprzez klasyfikatory charakteryzujące się zrozumiałą reprezentacją wiedzy, takie jak opisane w drugim rozdziale rozprawy drzewa i reguły decyzyjne. Wadą klasycznych metod konstruowania modeli o zrozumiałej reprezentacji wiedzy jest ich niska jakość klasyfikacji i niewielka odporność na złą jakość danych, dlatego zachodzi konieczność wykorzystania modeli wysokiej jakości predykcyjnej do konstrukcji modeli o wysokim stopniu interpretowalności.

W literaturze wyróżnia się trzy główne podejścia do problemu ekstrakcji reguł bezpośrednio z charakteryzujących się wysoką skutecznością predykcji nieinterpretowalnych modeli [27, 129]. Pierwsze z nich, nazywane dekompozycyjnym (*ang. decompositional*), dokonuje indukcji reguł poprzez analizę struktury wyuczonego modelu, np. sieci neuronowej [28] lub marginesu separującego wyznaczonego przez SVM [94]. Drugie z podejść, nazywane pedagogicznym (*ang. pedagogical*), nie wymaga analizy struktury modelu „czarnej skrzynki”, działa niezależnie od wykorzystywanego klasyfikatora i polega na wygenerowaniu nowej porcji danych i zaetykietowaniu ich, bądź też modyfikacji etykiet obserwacji ze zbioru uczącego poprzez zastosowanie wysokiej jakości metody klasyfikacji wyuczonej na pierwotnym zbiorze treningowym. Zmodyfikowany zbiór danych jest następnie wykorzystywany do konstrukcji drzewa decyzyjnego, bądź reguł decyzyjnych poprzez zastosowanie klasycznych metod, np. *RIPPER*, bądź *C4.5* [30]. Wszystkie pozostałe metody, które łączą w sobie wymienione techniki, nazywane są podejściami eklektycznymi (*ang. eclectic*) [5].

Opracowane w ramach rozprawy metody łączą w sobie dwa trudne do interpretacji modele – klasyfikator wzmacniany i model typu SVM. Zastosowanie podejść analizujących strukturę klasyfikatora jest więc zadaniem trudnym, gdyż wymaga analizy zarówno struktury zewnętrznej (zespół modeli), jak i wewnętrznej (klasyfikatorów bazowych typu *SVM*). Z tego względu w rozprawie proponuje się zastosowanie podejścia typu pedagogicznego, polegającego na reetykizacji zbioru uczącego z wykorzystaniem wyuczonego klasyfikatora. Podejście to, w literaturze nazywane wyrocznią (*ang. oracle-based approach*) [27], zostało opisane Algorytmem 6.

Algorithm 6: Indukcja reguł z modelu *BoostingSVM-IB*.

Input : Zbiór uczący $\mathbb{S}_N$, klasyfikator *BoostingSVM-IB* $\Psi(\mathbf{x})$

Output: Zbiór reguł decyzyjnych $\mathcal{R}$

$\tilde{\mathbb{S}}_N \longleftarrow \emptyset$;

Wyucz $\Psi(\mathbf{x})$ na zbiorze $\mathbb{S}_N$ wykorzystując optymalne parametry uczenia;

foreach $(\mathbf{x}_n, y_n) \in \mathbb{S}_N$ **do**

  $\tilde{y}_n \longleftarrow \Psi(\mathbf{x}_n)$;

  $\tilde{\mathbb{S}}_N \longleftarrow \tilde{\mathbb{S}}_N \cup \{(\mathbf{x}_n, \tilde{y}_n)\}$;

end

Utwórz zbiór $\mathcal{R}$ wykorzystując $\tilde{\mathcal{S}}$ w procesie indukcji reguł;

W pierwszym kroku Algorytmu 6 następuje wyuczenia klasyfikatora *BoostingSVM-IB* $\Psi(\mathbf{x})$ na wejściowym zestawie danych $\mathbb{S}_N$.[1] Krok ten powinien być poprzedzony dobraniem odpowiednich wartości parametrów uczenia dla metody, z wykorzystaniem np. walidacji krzyżowej. W dalszej kolejności klasyfikator $\Psi(\mathbf{x})$ każdej z obserwacji $\mathbf{x}_n$ znajdujących się w zbiorze uczącym $\mathbb{S}_N$ przypisuje nową etykietę $\tilde{y}_n$ będącą wynikiem procesu klasyfikacji. W ostatnim kroku zmodyfikowany poprzez nadanie nowych etykiet zbiór uczący $\tilde{\mathbb{S}}_N$ zostaje wykorzystany do wyznaczenia zestawu reguł decyzyjnych, np. poprzez zastosowanie algorytmu RIPPER [29].

[1] Proponowany algorytm jest niewrażliwy na wybór metody klasyfikacji, dlatego możliwe jest zamienne wykorzystanie innego klasyfikatora charakteryzującego się wysoką skutecznością predykcji (tzn. klasyfikatora silnego, *ang. strong learner*).

5.4 Charakterystyka zbioru danych

W opisanych w tym rozdziale badaniach dotyczących modelowania ryzyka operacyjnego wykorzystano rzeczywiste dane kliniczne o 1203 pacjentach leczonych operacyjne z powodu raka płuc we Wrocławskim Ośrodku Torakochirurgii (WTO) w latach 2007-2011. Wejściowy zbiór danych do analizy zawierał 37 atrybutów charakteryzujących pacjenta przed operacją. W ramach rozprawy skoncentrowano się na binarnym zadaniu decyzyjnym, w ramach którego należało określić, czy pacjent przeżyje 1 rok po operacji, czy też nastąpi zgon w zadanym okresie.

5.5 Selekcja cech i czyszczenie danych

Wejściowy zestaw cech został poddany procesowi selekcji celem zidentyfikowania tych atrybutów, które mają istotny wpływ na klasę. Zgodnie z wnioskami opisanymi w pracy [79] rekomenduje się przeprowadzenie procesu selekcji cech dla silnie niezbalansowanych problemów podejmowania decyzji w dziedzinie medycyny. W niniejszej pracy rozważamy zastosowanie kryterium zysku informacyjnego (*ang. info gain*) [144]. W ramach tej metody dla każdego z atrybutów niezależnie liczona jest wartość kryterium postaci:

$$InfoGain(\mathbf{X}, \mathbf{Y}) = Entr(\mathbf{Y}) - Entr(\mathbf{Y}|\mathbf{X}), \tag{5.1}$$

gdzie $Entr(\mathbf{Y})$ reprezentuje entropię liczoną względem zmiennej losowej $\mathbf{Y}$ reprezentującej klasę:

$$Entr(\mathbf{Y}) = -\sum_{y \in \mathbb{Y}} p(y) \log p(y), \tag{5.2}$$

natomiast $Entr(\mathbf{Y}|\mathbf{X})$ entropię warunkowaną, która zadaną wzorem:

$$Entr(\mathbf{Y}|\mathbf{X}) = \sum_{x \in \mathbb{X}} p(x) Entr(\mathbf{Y}|x). \tag{5.3}$$

W procesie selekcji cech z wykorzystaniem kryterium informacyjności wybierane są te atrybuty, dla których wartość $InfoGain(\mathbf{X}, \mathbf{Y})$ jest niższa od zadanej wartości progowej.

ID	Opis	InfoGain
PRE14	wielkość pierwotnego guza (OC11 (najmniejsza) – OC14 (największa))	0.029
DGN	diagnoza (kombinacja kodów diagnostycznych ICD-10 dotyczących guza)	0.013
PRE4	natężona pojemność życiowa (*ang. forced vital capacity*, FVC)	0.008
PRE7	ból (przed operacją)	0.008
AGE	wiek podczas operacji	0.008
PRE6	stan sprawności (skala Zubroda)	0.007
PRE11	osłabienie (przed operacją)	0.004
PRE9	duszności (przed operacją)	0.004
PRE10	kaszel (przed operacją)	0.003
PRE8	krwioplucie (przed operacją)	0.003
PRE25	choroba tętnic obwodowych (*ang. peripheral artery disease*, PAD)	0.003
PRE19	wartość MI do 6 miesięcy	0.003
PRE5	Natężona objętość wydechowa jednosekundowa (FEV1)	0.002
PRE32	astma	0.002
PRE30	palący	0.002
PRE17	cukrzyca typu 2	0.002
Risk1Y	Ryzyko zgonu w przeciągu roku (*(T)rue*, jeśli nastąpił zgon, *(F)alse* w przeciwnym wypadku)	

Tabela 5.1: Charakterystyka cech wybranych w procesie selekcji.

Wybór metody selekcji cech dla zadanego problemu decyzyjnego podyktowany jest przede wszystkim zrozumiałym dla specjalistów medycznych mechanizmem działania.

W wyniku zastosowania kryterium *InfoGain* zredukowano liczbę atrybutów z 36 do 16, przyjmując graniczną wartość kryterium równą 0.001. Wybrane atrybuty uszeregowane względem rozpatrywanego wskaźnika jakości zostały opisane w Tabeli 5.1. Atrybuty *PRE4*, *PRE5* and *AGE* są atrybutami numerycznymi, atrybuty *PRE14*, *DGN* and *PRE6* przyjmują cechy nominalne, natomiast reszta z atrybutów przyjmuje wartość binarne. [2]

Następnie usunięto wszystkie rekordy, w przypadku których zaobserwowano brakujące wartości atrybutów. Ostatecznie, zbiór danych wykorzystany do eksperymentu składał się z 470 rekordów ze wskaźnikiem niezbalansowania równym 5.71.

5.6 Badania empiryczne

Bazując na wynikach eksperymentu przeprowadzonego na zestawie benchmarkowych zbiorów danych (Rozdział 4) w przypadku zadania predykcji przeżywalności pooperacyjnej w analizie jakości działania wzięto pod uwagę jedynie metody **UB**, **RUS**, **SSVM**, **BSI**, **BS1**

[2] T(rue) jeżeli symptom kliniczny występuje, (F)alse w przeciwnym wypadku.

i **BS2**. W ramach metodyki eksperymentu, podobnie jak w przypadku badań opisanych w poprzednim rozdziale wykorzystano walidację krzyżową. Wyniki eksperymentu zostały zamieszczone w Tabeli 5.2. W analizie wzięto pod uwagę kryteria oceny takie jak TP_{rate}, TN_{rate}, poprawność klasyfikacji (Acc), oraz $GMean$.

Analizując otrzymane wyniki ze względu na kryterium $GMean$ metoda **BSI** osiągnęła nieznacznie wyższą skuteczność niż inne algorytmy rozpatrywane w badaniu. Porównywalna wartość wskaźnika została również zaobserwowana dla metody **UB**, jednak ta złożona technika klasyfikacji wykorzystująca mechanizm eliminacji losowej ma tendencję do zbytniego faworyzowania klasy zdominowanej, co w praktyce skutkuje znacznym obniżeniem poprawności klasyfikacji. W przypadku modyfikacji **BS1** oraz **BS2** zaobserwowano niższe wartości kryterium $GMean$ niż dla **BSI**. Wyniki przedstawione w Tabeli 5.2, poparte wynikami badań z Rozdziału 3. pozwalają stwierdzić, że najlepszym modelem „czarnej skrzynki" do indukcji reguł okazuje się być metoda **BSI**.

Metoda	$\mathbf{TP_{rate}}$	$\mathbf{TN_{rate}}$	**Acc**	**GMean**
UB	**68.57**	61.75	62.77	65.07
RUS	52.85	65.50	63.62	58.84
SSVM	57.14	68.25	66.60	62.45
BSI	60.00	72.00	70.21	**65.73**
BSI1	55.71	72.75	70.21	63.66
BSI2	54.29	68.75	65.96	60.76
JRip	0.00	**100.00**	**85.11**	0.00
JRip + BSI	60.00	70.00	68.51	64.81

Tabela 5.2: Wyniki dla zbioru danych dotyczącego przeżywalności pooperacyjnej.

Ja wcześniej wspomniano w pracy **BSI** jest trudnym do interpretacji klasyfikatorem, ze względu na to, że łączy ze sobą dwa modele: SVM i klasyfikatory wzmacniane. W pracy, dla zadanego problemu predykcji przeżywalności pooperacyjnej, proponuje się więc do indukcji reguł decyzyjnych zastosowanie podejścia wykorzystującego wyrocznię.

Tabela 5.2 zawiera również wyniki obrazujące jakość działania algorytmu regułowego RIPPER (oznaczanego jako **JRip**). Zastosowanie klasycznego podejścia do indukcji reguł wykorzystującego jedynie metodę **JRip** do silnie niezbalansowanego problemu doprowa-

dziło do wygenerowania jedynie reguły, która klasyfikuje wszystkie obiekty do klasy dominującej. W rezultacie wykrywalność przeżyć wyniosła 100% ($TN_{rate} = 100.00$, patrz Tabela 5.2), natomiast nie udało się zidentyfikować żadnego zgonu ($TP_{rate} = 0.00$). Mimo wysokiej poprawności klasyfikacji ($Acc = 85.11$), wyuczony na niezbalansowanym zbiorze danych klasyfikator **JRip** nie może zostać bezpośrednio wykorzystany do predykcji przeżywalności pooperacyjnej.

Klasyfikator **BSI** charakteryzuje się niższą poprawnością klasyfikacji niż **JRip**, jednak wykrywalność zgonów dla rozpatrywanego modelu „czarnej skrzynki" była na poziomie 60% przy wykrywalności przeżyć równej 72%. Wyuczony klasyfikator **BSI** został więc wykorzystany do ponownego nadania etykiet obiektom znajdującym się w zbiorze uczącym poprzez zastosowanie Algorytmu 6. Po wykonaniu procesu reetykietyzacji ponownie zastosowano metodę **JRip** do wygenerowania reguł decyzyjnych (**JRip** + **BSI** w Tabeli 5.2). W rezultacie wygenerowane zostały reguły charakteryzujące się wykrywalnością zgonów na poziomie 60% przy liczbie zidentyfikowanych przeżyć równej 70%. Jakość wygenerowanych reguł po zmianie etykiet klas była nieznacznie niższa niż jakość klasyfikatora **BSI** co w praktyce oznacza, że otrzymany zestaw reguł dobrze naśladuje mechanizmy działania tego silnego modelu.

Reguły	**Pokrycie**	**Dokładność**
(DGN = DGN5) => Risk1Yr=T	0.03	0.47
(PRE14 = OC14) => Risk1Yr=T	0.04	0.41
(PRE17 = T) and (PRE30 = T) i (AGE >= 57) => Risk1Yr=T	0.05	0.38
(PRE11 = T) i (PRE5 <= 2.16) i (PRE4 >= 2.44) => Risk1Yr=T	0.05	0.35
(PRE9 = T) i (AGE >= 54) i (PRE5 <= 66.4) => Risk1Yr=T	0.05	0.35
(PRE14 = OC13) => Risk1Yr=T	0.04	0.32
(DGN = DGN2) i (PRE30 = T) i (PRE14 = OC12) i (PRE5 <= 3.72) => Risk1Yr=T	0.04	0.30
(PRE8 = T) and (PRE30 = T) and (PRE4 <= 3.52) => Risk1Yr=T	0.08	0.26
OTHERWISE => Risk1Yr=F	0.62	0.97

Tabela 5.3: Reguły decyzyjne wygenerowane dla zbioru danych dotyczącego przeżywalności pooperacyjnej.

5.7 Indukcja reguł

W wyniku zastosowania metody **JRip** do wygenerowania reguł decyzyjnych ze zbioru o zmienionych etykietach otrzymano 9 reguł, które zostały przedstawione w Tabeli 5.3. Dla każdej reguły podane zostało jej pokrycie (*ang. coverage*), oraz dokładność (*ang. accuracy*). Pokrycie reguły definiowane jest jako stosunek liczby obserwacji których dana reguła dotyczy do całkowitej liczby obserwacji. Dokładność reguły określa procentową liczbę obserwacji prawidłowo klasyfikowanych przez daną regułę spośród wszystkich przykładów przez nią pokrytych.

Zakres dokładności wygenerowanych reguł dla klasy zdominowanej wahał się od 0.26 do 0.47. Dla zadanego problemu decyzyjnego nie możliwe jest więc wyodrębnienie reguł o wysokiej pewności dla klasy pozytywnej. Wygenerowany zestaw reguł pozwolił zidentyfikować obszary podprzestrzeni cech w przypadku których występuje podwyższone ryzyko zgonu. Pokrycie dla wykrytych reguł z klasy pozytywnej wahało się od 3% to 8% obserwacji ze zbioru.

Podprzestrzeń cech nie pokryta przez reguły z klasy zdominowanej charakteryzowała się wysoką (97%) dokładnością, przy pokryciu równym 62%. Zestaw reguł daje więc możliwość bardzo dokładnego zidentyfikowania przypadków, dla których pacjent przeżył okres dłuższy niż rok po operacji.

Zastosowanie podejścia wyroczni w połączeniu z wzmacniamy algorytmem SVM pozwoliło określić obszary podwyższonego ryzyka zgonu. Wygenerowane reguły, podawane wraz z wartościami pokrycia i dokładności, stanowią istotną wiedzę dla ekspertów z dziedziny medycyny pozwalającą zaplanować dalsze leczenie pacjenta, oraz lepsze zrozumienie rozpatrywanego zjawiska.

5.8 Problem brakujących wartości atrybutów

Przeprowadzone dotychczas badania dotyczące zagadnienia analizy ryzyka operacyjnego były wykonane na zbiorze obserwacji opisanych kompletnym wektorem cech. Wejściowy zbiór danych został ograniczony z 1203 do 470 obserwacji niezawierających brakujących wartości cech. Atrybutami wśród których zaobserwowano brakujące wartości były dwie cechy związane z mierzoną wydolnością płuc pacjenta: *PRE4* oraz *PRE5* ($61.\%$ brakujących

wartości atrybutów).

Zagadnienie brakujących wartości atrybutów jest jednym z typowych problemów z danymi wykorzystywanymi do konstrukcji modeli decyzyjnych [50]. Brak znajomości wartości niektórych cech klasyfikowanych obiektów może wynikać z różnych przyczyn, braku możliwości przeprowadzenia pewnych badań u pacjentów, błędnego sposobu gromadzenia danych, czy też utraty danych ze względu na błędy w procesie przetwarzania. W literaturze wyróżnia się szereg podejść do problemu niekompletności danych w ramach których wyróżnić można cztery grupy metod [148]:

1. Techniki eliminacji braków danych w ramach których wyróżnia się eliminację przypadkami (jednorazową redukcję zbioru uczącego do kompletnych obserwacji) i eliminację parami (każdorazowe usuwanie z obliczeń przypadków z brakami danych dla wykorzystywanych zmiennych).

2. Techniki imputacji (*ang. imputation techniques*), polegające na uzupełnianiu brakujących obserwacji atrybutów na podstawie metod statystycznych lub technik uczenia maszynowego.

3. Techniki polegające na estymacji funkcji gęstości rozkładu generującego dane.

4. Techniki eliminujące problem brakujących wartości atrybutów na poziomie uczenia klasyfikatora.

Celem uniknięcia redukcji obserwacji w rozpatrywanym zbiorze danych dla których wartości *PRE4* , oraz *PRE5* są nieznane przeanalizowano jakość działania wybranych metod dedykowanych do rozwiązania problemu brakujących wartości atrybutów. W badaniu wzięto pod uwagę następujące algorytmy:

- Metodę eliminacji atrybutów (**AE-MV**). Atrybuty zawierające brakujące wartości nie są rozpatrywane w procesie klasyfikacji.

- Metodę polegającą na maksymalizacji wartości oczekiwanej (*ang. Expectation Maximization*, **EM-MV**) [111].

- Metodę bazującą na Bayesowskiej Analizie Składowych Głównych (*ang. Bayesian Principal Component Analysis*, **BPCA-MV**) [95]. Metoda wykonuje estymację brakujących wartości poprzez równoczesne wykorzystanie regresji składowych głównych (*ang. Principal Component Regression*), wnioskowanie Bayesowskie i algorytm EM.

- Metodę bazującą na algorytmie K najbliższych sąsiadów (**KNN-MV**) [7]. Brakująca wartość stanowi średnią wartość (bądź wartość najczęściej obserwowaną, w przypadku atrybutów nominalnych) K najbliższych sąsiadów obserwacji.

- Metodę wykorzystującą ważoną wersję algorytmu K najbliższych sąsiadów (**WKNN-MV**) [133]. Modyfikacja metody **KNN-MV** uwzględniająca dodatkowo odległości pomiędzy sąsiadami.

- Metodę wykorzystującą algorytm K-średnich (**KMeans-MV**) [81]. Brakująca wartość atrybutu obserwacji stanowi średnią wartość (bądź wartość najczęściej obserwowaną, w przypadku atrybutów nominalnych) cechy dla obserwacji znajdujących się w klastrze otrzymanym poprzez zastosowanie algorytmu K-średnich.

- Metodę wykorzystującą rozmyty algorytm (**FKMeans-MV**) [81]. Modyfikacja podejścia **KMeans-MV** polegająca na uwzględnieniu w procesie wstawiania wartości stopnia przynależności obiektu do danego klastra.

- Technikę wstawiania brakujących wartości wykorzystującą metodę lokalnych najmniejszych kwadratów (*ang. Local Least Squares Imputation*, **LLSI-MV**) [69]. Każda obserwacja z brakującymi wartościami atrybutów stanowi kombinację liniową kompletnych obserwacji podobnych. W ramach metody wyróżnia się dwa kroki. W pierwszym kroku wyznaczane są lokalne obserwacje znajdujące się najbliżej ze względu na przyjętą normę L_2. W drugim kroku dla lokalnych obserwacji estymowane są parametry regresji z wykorzystaniem metody najmniejszych kwadratów.

- Technikę wstawiania polegającą na uzupełnianiu brakujących wartości z wykorzystaniem regresji wektorów wspierających (*ang. Support Vector Regression*, **SVR-MV**) [60]. Brakująca wartość traktowana jest jako wartość wyjściowa modelu regresji i estymowana jest na podstawie znanych wartości innych atrybutów.

Metoda	$\mathbf{TP_{rate}}$	$\mathbf{TN_{rate}}$	**Acc**	**GMean**
AE-MV	62.27	63.68	63.42	62.97
EM-MV	37.73	**78.84**	**71.32**	54.54
BPCA-MV	40.91	56.97	54.03	48.28
KNN-MV	61.36	64.80	64.17	63.06
WKNN-MV	**63.18**	62.97	63.01	**63.08**
KMeans-MV	55.00	67.45	65.17	60.91
FKMeans-MV	60.91	64.50	63.84	62.68
LLSI-MV	59.09	66.84	65.42	62.84
SVR-MV	37.27	77.62	70.24	53.79

Tabela 5.4: Wyniki dla różnych technik eliminacji brakujących wartości atrybutów. Analizę jakości zastosowanych metod przeprowadzono dla klasyfikatora **BSI**.

Wyniki badań jakości przedstawionych metod w kontekście ich zastosowania do eliminacji problemu brakujących wartości atrybutów przedstawiono w Tabeli 5.4. Jako metodę klasyfikacji, bazując na wynikach przedstawionych w Tabeli 5.3, wybrano do badań algorytm **BSI**. Jako kryteria oceny metod przyjęto, podobnie jak w poprzednich badaniach, wartości TP_{rate}, TN_{rate}, Acc, oraz $GMean$. Jako metodykę eksperymentu przyjęto walidację krzyżową z podziałem na 5 podzbiorów.

Najwyższą wartość wskaźnika $GMean$ została osiągnięta po zastosowaniu metody uzupełniania wartości z wykorzystaniem ważonego algorytmu K-najbliższych sąsiadów (**WKNN-MV**). Nieznacznie niższą wartość zaobserwowano dla klasycznej odmiany metody uzupełniania (**KNN-MV**), oraz dla podejścia polegającego na usunięciu atrybutów z brakującymi wartościami (**AE-MV**). Ze względu na silny stopień niezbalansowania danych, oraz wysoki procent brakujących wartości atrybutów zaobserwowano, że wyniki osiągnięte przez metody **EM-MV**, **BPCA-MV**, **SVR-MV** osiągnęły istotnie niższą wartość wskaźnika $GMean$ niż wyniki innych rozpatrywanych metod. Najwyższa wartość wskaźnika $\mathbf{TP_{rate}}$ została zaobserwowana dla metody **WKNN-MV** i była to jedyna metoda, która osiągnęła wyższą wartość wskaźnika niż metoda eliminacji atrybutów **AE-MV**. Wysoka jakość metody **WKNN-MV** wyrażona poprzez kluczowe dla problemu niezbalansowania wskaźniki $\mathbf{TP_{rate}}$, oraz $GMean$ świadczy o tym, że jest ona najlepszą metodą wstawiania brakujących wartości atrybu-

tów dla rozpatrywanego problemu klasyfikacji wśród algorytmów rozpatrywanych w badaniu. Jednak porównując wyniki otrzymane dla metody **BSI** przedstawione w Tabeli 5.2, w przypadku których zastosowano technikę eliminacji obserwacji z brakującymi wartościami, (wartość $GMean$ równa 65.73) z wynikami zamieszonymi w Tabeli 5.4 (najwyższa wartość $GMean$ równa 63.08 dla metody **WKNN-MV**), stwierdza się, że metoda usunięcia brakujących obserwacji jest rozwiązaniem lepszym w stosunku do innych technik wstawiania wartości dla rozpatrywanego zadania klasyfikacji.

5.9 Dyskusja

W niniejszym rozdziale przedstawiono przykład zastosowania opracowanych w ramach rozprawy metod wzmacnianych klasyfikatorów SVM dla danych niezbalansowanych do problemu predykcji przeżywalności pacjenta po operacji raka płuc. W pierwszej kolejności przebadano jakość opracowanych rozwiązań poprzez porównanie wyników osiągniętych przez najlepsze rozwiązania dedykowane do problemu niezbalansowania (włączając opracowane w ramach rozprawy algorytmy **BSI**, **BSI1**, oraz **BSI2**) na rzeczywistym zbiorze danych medycznych. W oparciu o wyniki przeprowadzonych badań stwierdzono, że najwyższą jakością na rozpatrywanym zbiorze charakteryzuje się metoda **BSI**. W dalszej kolejności opracowano metodę generowania reguł decyzyjnych z modelu „czarnej skrzynki" bazującą na podejściu „wyroczni" i zastosowano ją do indukcji reguł z **BSI**. Analiza jakości otrzymanego w rezultacie interpretowalnego modelu wykazała nieznacznie niższą wartość $GMean$ w porównaniu do bazowego modelu **BSI**. Otrzymane reguły decyzyjne dla rozpatrywanego zbioru danych medycznych przedstawiono w Tabeli 5.3. W ostatnim kroku, ze względu na występujący w przypadku dwóch atrybutów problem brakujących wartości, przeanalizowano różne techniki uzupełniania niekompletnych wartości cech. Najwyższą skutecznością spośród technik wstawiania charakteryzowała się metoda wykorzystująca ważoną wersję algorytmu K-NN. Osiągnięty przez nią wynik był jednak gorszy niż w przypadku zastosowania podejścia polegającego na usunięciu obserwacji z niekompletnymi wartościami atrybutów.

Rozdział 6

Zastosowanie metod w systemach o paradygmacie SOA

W rozdziale tym opisano koncepcję architektury udostępniania usług uczenia maszynowego celem komercjalizacji rozwiązań opisanych w rozprawie. Poprzez opracowany paradygmat udostępniania usług eksploracji danych opracowane metody zostały zastosowane do rozwiązania dwóch rzeczywistych problemów:

- problemu oceny ryzyka kredytowego,
- problemu detekcji anomalii w trybie nadzorowanym.

6.1 Systemy o paradygmacie SOA

W obecnych czasach powszechne stało się projektowanie systemów zgodnie z paradygmatem SOA (*ang. Service Oriented Architecture*). Podstawą tej architektury stanowi semantycznie opisana usługa (*ang. service*), która poprzez ustrukturalizowany interfejs realizuje charakterystyczną dla danej domeny funkcjonalność. Podejście usługowe umożliwia organizacjom i klientom budowanie, dystrybucję i integrowanie usług niezależnie od wykorzystywanych technologii. Aby sprostać złożonym wymaganiom klientów usługi atomowe (*ang. atomic services*) łączone są w ramach procesu kompozycji w bardziej złożone struktury tworząc tzn. usługi złożone (*ang. complex services*) [118].

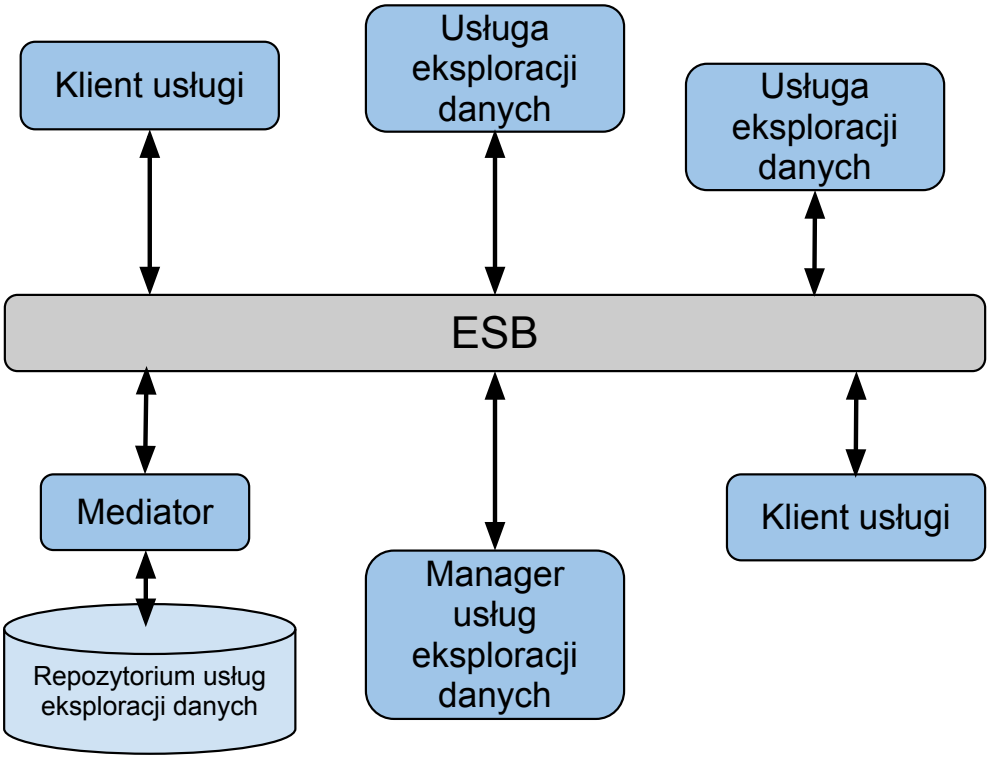
Najbardziej rozpowszechnionym i najczęściej stosowanym przykładem usługi jest usługa Webowa (*ang. Web service*) [93]. Usługa Webowa jest udostępnianą w sieci usługą wykorzystującą w opisie i komunikacji standardy bazujące na XML (*ang. Extensible Markup Language*). W obecnym kształcie stosuje się je celem integracji heterogenicznych aplikacji poprzez zunifikowane interfejsy Webowe (*ang. Web interfaces*). Standardem opisu usług jest język WSDL (*ang. Web Services Description Language*), natomiast typowym protokołem komunikacji jest protokół SOAP (*ang. Simple Object Access Protocol*).

Usługi Webowe są udostępniane głównie za pomocą tzn. szyny danych ESB (*ang. Enterprise Service Bus*). ESB stanowi rozszerzalną infrastrukturę warstwy pośredniej umożliwiającą zarządzanie komunikacją pomiędzy usługami [16]. Dzięki szynie danych możliwa jest integracja narzędzi działających w różnych technologiach.

6.2 Architektura zorientowanego na usługi systemu eksploracji danych

Rozwiązania uczenia maszynowego w zastosowaniu do eksploracji danych są udostępniane za pomocą wielu różnych narzędzi, takich jak: *Weka* [140], *KEEL Software* [1], *Statistica* [101], oraz pakiety do *Microsoft Business Intelligence* [90] czy też narzędzia *Matlab* [88]. Większość z opisanych narzędzi funkcjonuje w trybie *offline* i wykorzystywana jest główne do analizy własności udostępnianych algorytmów. Tylko nieliczne z wymienionych narzędzi (*Statistica*, *Microsoft Business Intelligence*) w sposób wybiórczy i komercyjny udostępniają swoje rozwiązania za pośrednictwem usług Webowych. W literaturze zaobserwowano jedynie kilka prototypowych rozwiązań dotyczących eksploracji danych implementowanych zgodnie z paradygmatem SOA: *WebDiscC* [134], oraz *DisDaMin* [45]. Celem komercjalizacji opracowanych w ramach rozprawy rozwiązań zaproponowano zorientowaną na usługi architekturę projektowania systemów eksploracji danych (*ang. Service Oriented Data Mining Architecture*, SODMA) [104]. [1]

[1]Opracowane w tym rozdziale rozwiązania zostały współfinansowane ze środków Unii Europejskiej poprzez Europejski Fundusz Rozwoju Regionalnego w ramach Programu Operacyjnego Innowacyjna Gospodarka na lata 2007-2013, numer projektu: POIG.01.03.01-00-008/08.



Rysunek 6.1: System eksploracji danych zgodny z SODMA.

Istotą opracowanej architektury SODMA jest to, że każda metoda uczenia maszynowego udostępniana jest za pomocą usługi Webowej o ustandaryzowanym interfejsie. W ramach systemu zgodnego z SODMA wyodrębnia się następujące komponenty (Rysunek 6.1): usługi eksploracji danych (*ang. data mininig services*, DMS), manager usług eksploracji danych (*ang. data mining services manager*, DMSM), oraz repozytorium usług eksploracji danych (*ang. data mining services repository*, DMSR). DMS reprezentują rozwiązania uczenia maszynowego w zakresie klasyfikacji, regresji, oraz grupowania. Moduł zarządzający usługami (DMSM) pozwana na podstawie wymagania klienta i dostarczonych przez niego danych określić, która usług występujących w ramach DMS w najlepszym stopniu pozwoli spełnić jego wymaganie. Każda z usług DMS opisana jest w DMSR, a dostęp do repozytorium odbywa się za pośrednictwem tzn. mediatorów (*ang. mediators*) [52], czyli usług zarządzającymi dostępem do danych. Każda z funkcjonalności komponentów realizowana jest za pośrednictwem usługi Webowej. Komunikacja pomiędzy klientami i usługami

odbywa się za pośrednictwem szyny ESB, z wykorzystaniem protokołu SOAP. Klientem systemu eksploracji danych zgodnego z paradygmatem SODMA może być użytkownik (za pośrednictwem interfejsu graficznego komunikującego się z DMS za pomocą ESB), proces biznesowy, czy też inna aplikacja komunikująca się za pośrednictwem szyny danych.

6.2.1 Funkcjonalność SODMA

Zgodnie z paradygmatem SODMA każda usługa Webowa reprezentuje jedną z metod uczenia maszynowego. Usługi DMS można podzielić na dwie grupy: usługi realizujące funkcjonalności metod działających w trybie nienadzorowanym (metody klasteryzacji), oraz działających w trybie nadzorowanym (metody klasyfikacji i regresji). Usługi realizujące funkcjonalności w ramach pierwszej z wymienionych grup ograniczonej do metod klasteryzacji charakteryzują się jedną operacją „*cluster*" która dla zadanego na wejściu zbioru danych i zależnych od konkretnej metody parametrów przeprowadzała procedurę grupowania danych wejściowych zwracając zaetyiketowany zbiór danych. Każda z etykiet reprezentowała indeks klastra otrzymanego w procesie grupowania.

Metody należące do grupy algorytmów z uczeniem nadzorowanym reprezentowane są przez usługi w ramach których definiuje się następujące operacje:

- *BuildClassifier*. Operacja budowy (uczenia) klasyfikatora (bądź modelu regresji) z wykorzystaniem zadanego na wejściu zbioru danych. Parametrami wejściowymi operacji są: zaetyiketowany zbiór treningowy, oraz zestaw parametrów charakterystycznych dla zadanego problemu uczenia. W wyniku wykonania operacji budowany jest klasyfikator (model regresji) poprzez wykorzystanie zadanego na wejściu zbioru danych. Skonstruowany model jest następnie zapisywany na serwerze. Unikalny klucz do wyuczonego klasyfikatora jest zwracany jako wyjściowy parametr operacji.

- *ClassifyInstances*. Operacja klasyfikacji (wyznaczenia wartości wyjściowych dla zadania regresji) nowych obserwacji. Parametrami wejściowymi operacji są: niezaetykietowany zbiór obserwacji, oraz klucz identyfikujący wyuczony model, który ma być wykorzystany do nadania etykiet obserwacjom. W wyniku wykonania operacji każdej obserwacji nadawana jest etykieta poprzez wykorzystanie wyuczonego modelu klasyfikacji (regresji) zidentyfikowanego poprzez wartość klucza. Parametrem wyjściowym

dla operacji jest zbiór danych z uzupełnionymi wartościami etykiet.

- *TestClassifer*. Operacja testowania klasyfikatora (modelu regresji) na zadanym na wejściu zbiorze danych. Podobnie jak w przypadku operacji budowy klasyfikatora parametrami wejściowymi operacji są: zetykietowany zbiór danych, oraz zestaw parametrów uczenia dla metody reprezentowanej przez usługę. W wyniku wykonania operacji uruchamiania jest procedura testowania modelu na zbiorze danych z wykorzystaniem walidacji krzyżowej. Parametrem wyjściowym operacji jest zestaw statystyk pozwalający określić jakość modelu dla rozpatrywanych danych.

- *UpdateClassifier*. Operacja aktualizacji klasyfikatora (modelu regresji) w oparciu o nową porcję danych uczących (dotyczy metod funkcjonujących w trybie przyrostowym). Parametrami wejściowymi operacji są: zetykietowany zbiór treningowy, oraz klucz identyfikujący wyuczony model. W wyniku wykonania operacji model o zadanym poprzez klucz identyfikatorze zostanie zaktualizowany w trybie uczenia przyrostowego poprzez wykorzystanie zadanych na wejściu danych. Parametrem wyjściowym jest status operacji informujący o efekcie wykonania procesu aktualizacji.

- *PrintClassifier*. Operacja wydruku modelu w formie tekstowej. W zależności od typu modelu operacja może zwracać w formie tekstowej np. zestaw reguł decyzyjnych, drzewo decyzyjne, bądź też wartości parametrów modeli funkcyjnych. Parametrem wejściowym dla operacji jest klucz identyfikujący wyuczony model. Parametrem wyjściowym jest tekstowy opis modelu.

- *GetCapabilities*. Operacja zwracająca własności typu modelu. Operacja nie pobiera parametrów wejściowych. Operacja jako parametr wyjściowy zwraca zestaw własności opisujących typ modelu klasyfikatora (regresji) reprezentowanego przez daną usługę. Przykładową własnością zwracaną przez operację może być możliwość obsługi danych niezbalansowanych, czy też brakujących wartości atrybutów.

Zgodnie z paradygmatem SODMA udostępniono opracowane w rozprawie algorytmy *BoostingSVM-IB*, *BoostingSVM-IB.M1*, oraz *BoostingSVM-IB.M2* poprzez odrębne usługi Webowe. Wymienione operacje dla każdej z usług zostały zrealizowane poprzez algorytmy uczenia opisane w rozprawie. Dodatkowo, w wyniku wykonania operacji *PrintClassifier*

zwracany jest zestaw reguł decyzyjnych wygenerowanych poprzez zastosowanie opisanego w pracy podejścia „wyroczni". Ponadto, zgodnie z paradygmatem SODMA, udostępniono poprzez usługi Webowe następujące typowe metody uczenia maszynowego:

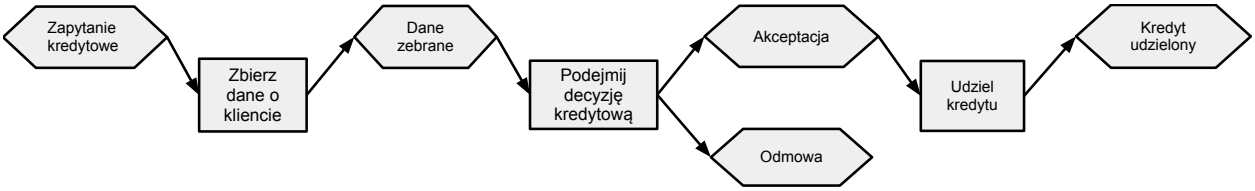
- algorytm Naiwnego Bayesa (klasyfikacja);
- algorytm konstrukcji drzewa decyzyjnego C 4.5 (klasyfikacja);
- metoda indukcji reguł algorytmem RIPPER (klasyfikacja);
- algorytm regresji logistycznej (klasyfikacja);
- algorytm trzywarstwowej sieci neuronowej uczonej metodą propagacji wstecznej (klasyfikacja, regresja);
- algorytm hierarchicznego grupowania, ROCK (*ang. robust clustering algorithm for categorical attributes*) [102] (grupowanie);
- algorytm EM (*ang. expectation maximization*) (grupowanie);
- algorytm K-średnich (grupowanie).

6.3 Przykład użycia - problem oceny ryzyka kredytowego

Metody uczenia maszynowego są powszechnie stosowane do rozwiązywania problemów podejmowania decyzji w dziedzinie ekonomii. Konieczne jest więc opracowanie rozwiązań pozwalających na efektywne ich wykorzystanie w procesach biznesowych wymagających rozwiązania problemów decyzyjnych. Dzięki opracowanej architekturze SODMA możliwe jest bezproblemowe wykorzystanie rozwiązań poprzez wywołanie odpowiednich usług DMS bez konieczności ingerencji w strukturę systemów informatycznych wykorzystywanych przez instytucje finansowe.

Typowym problemem decyzyjnym w dziedzinie ekonomii jest ocena ryzyka kredytowego, który w wielu przypadkach sprowadzany jest do problemu klasyfikacji, gdzie klasy odpowiadają możliwym wariantom decyzji kredytowych. Problem modelowania ryzyka kredytowego po raz pierwszy został rozwiązany przez Durmana w 1941 [19, 150], który

zaproponował funkcję dyskryminacyjną celem oddzielenia „dobrych" i „złych" klientów. W latach 80-tych rozwój pożyczek osobistych, kart kredytowych, oraz kredytów gotówkowych idący w parze z rozwojem systemów informatycznych spowodował, że powszechne do oceny zdolności kredytowej stało się wykorzystanie modeli regresji logistycznej i programowania liniowego. Lata 90-te zapoczątkowały stosowanie do oceny ryzyka metod eksploracji danych, takich jak drzewa decyzyjne i algorytmy regułowe. Z kolei początek XXI przyniósł rozwój hybrydowych i złożonych metod klasyfikacji [61, 138, 150, 152].



Rysunek 6.2: Proces przydziału kredytu klientowi bankowemu.

Proces biznesowy charakteryzujący procedurę przydziału kredytu klientowi został umieszczony na Rysunku 6.2. W pierwszej kolejności zbierane są dane na temat klienta, które mają istotny wpływ na decyzję kredytową. Dane te wprowadzane są następnie do systemu, który wykonuje ocenę analizy ryzyka kredytowego i wspomaga pracownika w podjęciu ostatecznej decyzji o przydzieleniu kredytu. W obecnych systemach wspomagających decyzje kredytowe stosowane są tzn. tablice scoringowe (*ang. scoring tabels*) [91, 117], w ramach której każdemu z atrybutów charakteryzujących kredytobiorcę przyporządkowywana jest określona liczba punktów. [2] Im wyższa jest wartość przyporządkowywanych punktów, tym większy jest wpływ danej cechy na pozytywną decyzję kredytową. Model w postaci tabeli scoringowej konstruowany jest za pomocą eksperta, bądź też z wykorzystaniem metod uczenia maszynowego, głównie modeli regresji logistycznej. Podejścia wykorzystujące tablice scoringowe są powszechnie stosowane w wielu polskich bankach ze względu na zrozumiałą dla człowieka procedurę podejmowania decyzji, możliwość modyfikacji samego modelu decyzyjnego poprzez zmianę liczby punktów skojarzonych z atrybutem, oraz dzięki prostocie w implementacji. Tablice scoringowe charakteryzują się jednak niższą jakością klasyfikacji, niż metody takie jak SVM, czy klasyfikatory wzmacnianie.

[2] Atrybuty numeryczne poddawane są dyskretyzacji, następnie wszystkie atrybuty nominalne są binaryzowane. W rezultacie każdy z atrybutów określa wystąpienie cechy u kredytobiorcy

Przyszłość zagadnień dotyczących analizy ryzyka kredytowego należeć będzie do zaawansowanych technik uczenia maszynowego, takich jak opracowane w ramach rozprawy wzmacniane algorytmy SVM. Konieczne jest więc opracowanie mechanizmów umożliwiających systemom bankowym dostęp do tego typu rozwiązań. Dzięki opracowanej architekturze udostępniania metod uczenia maszynowego SODMA możliwe jest wykorzystanie tego typu algorytmów poprzez uniwersalne interfejsy Webowe. W takim podejściu komponent związany z podejmowaniem decyzji w rozpatrywanym procesie biznesowym może być realizowany poprzez wywołanie odpowiednich usług DMS. Zestaw modeli wspomagających decyzje kredytowe nie jest więc ograniczony do zaimplementowanych w ramach systemu bankowego tablic scoringowych. Dzięki elastycznej, zunifikowanej komunikacji z DMS możliwe jest wykorzystanie dowolnego algorytmu klasyfikacji dostępnego w ramach systemu zgodnego z SODMA. Pracownik banku ma możliwość porównania wyników analizy ryzyka kredytowego z wykorzystaniem różnych modeli klasyfikacyjnych. Dodatkowo, dla nowych metod udostępnianych za pośrednictwem DMS ze względu na ujednolicony sposób komunikacji z usługami możliwe jest wykorzystanie najnowszych rozwiązań bez konieczności ingerencji w implementację system informatycznego dostępnego w banku. Konieczna jest jedynie rejestracja usługi DMS w systemie informatycznym banku poprzez podanie lokalizacji jej opisu w języku WSDL.

6.3.1 Analiza jakości metod niezbalansowanych w kontekście oceny ryzyka dla kredytów 30-dniowych

Opracowane w ramach rozprawy złożone metody klasyfikacji zostały wdrożone w przedsiębiorstwie projektującym systemy informatyczne dla instytucji parabankowych poprzez wykorzystanie architektury SODMA. W ramach wdrożenia wykorzystano usługi DMS do konstrukcji i późniejszego wykorzystania modelu decyzyjnego określającego ryzyko spłacalności kredytów 30-dniowych. Problem decyzyjny został zdefiniowany jako dychotomiczne zadanie klasyfikacji w którym jedna z możliwych klas reprezentowała sytuację w której klient spłacił kredyt (terminowo, bądź z opóźnieniem bez konieczności wszczynania procedury windykacyjnej), druga natomiast dotyczyła sytuacji w której klientowi nie udało się spłacić kredytu i konieczna była windykacja należności.

Celem doboru odpowiedniej metody klasyfikacji dokonano analizy jakości dostępnej w ramach DMS algorytmów na rzeczywistym zbiorze danych dotyczącym kredytów 30-dniowych. Wykorzystany w eksperymencie zbiór danych składał się z 1146 obiektów, z których 1005 klientów spłaciło kredyt, natomiast w przypadku 141 osób wszczęto procedurę windykacyjną. Każdy z klientów był opisany wektorem 9 cech, takich jak kwota kredytu, miesięczny dochód, czy też rodzaj źródła dochodu. W badaniu przeanalizowano udostępnione w ramach DMS metody klasyfikacji: algorytm Naiwnego Bayesa (**NB**), drzewo decyzyjne C 4.5 (**J48**), RIPPER (**JRip**), regresję logistyczną (**LR**), sieć neuronową (**MLP**), oraz opracowane w ramach rozprawy metody **BSI**, **BSI1** i **BSI2**. Dodatkowo, ze względu na wysoki stopień niezbalansowania danych, przeanalizowano jakość najefektywniejszych metod rozwiązujących problem dysproporcji pomiędzy klasami: **UB**, **RUS** i **SSVM**. Wyniki przeprowadzonych badań udostępniono w Tabeli 6.1.

Metoda	$\mathbf{TP_{rate}}$	$\mathbf{TN_{rate}}$	**Acc**	**GMean**
UB	**63.83**	59.30	59.86	61.53
RUS	44.68	73.93	70.33	57.47
SSVM	65.96	55.92	57.16	60,73
BSI	59.57	64.48	63.87	61,98
BSI1	62.41	63.18	63.09	62.80
BSI2	63.12	63.88	63.79	**63.50**
JRip	0.00	**100.00**	**87.70**	0.00
J48	0.00	**100.00**	**87.70**	0.00
NB	25.53	87.76	80.10	47.34
MLP	4.26	96.42	85.08	20.26
LR	0.00	**100.00**	**87.70**	0.00

Tabela 6.1: Wyniki dla zbioru danych dotyczącego analizy ryzyka kredytowego.

Tradycyjne metody klasyfikacji udostępnione w ramach DMS charakteryzowały się zerową, bądź bardzo niską wartością $GMean$. Najwyższą wartość wskaźnika spośród metod nieposiadających mechanizmów obsługi danych niezbalansowanych zaobserwowano dla algorytmu Naiwnego Bayesa. W przypadku metod **UB**, **RUS** i **SSVM** posiadających wbudowane mechanizmy redukujące niezbalansowanie danych najwyższa wartość $GMean$ zosta-

ła zaobserwowana dla metody **UB**. Metoda ta, podobnie jak w przypadku badań opisanych w poprzednich rozdziałach charakteryzowała się najwyższą wartością wskaźnika TP_{rate}, przy najniższej wartości TN_{rate}. Najwyższą wartość wskaźnika $GMean$ zaobserwowano dla metod opracowanych w ramach rozprawy: **BSI**, **BSI1** oraz **BSI2**. Algorytmy **BSI1** oraz **BSI2**, które wykorzystują mechanizmy eliminacji obserwacji nieinformacyjnych osiągnęły wyższą wartość wskaźnika $GMean$. Potwierdza to tezę, że metody wykorzystujące eliminację charakteryzują się wyższą jakością działania dla danych wysoce niezbalansowanych (Wskaźnik niezbalansowania dla danych dotyczących ryzyka kredytowego był równy 7.13).

6.4 Przykład użycia - detekcja anomalii

Problem detekcji anomalii dotyczy znajdowania wzorców w danych które odbiegają od oczekiwanych zachowań [22]. Zagadnienie detekcji anomalii stanowi istotny element systemów działających zgodnie z paradygmatem SOA [71, 72]. W zależności od charakteru danych wyróżnia się trzy podejścia do problemu detekcji anomalii:

- techniki nienadzorowane (*ang. unsupervised*).
- techniki częściowo nadzorowane (*ang. semi-supervised*).
- techniki nadzorowane (*ang. supervised*).

Techniki nienadzorowane wykorzystują dane bez korespondujących etykiet klas określających wystąpienie anomalii i bazują na założeniu, że nieoczekiwane zachowania pojawiają się rzadziej w stosunku do zachowań normalnych. Do rozwiązania problemu detekcji anomalii w trybie nadzorowanym stosuje się metody grupowania takie jak algorytmy hierarchicznego grupowania Bayesa [57, 128], czy jednoklasowy SVM [82].

Techniki częściowo nadzorowane wykorzystują dane, w przypadku których jedynie część obserwacji posiada nadane etykiety klasy. W większości przypadków wybiórcze występowanie wartości klas w zbiorze uczącym jest podyktowane wysokim kosztem uzyskania informacji o klasie dla pojedynczej informacji. Do rozwiązania problemu detekcji anomalii w trybie częściowo nadzorowanym stosuje się m. in. Gaussowskie pola losowe [145], czy też zmodyfikowane klasyfikatory SVM [9].

W przypadku technik nadzorowanych każda obserwacja w danych posiada korespondującą etykietę klasy. Ze względu na znaczną dysproporcję pomiędzy liczbą obserwacji traktowanych jako anomalie, a liczbą normalnych obiektów problem detekcji anomalii w trybie nadzorowanym definiuje się jako zadanie klasyfikacji dla danych niezbalansowanych. Do rozwiązania problemu konieczne jest więc zastosowanie metod niwelujących negatywne skutki niezbalansowania, takich jak opracowane w ramach rozprawy wzmacniane algorytmy SVM.

Metody **BSI**, **BSI1** oraz **BSI2** zostały wykorzystane do detekcji niepożądanych wywołań usług Webowych w testowym systemie działającym zgodnie z paradygmatem SOA z wykorzystaniem opracowanej architektury SODMA. Zadaniem opracowanych metod klasyfikacji była ocena na podstawie cech dotyczących połączenia, czy wywołanie usługi Webowej było niepożądane.

Metoda	$\mathbf{TP_{rate}}$	$\mathbf{TN_{rate}}$	**Acc**	**GMean**
UB	88.89	96.34	96.00	92.54
RUS	**90.48**	97.86	98.21	**94.26**
SSVM	87.30	97.76	97.29	92.38
BSI	87.30	99.93	**99.36**	93.40
BSI1	85.71	99.85	99.21	92.51
BSI2	85.71	99.93	85.71	92.55
JRip	85.71	99.55	98.93	92.37
J48	85.71	99.70	99.07	92.44
NB	88.89	92.83	92.65	90.84
MLP	85.71	**100.00**	**99.36**	92.58
LR	71.43	97.61	96.43	83.50

Tabela 6.2: Wyniki dla zbioru danych dotyczącego detekcji anomalii.

Ze względu na brak dostępu do rzeczywistych danych analiza jakości opracowanych metod została przeprowadzona na *benchmarkowym* zbiorze danych *KDD Cup 1999* udostępnionych w repozytorium UCI [4].[3] Zbiór danych składał się z obiektów reprezentujących

[3]Zbiór danych był przedmiotem konkursu *International Knowledge Discovery and Data Mining Tools Competition* w roku 1999.

połączenia w sieci będące atakami („złe połączenia"), oraz normalne połączenia sieciowe („dobre połączenia").

Zbiór danych został przetworzony celem osiągnięcia niebalansowania pomiędzy klasami. Wynikowy zbiór danych składał się z 1338 obserwacji reprezentujących „dobre połączenia", oraz 63 obserwacje będące anomaliami. Wykonano selekcję cech wykorzystując kryterium pojemności informacyjnej zadane wzorem (5.1). Wskaźnik niezbalansowania dla rozpatrywanego zbioru danych był równy 21.24.

W badaniu wykorzystano udostępnione poprzez DMS metody: **JRip**, **J48**, **NB**, **MLP**, **LR**, najskuteczniejsze metody dedykowane do problemu niezbalansowania: **UB**, **RUS**, **SSVM**, oraz opracowane w ramach rozprawy algorytmy: **BSI**, **BSI1**, oraz **BSI2**. Badania przeprowadzono z wykorzystaniem walidacji krzyżowej z podziałem na 5 podzbiorów.

Wyniki przeprowadzonych badań zostały przedstawione w Tabeli 6.2. Pomimo wysokiego stopnia niezbalansowania zbioru danych wyniki osiągnięte przez metody dedykowane dla problemów zbalansowanych były porównywalne z wynikami osiągniętymi przez metody posiadające mechanizmy przeciwdziałania dysproporcjom w zbiorze uczącym. Wynika to z charakteru samego problemu decyzyjnego dla którego obserwacje należące do klasy zdominowanej charakteryzują się unikatowymi wartościami atrybutów w stosunku do obiektów reprezentujących „dobre połączenia" co powoduje, że obiekty są łatwo separowalne z wykorzystaniem klasycznych klasyfikatorów. Przekłada się to bezpośrednio na wysoki poziom poprawności klasyfikacji ($90\% - 99\%$ poprawnie sklasyfikowanych obiektów dla większości z metod klasyfikacyjnych uwzględnionych w eksperymencie).

Najwyższa wartość $GMean$ osiągnięta została przez metodę **RUS**. Nieznacznie niższą wartość kryterium zaobserwowano dla opracowanej w ramach rozprawy metody **BSI**, jednak poprawność klasyfikacji dla autorskiego rozwiązania była o ponad 1% wyższa niż w przypadku algorytmu **RUS** i najwyższa (obok **MLP**) spośród wszystkich metod uwzględnionych w badaniu. Analizując wyniki badania przedstawione w Tabeli 6.2 stwierdza się, że pomimo wysokiego poziomu niezbalansowania samych danych wykorzystywanych w procesie uczenia, większość z metod klasyfikacji podejmowała decyzje nieobciążone w kierunku klasy dominującej.

6.5 Inne zastosowania

Podane przykłady zastosowania nie wyczerpują możliwości wdrożenia opracowanych metod w innych obszarach. Przedstawione rozwiązania zostały również zastosowane w zadaniach: selekcji usług na podstawie wymagań zdefiniowanych przez klienta, oraz w zadaniu nadania priorytetów klientom dla usług o ograniczonej liczbie wykonań.

Proces selekcji usług (PSU) jest jednym z kluczowych elementów scenariuszy realizowanych w systemach zorientowanych na usługi. Głównym zadaniem PSU jest przefiltrowanie zbioru usług (prostych i złożonych) pod kątem spełnialności wymagań funkcjonalnych, niefunkcjonalnych, oraz preferencji klienta celem wyboru usługi w najwyższym stopniu spełniającej jego wymagania. W ramach usługi wyodrębnia się trzy moduły:

- moduł wyboru usług spełniających wymagania funkcjonalne;
- moduł wyboru usług spełniających wymagania niefunkcjonalne;
- moduł wyboru usługi odpowiadającej preferencjom klienta.

Pierwszy z wymienionych modułów dokonuje filtracji dostępnych w repozytorium usług (atomowych i złożonych), poprzez wybranie tych, które spełniają wymagania funkcjonalne. Lista usług spełniających wymagania funkcjonalne jest następnie przekazana do modułu szeregowania ze względu na wymagania niefunkcjonalne, gdzie każdej z usług zostaje przypisany wskaźnik dopasowania do wymagań niefunkcjonalnych. W ostatnim z wymienionych modułów wybierana jest usługa najlepiej dopasowana do profilu obiektu generującego żądanie (użytkownika, innej usługi, procesu). Ostatni z modułów wspomagany jest poprzez zastosowanie opracowanych w ramach rozprawy metod klasyfikacji dla danych niezbalansowanych.

Problem nadania priorytetów klientom dla usług o ograniczonej liczbie wykonań odnosi się do sytuacji w której pewna grupa usług często cieszących się znacznie większym zainteresowaniem wśród klientów nie może być udostępniona wszystkim zainteresowanym obiektom. Przyczyną takiej sytuacji może być ograniczona liczba zasobów obliczeniowych, czy też ograniczona liczba wywołań pewnych usług, np. usług zapisu na kursy w ramach studiów [103]. Koniecznie jest więc opracowanie skutecznych metod klasyfikacji pozwalających na konstrukcję modeli nadających priorytety klientom w trybie nadzorowanym.

W tym obszarze zasadne okazuje się być wykorzystanie opisanych w ramach rozprawy wzmacnianych klasyfikatorów SVM.

6.6 Dyskusja

W niniejszym rozdziale zaproponowano architekturę udostępniania rozwiązań eksploracji danych z wykorzystaniem usług Webowych. W ramach zestawu usług udostępniono podstawowe metody klasyfikacji, oraz metody opracowane w ramach rozprawy doktorskiej dedykowane do rozwiązania problemu niezbalansowania. Opracowane rozwiązania zostały wdrożone w przedsiębiorstwie zajmującym się projektowaniem systemów informatycznych dla instytucji finansowych. Na podstawie analizy jakości metod z wykorzystaniem rzeczywistego zbioru danych odrzucono możliwość wykorzystania typowych metod klasyfikacji, na rzecz opracowanych w ramach rozprawy metod rozwiązujących problem niezbalansowania danych. Dodatkowo, dokonano analizy możliwości zastosowania metod udostępnionych jako usługi Webowe do problemu detekcji anomalii w sieciach.

Rozdział 7

Uwagi końcowe

W pracy postawiono następującą tezę badawczą:

„Zastosowanie zespołów klasyfikatorów SVM zwiększa skuteczność klasyfikacji w zadaniach o niezbalansowanym zbiorze uczącym.”

Wykazano słuszność tezy pracy poprzez zaproponowanie wzmacnianego klasyfikatora SVM (*BoostingSVM-IB*) i jego dwóch modyfikacji (*BoostingSVM-IB.M1* i *BoostingSVM-IB.M2*) charakteryzujących się wyższą jakością predykcji, niż inne metody dedykowane do rozwiązania problemu niezbalansowania. Wysoka jakość rozwiązań potwierdzona została trzema zastosowaniami praktycznymi:

- zastosowaniem metod do predykcji przeżywalności pooperacyjnej,
- zastosowaniem rozwiązań do oceny ryzyka kredytowego,
- zastosowaniem algorytmów do detekcji anomalii w sieciach.

7.1 Oryginalny wkład w obszarze uczenia maszynowego

Do najważniejszych osiągnięć pracy stanowiących wkład w obszar uczenia maszynowego należy zaliczyć:

- **Opracowanie wzmacnianego klasyfikatora SVM dla danych niezbalansowanych i jego dwóch modyfikacji**. W ramach rozprawy opracowano wysokiej jakości metodę rozwiązującą problem dysproporcji w zbiorze uczącym. Opracowany algorytm klasyfikacji łączył w sobie dwie metody charakteryzujące się wysoką jakością predykcji: zespoły klasyfikatorów, oraz model typu SVM. Zaproponowano wrażliwą na koszt technikę uczenia proponowanego klasyfikatora wykorzystującą mechanizmy wzmacniania, oraz zmodyfikowaną procedurę SMO do konstrukcji bazowych klasyfikatorów SVM. Wykazano ponadto, że opracowany algorytm *BoostingSVM-IB* minimalizuje ważoną, wykładniczą funkcję błędu. Zaproponowano dwie modyfikacje metody wykorzystujące metody eliminacji obserwacji nieinformacyjnych ze zbioru uczącego. Wykonano badania empiryczne mające na celu potwierdzenie wysokiej jakości proponowanych metod poprzez porównanie ich wyników działania z rezultatami osiągniętymi przez referencyjne metody dedykowane do rozwiązania problemu niezbalansowania. Wysoka jakość opracowanych rozwiązań potwierdzona została szeregiem testów statystycznych.

- **Rozwiązanie problemu predykcji przeżywalności pooperacyjnej z wykorzystaniem opracowanych rozwiązań**. W pracy postawiono problem z dziedziny analizy ryzyka operacyjnego, który został rozwiązany poprzez zastosowanie proponowanych algorytmów na rzeczywistych, wcześniej niepublikowanych danych pozyskanych od instytucji medycznych. Zakres prac związanych z zastosowaniem objął również opracowanie metody ekstrakcji reguł decyzyjnych z modelu „czarnej skrzynki”, oraz analizę jakości metod wstawiania brakujących wartości atrybutów.

- **Opracowanie architektury udostępniania usług przetwarzania danych zgodniej z paradygmatem SOA**. Zaproponowano paradygmat konstruowania usług przetwarzania danych z wykorzystaniem metod uczenia maszynowego celem komercjalizacji opracowanych w rozprawie metod. Proponowane algorytmy dla danych niezbalansowanych zostały wdrożone w firmie zajmującej się projektowaniem systemów dla

instytucji finansowych z wykorzystaniem proponowanego paradygmatu udostępniania rozwiązań uczenia maszynowego poprzez usługi.

Wybranie zagadnienia będące przedmiotem rozprawy były przedmiotem następujących artykułów: [84, 104, 148, 150, 151, 152, 153].

7.1.1 Proponowane kierunki dalszych prac

Przeprowadzona analiza przedmiotu oraz otrzymane wyniki prowadzą do wskazania następujących, dalszych kierunków prac:

1. Wzbogacenie opracowanych w ramach rozprawy algorytmów uczenia wzmacnianych klasyfikatorów SVM dla danych niezbalansowanych o mechanizmy rozwiązujące problem brakujących wartości atrybutów inne niż przedstawione w rozprawie techniki wstawiania.

2. Zaproponowanie algorytmów uczenia wzmacnianych klasyfikatorów SVM działających w trybie przyrostowym.

3. Opracowanie wersji algorytmu uczenia wzmacnianych klasyfikatorów SVM działającej w trybie częściowo nadzorowanym.

4. Opracowanie metody uczenia zespołów klasyfikatorów, która minimalizuje inne niż wykładniczy, ważony błąd klasyfikacji kryterium niezbalansowania danych.

Bibliografia

[1] J. Alcalá, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, 17:255–287, 2010. [cytowanie na str. 61, 89]

[2] E.L. Allwein, R.E. Schapire, Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *The Journal of Machine Learning Research*, 1:113–141, 2001. [cytowanie na str. 58]

[3] U. Aydogmus, L. Cansever, Y. Sonmezoglu, K. Karapinar, C. I. Kocaturk, M. A. Bedirhan. The impact of the type of resection on survival in patients with n1 non-small-cell lung cancers. *European Journal of Cardio-Thoracic Surgery*, 37:446–450, 2010. [cytowanie na str. 76]

[4] K. Bache and M. Lichman. UCI machine learning repository, 2013. [cytowanie na str. 98]

[5] N. Barakat, J. Diederich. Eclectic rule-extraction from support vector machines. *International Journal of Computational Intelligence*, 2(1):59–62, 2005. [cytowanie na str. 77]

[6] A. Barua, S. D. Handagala, L. Socci, B. Barua, M. Malik, N. Johnstone, A. E. Martin-Ucar. Accuracy of two scoring systems for risk stratification in thoracic surgery. *Interactive Cardiovascular and Thoracic Surgery*, 14(5):556–559, 2012. [cytowanie na str. 76]

[7] G. E. Batista, M. C. Monard. An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, 17(5-6):519–533, 2003. [cytowanie na str. 85]

[8] G. E. Batista, R. C. Prati, M. C. Monard. A study of the behaviour of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1):20–29, 2004. [cytowanie na str. 30]

[9] K. P. Bennett, A. Demiriz. Semi-supervised support vector machines. In *Advances in Neural Information Processing Systems*, s. 368–374. MIT Press, 1998. [cytowanie na str. 97]

[10] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. [cytowanie na str. 5, 15, 40, 41, 57, 58]

[11] A.P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997. [cytowanie na str. 11]

[12] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, August 1996. [cytowanie na str. 22, 23]

[13] L. Breiman. Randomizing outputs to increase prediction accuracy. *Machine Learning*, 40(3):229–242, 2000. [cytowanie na str. 23, 26]

[14] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. [cytowanie na str. 22, 25]

[15] L. Breiman, J. Friedman, C. J. Stone, R. A. Olshen. *Classification and Regression Trees*. Chapman and Hall/CRC, 1984. [cytowanie na str. 15, 19]

[16] P. F Brown, R. Metz, B. A. Hamilton. Reference model for service oriented architecture 1.0. Raport techniczny, http://docs. oasis-open. org/soa-rm/v1. 0/soa-rm. pdf, 2005. [cytowanie na str. 89]

[17] K. Brzostowski, M. Zięba. Analysis of human arm motions recognition algorithms for system to visualize virtual arm. In *21st International Conference on Systems Engineering*, s. 422–426. IEEE, 2011. [cytowanie na str. 3]

[18] Z. Bubnicki. *Analysis and Decision Making in Uncertain Systems*. Springer, 2004. [cytowanie na str. 8, 15, 17]

[19] D. B. Edelman, C. T. Lyn, J. N. Crook. *Credit scoring and its applications*. Society for Industrial and Applied Mathematics, 2002. [cytowanie na str. 93]

[20] J. Cendrowska. Prism: An algorithm for inducing modular rules. *International Journal of Man-Machine Studies*, 27:349–370, 1987. [cytowanie na str. 15, 17]

[21] P. Chan, S. Stolfo. Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, s. 164–168. AAAI Press, 1998. [cytowanie na str. 32]

[22] V. Chandola, A. Banerjee, V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15, 2009. [cytowanie na str. 97]

[23] E.Y. Chang, B. Li, G. Wu, K. Goh. Statistical learning for effective visual information retrieval. In *IEEE International Conference on Image Processing*, s. 609–612. IEEE, 2003. [cytowanie na str. 31]

[24] N. V. Chawla, K. W. Bowyer, L. O. Hall. SMOTE : Synthetic Minority Over-sampling TEchnique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. [cytowanie na str. 30]

[25] N. V. Chawla, A. Lazarevic, L. O. Hall, K. Bowyer. SMOTEBoost: Improving prediction of the minority class in boosting. In *Proceedings of the Principles of Knowledge Discovery in Databases, PKDD-2003*, s. 107–119. Springer, 2003. [cytowanie na str. 23, 26, 31, 63]

[26] S. Chen, H. He, E.A. Garcia. RAMOBoost: Ranked minority oversampling in boosting. *Neural Networks, IEEE Transactions on*, 21(10):1624–1642, 2010. [cytowanie na str. 23, 26, 31]

[27] J. Chorowski. *Learning understandable classifier models*. rozprawa doktorska, Wroclaw University of Technology, 2011. [cytowanie na str. 77, 78]

[28] J. Chorowski, J. M. Zurada. Extracting rules from neural networks as decision diagrams. *Neural Networks, IEEE Transactions on*, 22(12):2435–2446, 2011. [cytowanie na str. 77]

[29] W. W. Cohen. Fast effective rule induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, s. 115–123. Morgan Kaufmann, 1995. [cytowanie na str. 15, 18, 78]

[30] M. Craven, J. Shavlik. Rule extraction: Where do we go from here? Raport techniczny, University of Wisconsin Machine Learning Research Group Working Paper, 1999. [cytowanie na str. 77]

[31] J. A. Chambers, D. P. Mandic. *Recurrent neural networks for predicition* . A John Wiley and Sons, Inc. Publication, 2001. [cytowanie na str. 15]

[32] M. de Sa. *Pattern Recognition*. Springer, 2001. [cytowanie na str. 3]

[33] T. G. Dietterich. Machine learning for sequential data: A review. In *Structural, Syntactic, and Statistical Pattern Recognition*, s. 15–30. Springer, 2002. [cytowanie na str. 8]

[34] J. X. Dong, A. Krzyżak, C.Y. Suen. A practical SMO algorithm. In *Proc. Int. Conf. on Pattern Recognition*, 3, 2002. [cytowanie na str. 59]

[35] J. Dowie, M. Wildman. Choosing the surgical mortality threshold for high risk patients with stage la non-small cell lung cancer: Insights from decision analysis. *Thorax*, 57:7–10, 2002. [cytowanie na str. 76]

[36] C. Drummond, R.C. Holte. Exploiting the cost (in)sensitivity of decision tree splitting criteria. In *Proceedings of the Seventeenth International Conference on Machine Learning*, s. 239–246. Morgan Kaufmann, 2000. [cytowanie na str. 33]

[37] C. Elkan. The foundations of cost-sensitive learning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, s. 973–978. Lawrence Erlaum Associates, LTD, 2001. [cytowanie na str. 33]

[38] S. Ertekin, J. Huang, L. Bottou, L. Giles. Learning on the border: active learning in imbalanced data classification. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, s. 127–136. ACM, 2007. [cytowanie na str. 32, 54]

[39] S. Ertekin, J. Huang, C.L. Giles. Active learning for class imbalance problem. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, s. 823–824. ACM, 2007. [cytowanie na str. 32, 54]

[40] H. Esteva, T. G. Núñez, R. O. Rodríguez. Neural networks and artificial intelligence in thoracic surgery. *Thoracic Surgery Clinics*, 17(3):359–367, 2007. [cytowanie na str. 76]

[41] P. E. Falcoz, M. Conti, L. Brouchet, S. Chocron, M. Puyraveau, M. Mercier, J. P. Etievent, M. Dahan. The Thoracic Surgery Scoring System (Thoracoscore): risk model for in-hospital death in 15,183 patients requiring thoracic surgery. *The Journal of Thoracic and Cardiovascular Surgery*, 133(2):325–332, 2007. [cytowanie na str. 76]

[42] W. Fan, S.J. Stolfo, J. Zhang, P.K. Chan. AdaCost: misclassification cost-sensitive boosting. In *Proc. 16th International Conf. on Machine Learning*, s. 97–105. Morgan Kaufmann, 1999. [cytowanie na str. 22, 24, 33, 63]

[43] M. K. Ferguson, J. Siddique, T. Karrison. Modeling major lung resection outcomes using classification trees and multiple imputation techniques. *European Journal of Cardio-Thoracic Surgery*, 34(5):1085–1089, 2008. [cytowanie na str. 76]

[44] A. Fernández, J. Luengo, J. Derrac, J. Alcalá-Fdez, F. Herrera. Implementation and integration of algorithms into the KEEL data-mining software tool. In *Intelligent Data Engineering and Automated Learning*, s. 562–569. Springer, 2009. [cytowanie na str. 61]

[45] V. Fiolet, R. Olejnik, G. Lefait, B. Toursel. Optimal grid exploitation algorithms for data mining. In *Proceedings of The Fifth International Symposium on Parallel and Distributed Computing*, s. 246–252. IEEE, 2006. [cytowanie na str. 89]

[46] C.A. Floudas, P.M. Pardalos (Editors). *Encyclopedia of Optimization*. Springer, 2009. [cytowanie na str. 41]

[47] Y. Freund, R. E. Schapire, M. Hill. Experiments with a New Boosting Algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, 1996. [cytowanie na str. 22, 24]

[48] J. Friedman, T. Hastie, R. Tibshirani. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics*, 28(2):337–407, 2000. [cytowanie na str. 24]

[49] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera. A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches. *IEEE Systems, Man, and Cybernetics Society*, 42(4):3358–3378, 2012. [cytowanie na str. 20, 29, 31, 62, 63, 68]

[50] P. J. Garcia-Laencina, J. L. Sancho-Gomez, A. R. Figueiras-Vidal. Pattern classification with missing data: a review. *Neural Computing and Applications*, 19(2):263–282, September 2009. [cytowanie na str. 8, 11, 16, 20, 76, 84]

[51] E. A. Gehan. A generalized Wilcoxon test for comparing arbitrarily singly-censored samples. *Biometrika*, 52(1-2):203–223, 1965. [cytowanie na str. 66]

[52] A. Grzech, K. Juszczyszyn, P. Stelmach, Ł. Falas. Link prediction in dynamic networks of services emerging during deployment and execution of web services. In *Computational Collective Intelligence. Technologies and Applications*, s. 109–120. Springer, 2012. [cytowanie na str. 90]

[53] H. Guo, H. L. Viktor. Learning from imbalanced data sets with boosting and data generation: the DataBoost-IM approach. *ACM SIGKDD Explorations Newsletter*, 6(1):30–39, 2004. [cytowanie na str. 23, 26, 31]

[54] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009. [cytowanie na str. 61]

[55] L.K. Hansen, P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990. [cytowanie na str. 22]

[56] H. He, E. A. Garcia. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, September 2009. [cytowanie na str. 8, 20, 29, 31, 33]

[57] K. A. Heller, Z. Ghahramani. Bayesian hierarchical clustering. In *Proceedings of the 22nd international conference on Machine learning*, s. 297–304. ACM, 2005. [cytowanie na str. 97]

[58] S. Hido, H. Kashima, Y. Takahashi. Roughly balanced bagging for imbalanced data. *Statistical Analysis and Data Mining*, 2(5-6):412–426, 2009. [cytowanie na str. 32]

[59] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, s. 65–70, 1979. [cytowanie na str. 65]

[60] F. Honghai, C. Guoshun, Y. Cheng, Y/ Bingru, C. Yumei. A SVM regression based approach to filling in missing values. In *Knowledge-Based Intelligent Information and Engineering Systems*, s. 581–587. Springer, 2005. [cytowanie na str. 85]

[61] N. C. Hsieh, L. P. Hung. A data driven ensemble classifier for credit scoring analysis. *Expert Systems with Applications*, 37(1):534–545, January 2010. [cytowanie na str. 94]

[62] S. Hu, Y. Liang, L. Ma, Y. He. MSMOTE: improving classification performance when training data is imbalanced. In *Second International Workshop on Computer Science and Engineering.*, s. 13–17. IEEE, 2009. [cytowanie na str. 31]

[63] J. Huang, C.X. Ling. Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):299–310, 2005. [cytowanie na str. 11]

[64] H. Hui, W. Wang, B. Mao. Borderline-SMOTE : A New Over-Sampling Method in Imbalanced Data Sets Learning. In *Advances in Intelligent Computing*, s. 878 – 887. 2005. [cytowanie na str. 30]

[65] P. Icard, M. Heyndrickx, L. Guetti, F. Galateau-Salle, P. Rosat, J. P. Le Rochais, J. L. Hanouz. Morbidity, mortality and survival after 110 consecutive bilobectomies over 12 years. *Interactive Cardiovascular and Thoracic Surgery*, 16(2):179–185, 2013. [cytowanie na str. 76]

[66] T. Jo, N. Jopkowicz. Class imbalances versus small disjuncts. *ACM SIGKDD Explorations Newsletter*, 6(1):40–49, 2004. [cytowanie na str. 31]

[67] M.V. Joshi, V. Kumar, R.C. Agarwal. Evaluating boosting algorithms to classify rare classes: Comparison and improvements. In *Proceedings of IEEE International Conference on Data Mining*, s. 257–264. IEEE, 2001. [cytowanie na str. 33]

[68] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, K.R.K. Murthy. Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation*, 13(3):637–649, 2001. [cytowanie na str. 41]

[69] H. Kim, G. H. Golub, H. Park. Missing value estimation for DNA microarray gene expression data: local least squares imputation. *Bioinformatics*, 21(2):187–198, 2005. [cytowanie na str. 85]

[70] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of International Joint Conference on Artificial Intelligence*, s. 1137–1145. Lawrence Erlbaum Associates Ltd, 1995. [cytowanie na str. 61]

[71] G. Kołaczek, K. Juszczyszyn. Traffic pattern analysis for distributed anomaly detection. *Parallel Processing and Applied Mathematics*, 7204:648–657, 2012. [cytowanie na str. 97]

[72] G. Kołaczek, A. Prusiewicz. Anomaly detection system based on service oriented architecture. *Intelligent Information and Database Systems*, 7198:376–385, 2012. [cytowanie na str. 97]

[73] M. Krzyśko, W. Wolyński, T. Górecki, M. Skorzybut. *Systemy uczące się* . WNT Warszawa, 2008. [cytowanie na str. 15, 16, 19, 39, 40, 41]

[74] M. Kubat, S. Matwin. Addressing the curse of imbalanced training sets: one-sided selection. In *Proceedings of the Fourteenth International Conference on Machine Learning*, s. 179–186. Morgan Kaufmann, 1997. [cytowanie na str. 54]

[75] M. Kukar, I. Kononenko. Cost-sensitive learning with neural networks. In *Proceedings of the 13th European Conference on Artificial Intelligence*, s. 445–449. John Wiley and Sons, 1998. [cytowanie na str. 33]

[76] L. I. Kuncheva. *Combining Pattern Classifiers*. A John Wiley and Sons, Inc. Publication, 2004. [cytowanie na str. 19, 28]

[77] L.I. Kuncheva, C.J. Whitaker. Ten Measures of Diversity in Classifier Ensembles: Limits for Two Classifiers. In *Proceedings of IEE Workshop on Intelligent Sensor Processing*, s. 1–10, 2001. [cytowanie na str. 21]

[78] M. Kurzyński. *Rozpoznawanie obiektów - metody statystyczne*. Oficyna Wydawnicza Politechniki Wrocławskiej, 1997. [cytowanie na str. 2, 14, 19]

[79] C. Y. Lee, Z. J. Lee. A novel algorithm applied to classify unbalanced data. *Applied Soft Computing*, 12:2481—-2485, 2012. [cytowanie na str. 79]

[80] C. Li. Classifying imbalanced data using a bagging ensemble variation (bev). In *Proceedings of the 45th Annual Southeast Regional Conference*, s. 203–208. ACM, 2007. [cytowanie na str. 32]

[81] D. Li, J. Deogun, W. Spaulding, B. Shuart. Towards missing data imputation: A study of fuzzy k-means clustering method. *Rough Sets and Current Trends in Computing*, 3066:573–579, 2004. [cytowanie na str. 85]

[82] K. L. Li, H. K. Huang, S. F. Tian, W Xu. Improving one-class SVM for anomaly detection. In *International Conference on Machine Learning and Cybernetics*, s. 3077–3081. IEEE, 2003. [cytowanie na str. 97]

[83] X. Li, L. Wang, E. Sung. AdaBoost with SVM-based component classifiers. *Engineering Applications of Artificial Intelligence*, 21(5):785–795, 2008. [cytowanie na str. 22]

[84] M. Lubicz, M. Zięba, K. Pawełczyk, A. Rzechonek, J. Kołodziej. Modele eksploracji danych niezbalansowanych - procedury klasyfikacji dla zadania analizy ryzyka operacyjnego. *Prace Naukowe Uniwersytetu Ekonomicznego we Wrocławiu (w druku)*, (–):–, 2013. [cytowanie na str. 76, 104]

[85] L.M. Manevitz, M. Yousef. One-class SVMs for document classification. *The Journal of Machine Learning Research*, 2:139–154, 2002. [cytowanie na str. 32]

[86] J. Mani, I. Zhang. KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction. In *Proceedings of International Conference on Machine Learning, Workshop Learning from Imbalanced Data Sets*, 2003. [cytowanie na str. 30]

[87] G. Martínez-Muñoz, A. Suárez. Switching class labels to generate classification ensembles. *Pattern Recognition*, 38(10):1483–1494, 2005. [cytowanie na str. 23, 26]

[88] MathWorks. Machine Learning Toolbox for Matlab. URL http://www.mathworks.com/discovery/machine-learning.html. [cytowanie na str. 89]

[89] P. Melville, R. J. Mooney. Constructing Diverse Classifier Ensembles using Artificial Training Examples. In *Proceedings of the International Jointed Conference on Artificial Intelligence*, s. 505–510, 2003. [cytowanie na str. 20, 23, 26]

[90] Microsoft. Business intelligence. URL http://www.microsoft.com/en-us/bi/default.aspx. [cytowanie na str. 89]

[91] G. Migut. Modelowanie ryzyka kredytowego. In *Materiały Konferencyjne „Zastosowanie Statystyki i Data Mining w Finansach", Warszawa*, s. 39–54, 2003. [cytowanie na str. 94]

[92] K. Morik, P. Brockhausen, T. Joachims. Combining statistical learning with a knowledge-based approach-a case study in intensive care monitoring. In *Proceedings of International Conference on Machine Learning*, s. 268–277. Morgan Kaufmann, 1999. [cytowanie na str. 33, 36, 59]

[93] E. Newcomer, G. Lomow. *Understanding SOA with web services (independent technology guides)*. Addison-Wesley Professional, 2004. [cytowanie na str. 89]

[94] H. Núñez, C. Angulo, A. Català. Rule extraction from Support Vector Machines. In *Proceedings of the European Symposium on Artificial Neural Networks*, s. 107–112, 2002. [cytowanie na str. 77]

[95] S. Oba, M. Sato, I. Takemasa, M. Monden, K. Matsubara, S. Ishii. A bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, 19(16):2088–2096, 2003. [cytowanie na str. 85]

[96] S. Oh, M.S. Lee, B.T. Zhang. Ensemble learning with active example selection for imbalanced biomedical data classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 8(2):316–325, 2011. [cytowanie na str. 54]

[97] E. Osuna, R. Freund, and F. Girosi. An improved training algorithm for support vector machines. In *Proceedings of the IEEE Workshop of Neural Networks for Signal Processing*, s. 276–285. IEEE, 1997. [cytowanie na str. 41]

[98] J.C. Platt. Fast Training of Support Vector Machines using Sequential Minimal Optimization. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998. [cytowanie na str. 41, 42, 43]

[99] J.C. Platt, N. Cristianini, J. Shawe-Taylor. Large margin dags for multiclass classification. *Advances in neural information processing systems*, 12(3):547–553, 2000. [cytowanie na str. 58]

[100] R. Polikar, J. DePasquale, H. S. Mohammed, G. Brown, L. I. Kuncheva. Learn++.MF: A random subspace approach for the missing feature problem. *Pattern Recognition*, 43(11):1–16, 2010. [cytowanie na str. 20, 22, 26]

[101] StatSoft Polska. Statistica. URL http://www.statsoft.pl/. [cytowanie na str. 89]

[102] A. Prusiewicz, M. Zięba. Services Recommendation in Systems Based on Service Oriented Architecture by Applying Modified ROCK Algorithm. *Communications in Computer and Information Science*, 88(2):226–238, 2010. [cytowanie na str. 93]

[103] A. Prusiewicz, M. Zięba. On some method for limited services selection. *International Journal of Intelligent Information and Database Systems*, 5(5):493–509, 2011. [cytowanie na str. 100]

[104] A. Prusiewicz, M. Zięba. The proposal of service oriented data mining system for solving real-life classification and regression problems. In *Technological Innovation for Sustainability*, s. 83–90. Springer, 2011. [cytowanie na str. 89, 104]

[105] J. R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1(1):81–106, 1986. [cytowanie na str. 15, 19]

[106] J. R. Quinlan. C4.5: Programs for machine learning. *Machine Learning*, 16:235–240, 1994. [cytowanie na str. 15, 19]

[107] G. Rocco. eComment. Re: Accuracy of two scoring systems for risk stratification in thoracic surgery. *Interactive CardioVascular and Thoracic Surgery*, 14(5):559–559, 2012. [cytowanie na str. 76]

[108] R. Rojas. *Neural Networks - A Systematic Introduction*. Springer, 1996. [cytowanie na str. 15]

[109] G. Santos-Garcıa, G. Varela, N. Novoa, M. F. Jiménez. Prediction of postoperative morbidity after lung resection using an artificial neural network ensemble. *Artificial Intelligence in Medicine*, 30(1):61–69, 2004. [cytowanie na str. 76]

[110] R.E. Schapire, Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine learning*, 37(3):297–336, 1999. [cytowanie na str. 22]

[111] T. Schneider. Analysis of incomplete climate data: Estimation of mean values and covariance matrices and imputation of missing values. *Journal of Climate*, 14(5):853–871, 2001. [cytowanie na str. 84]

[112] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, R.C. Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001. [cytowanie na str. 32]

[113] C. Seiffert, T.M. Khoshgoftaar, J. Van Hulse, A. Napolitano. RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 40(1):185–197, 2010. [cytowanie na str. 31, 63]

[114] B. Settles. Active learning literature survey. Computer Sciences, Raport techniczny, University of Wisconsin–Madison, 2009. [cytowanie na str. 54]

[115] D.M. Shahian, F.H. Edwards. Statistical risk modeling and outcomes analysis. *Annals of Thoracic Surgery*, 86:1717–1720, 2008. [cytowanie na str. 76]

[116] M. Shapiro, S. J Swanson, C. D. Wright, C. Chin, S. Sheng, J. Wisnivesky, T. S. Weiser. Predictors of major morbidity and mortality after pneumonectomy utilizing the Society for Thoracic Surgeons General Thoracic Surgery Database. *The Annals of thoracic surgery*, 90(3):927–935, 2010. [cytowanie na str. 76]

[117] J. Sobczak. Analiza zdolności kredytowej. In *Innowacyjne Rozwiązania Biznesowe, Łódź*, s. 107–114, 2005. [cytowanie na str. 94]

[118] P. Stelmach, A. Grzech, K. Juszczyszyn. A Model for Automated Service Composition System in SOA Environment. In *Technological Innovation for Sustainability*, s. 75–82. Springer, 2011. [cytowanie na str. 88]

[119] Y. Sun, M. Kamel, A. Wong, Y. Wang. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 40(12):3358–3378, 2007. [cytowanie na str. 20, 22, 24, 33]

[120] J. Świątek. *Wybrane zagadnienia identyfikacji statycznych systemów złożonych* . Oficyna Wydawnicza Politechniki Wrocławskiej, 2009. [cytowanie na str. 15]

[121] R. Tadeusiewicz, M. Flasiński. *Rozpoznawanie obrazów*. Państwowe wydawnictwo naukowe, 1991. [cytowanie na str. 2, 8]

[122] Y. Tang, B. Jin, Y. Q. Zhang. Granular support vector machines with association rules mining for protein homology prediction. *Artificial Intelligence in Medicine*, 35(1-2):121–134, 2005. [cytowanie na str. 32]

[123] Y. Tang, B. Jin, Y. Q. Zhang, H. Fang, B. Wang. Granular support vector machines using linear decision hyperplanes for fast medical binary classification. In *The IEEE International Conference on Fuzzy Systems*, s. 138–142. IEEE, 2005. [cytowanie na str. 32]

[124] Y. Tang, Y. Q. Zhang. Granular SVM with repetitive undersampling for highly imbalanced protein homology prediction. In *IEEE International Conference on Granular Computing*, s. 457–460. IEEE, 2006. [cytowanie na str. 32]

[125] Y. Tang, Y.Q. Zhang, N.V. Chawla, S. Krasser. Svms modeling for highly imbalanced classification. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 39(1):281–288, 2009. [cytowanie na str. 62, 63]

[126] Y. Tang, Y.Q. Zhang, Z. Huang. Development of two-stage SVM-RFE gene selection strategy for microarray expression data analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4(3):365–381, 2007. [cytowanie na str. 32]

[127] D. Tao, X. Tang, X. Li, X. Wu. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1088–1099, 2006. [cytowanie na str. 31, 63]

[128] Y. W. Teh, H. D. Iii, D. Roy. Bayesian agglomerative clustering with coalescents. In *Advances in Neural Information Processing Systems*. MIT Press, 2008. [cytowanie na str. 97]

[129] A.B. Tickle, R. Andrews, M. Golea, J. Diederich. The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural networks. *IEEE Transactions on Neural Networks*, 9(6):1057–1068, 1998. [cytowanie na str. 77]

[130] K.M. Ting. A comparative study of cost-sensitive boosting algorithms. In *Proceedings of the 17th International Conference on Machine Learning*. Morgan Kaufmann, 2000. [cytowanie na str. 33]

[131] J. M. Tomczak, J. Świątek, K. Brzostowski. Bayesian classifiers with incremental learning for nonstationary datastreams. *Advances in System Science*, 1:251–260, 2010. [cytowanie na str. 16]

[132] I. Tomek. Two Modifications of CNN. *IEEE Transactions on Systems, Man and Cybernetics*, 6(11):769–772, 1976. [cytowanie na str. 30, 52]

[133] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, R. B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, 2001. [cytowanie na str. 85]

[134] G. Tsoumakas, N. Bassiliades, I. Vlahavas. A knowledge-based web information system for the fusion of distributed classifiers. In *Web information systems*, s. 271–308. Idea Group Publishing, 2004. [cytowanie na str. 89]

[135] V.N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, Inc., 1998. [cytowanie na str. 15, 16, 40, 41]

[136] K. Veropoulos, C. Campbell, N. Cristianini. Controlling the sensitivity of support vector machines. In *Proceedings of the International Joint Conference on Artificial Intelligence*, s. 55–60, 1999. [cytowanie na str. 33, 36, 59, 62]

[137] B. X. Wang, N. Japkowicz. Boosting support vector machines for imbalanced data sets. *Knowledge and Information Systems*, 25(1):1–20, 2010. [cytowanie na str. 20, 27, 33, 47, 59]

[138] G. Wang, J. Hao, J. Ma, H. Jiang. A comparative assessment of ensemble learning for credit scoring. *Expert Systems with Applications*, 38(1):223–230, January 2011. [cytowanie na str. 94]

[139] S. Wang, X. Yao. Diversity analysis on imbalanced data sets by using ensemble models. In *IEEE Symposium on Computational Intelligence and Data Mining*, s. 324–331. IEEE, 2009. [cytowanie na str. 31, 63]

[140] Weka Machine Learning Project. Weka. URL http://www.cs.waikato.ac.nz/˜ml/weka. [cytowanie na str. 89]

[141] I. Witten, E. Frank. *Data Mining*. Elsevier, 2005. [cytowanie na str. 8, 15, 17]

[142] D. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992. [cytowanie na str. 28]

[143] R. Yan, Y. Liu, R. Jin, A. Hauptmann. On predicting rare classes with SVM ensembles in scene classification. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, s. 21–34. IEEE, 2003. [cytowanie na str. 32]

[144] Y. Yang, J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the International Conference on Machine Learning*, s. 412–420. Morgan Kaufmann, 1997. [cytowanie na str. 79]

[145] X. Zhu, Z. Ghahramani, J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of International Conference on Machine Learning*, volume 20, s. 912. Morgan Kaufmann, 2003. [cytowanie na str. 97]

[146] L. Zhuang, H. Dai. Parameter estimation of one-class SVM on imbalance text classification. In *Advances in Artificial Intelligence*, s. 538–549. Springer, 2006. [cytowanie na str. 32]

[147] L. Zhuang, H. Dai. Parameter optimization of kernel-based one-class classifier on imbalance learning. *Journal of computers*, 1(7):32–40, 2006. [cytowanie na str. 32]

[148] M. Zięba, J. Błaszczyk, J. Kołodziej, K. Pawełczyk, M. Lubicz, A. Rzechonek. Analiza porównawcza wybranych technik eksploracji danych do klasyfikacji danych medycznych z brakującymi obserwacjami. *Prace Naukowe Uniwersytetu Ekonomicznego we Wrocławiu*, (242):416–425, 2012. [cytowanie na str. 76, 84, 104]

[149] M. Zięba. *Multistage neural networks for pattern recognition*. Lambert Publishing, 2010. [cytowanie na str. 3, 15]

[150] M. Zięba. *Ensemble decision trees: Ensemble decision trees for customer classification in service oriented systems*. Lambert Publishing, 2012. [cytowanie na str. 23, 26, 93, 94, 104]

[151] M. Zięba, M. Lubicz. Performance of Classifiers For Missing Data In Thoracic Surgery Risk Modelling Using Weka-Based Data Mining Approaches. *Information Systems Architecture and Technology: IT Models in Management Process*, s. 435–445, 2010. [cytowanie na str. 76, 104]

[152] M. Zięba, J. Świątek. Ensemble classifier for solving credit scoring problems. *Technological Innovation for Value Creation*, s. 59–66, 2012. [cytowanie na str. 23, 26, 94, 104]

[153] M. Zięba, J. Świątek. Various methods of combining classifiers for ensemble algorithms. *Applications of System Science*, 1:81–91, 2010. [cytowanie na str. 19, 27, 28, 104]

Spis symboli i skrótów

Symbol/skrót	Opis
$\mathbf{x} \in \mathbb{X}$	wektor cech
D	wymiar wektora cech
x_d	wartość pojedynczej cechy obiektu
$y \in \mathbb{Y}$	klasa obiektu
Ψ	klasyfikator
$\mathbb{D}_{\mathbf{x}}^{(y)}$	obszar decyzyjny
$\mathbb{S}_N$	zbiór uczący
$(\mathbf{x}_n, y_n)$	n-ty element zbioru uczącego
N	liczba elementów w zbiorze uczącym
$f(\cdot)$	funkcja dyskryminująca
$\mathbf{a}, b$	parametry funkcji dyskryminującej
$\mathbf{X}$	zmienna losowa dla wektora cech
$\mathbf{Y}$	zmienna losowa dla klasy
$p(\cdot)$	rozkład prawdopodobieństwa
E	błąd klasyfikacji
E_{Imb}	błąd klasyfikacji dla danych niezbalansowanych
$\Psi_k^{(1)}$	klasyfikator bazowy dla zespołu klasyfikatorów
$\Psi^{(2)}$	klasyfikator łączący
$Q(\cdot)$	kryterium uczenia klasyfikatora SVM (sformułowanie prymalne)

Symbol/skrót	Opis
$Q_D(\cdot)$	kryterium uczenia klasyfikatora SVM (sformułowanie dualne)
ξ	zmienna pomocnicza, dodatkowa (uczenie SVM)
β_k	współczynnik wzmacniania (zespoły klasyfikatorów)
$D_k^{(n)}$	prawdopodobieństwo wylosowania n-tej obserwacji w k-tej iteracji konstrukcji zespołu klasyfikatorów
C	parametr kosztu związany z błędną klasyfikacją (SVM)
C_+	parametr kosztu związany z błędną klasyfikacją obserwacji pozytywnych (SVM)
C_-	parametr kosztu związany z błędną klasyfikacją obserwacji negatywnych (SVM)
N_+	liczność obserwacji z klasy pozytywnej
N_-	liczność obserwacji z klasy negatywnej
$\mathbb{N}_+$	zbiór indeksów obserwacji z klasy pozytywnej
$\mathbb{N}_-$	zbiór indeksów obserwacji z klasy negatywnej
H	hiperpłaszczyzna separująca (SVM)
H_+, H_-	hiperpłaszczyzny tworzące margines (SVM)
$\boldsymbol{\omega}$	wektor wag penaltyzacji (SVM)
$\boldsymbol{\lambda}, \boldsymbol{\gamma}$	wektory mnożników Lagrange'a (SVM)
$y(\cdot)$	wyjście klasyfikatora(SVM)
$K(\cdot, \cdot)$	funkcja jądra
$\phi(\cdot)$	przekształcenie nieliniowe
$E_i^{(SMO)}$	różnica rzeczywistego wyjścia i wyjścia aktualnego modelu SVM (algorytm SMO)
L	dolne ograniczenie mnożników (algorytm SMO)
U	górne ograniczenie mnożników (algorytm SMO)

Symbol/skrót	Opis
$\mathbf{w}_k$	wektor wag obserwacji w k-tym kroku konstrukcji klasyfikatora bazowych (zespoły klasyfikatorów)
e_k	znormalizowana funkcja błędu dla funkcji E_{Imb} (zespoły klasyfikatorów)
K_{final}	liczba skonstruowanych klasyfikatorów bazowych (zespoły klasyfikatorów)
E_{exp}	wykładnicza funkcja błędu
$E_{exp,Imb}$	wykładnicza funkcja błędu dla danych niezbalansowanych
$g_k(\cdot)$	kombinacja liniowa k klasyfikatorów bazowych (zespoły klasyfikatorów)
$\mathbb{T}_+$	zbiór indeksów obserwacji klasyfikowanych poprawnie do klasy pozytywnej
$\mathbb{T}_-$	zbiór indeksów obserwacji klasyfikowanych poprawnie do klasy negatywnej
$\mathbb{F}_+$	zbiór indeksów obserwacji klasyfikowanych błędnie do klasy pozytywnej
$\mathbb{F}_-$	zbiór indeksów obserwacji klasyfikowanych błędnie do klasy negatywnej
$\boldsymbol{\omega}$	wektor wag penaltyzacji w k-tej iteracji konstrukcji klasyfikatorów bazowych (zespoły klasyfikatorów SVM)
α	współczynnik generalizacji dla procesu uczenia (zespoły klasyfikatorów SVM)
$d(\cdot,\cdot)$	miara odległości
$\mathbb{S}_{N_+}$	zbiór uczący zawierający jedynie obserwacje z klasy pozytywnej
$\mathbb{S}_{N_-}$	zbiór uczący zawierający jedynie obserwacje z klasy negatywnej
N_{SVM}	liczba wektorów wspierających (SVM)

Symbol/skrót	Opis
N_{active}	liczba obserwacji wybranych w procesie uczenia aktywnego
$N_{active,SVM}$	liczba wektorów wspierających wyznaczonych na zredukowanym poprzez zastosowanie aktywnego uczenia zbiorze (SVM)
$\mathcal{H}_i$	hipoteza zerowa dot. równości median (test Wilcoxona)
θ_{BSI}	mediana wskaźnika *GMean* dla testowanej metody (test Wilcoxona)
θ_i	mediana wskaźnika *GMean* dla i-tej metody referencyjnej (test Wilcoxona)
$pval$	p-wartość (test Wilcoxona)
M	liczba testowanych hipotez (procedura Holma-Bonferroniego)
α_{ist}	poziom ufności (test Wilcoxona)
$FWER$	wskaźnik *familywise error rate* (procedura Holma-Bonferroniego)
R^+	suma rang dla testowanej metody
R^-	suma rang dla metody referencyjnej
$\tilde{\mathbb{S}}_N$	zbiór uczący po reetykiezacji
$\mathbb{R}$	zbiór reguł decyzyjnych
l	rozmiar szerokiego marginesu (zespoły klasyfikatorów SVM)
$Entr(\cdot)$	funkcja entropii
$InfGain(\cdot,\cdot)$	kryterium zysku informacyjnego
SVM	Support Vector Machnies
SMO	Sequential Minimal Optimization
SMO_{Imb}	algorytm SMO dla danych niezbalansowanych
$GMean$	wskaźnik średniej geometrycznej czułości i specyficzności
AUC	pole powierzchni pod krzywą ROC

Symbol/skrót	Opis
TP_{rate}	wskaźnik czułości
TN_{rate}	wskaźnik specyficzności
Acc	poprawność klasyfikacji
K-NN	K najbliższych sąsiadów
BSI	BoostingSVM-IB
BSI1	BoostingSVM-IB.M1
BSI2	BoostingSVM-IB.M2
SSVM	SVM z próbkowaniem metodą SMOTE
SBSVM	SMOTEBoostSVM
CSVM	Cost-sensitive SVM
AdaC	AdaCost
SBO	SMOTEBoost
RUS	RUSBoost
SB	SMOTEBagging
UB	UnderBagging
NB	klasyfikator Naiwnego Bayesa
J48	drzewo decyzyjne C 4.5
JRip	algorytm RIPPER
LR	regresja logistyczna
MLP	sieć neuronowa
AE-MV	metod eliminacji atrybutów (brakujące wartości atrybutów)
EM-MV	metoda maksymalizacji wartości oczekiwanej (brakujące wartości atrybutów)
BPCA-MV	Bayesowska Analiza Składowych Głównych (brakujące wartości atrybutów)
KNN-MV	metoda K najbliższych sąsiadów (brakujące wartości atrybutów)
WKNN-MV	ważona metoda K najbliższych sąsiadów (brakujące wartości atrybutów)
KMeans-MV	metoda K-średnich (brakujące wartości atrybutów)

Symbol/skrót	Opis
FKMeans-MV	rozmyta metoda K-średnich (brakujące wartości atrybutów)
LLSI-MV	metoda lokalnych najmniejszych kwadratów (brakujące wartości atrybutów)
SVR-MV	metoda regresji wektorów wspierających (brakujące wartości atrybutów)
SOA	Service Oriented Architecture
SODMA	Service Oriented Data Mining Architecture
DMS	Data Mining Service
DMSM	Data Mining Service Manager
DMSR	Data Mining Service Repository

Wytłuszczone symbole odnoszą się do wektorów. W nawiasach podane są metody i algorytmy, których oznaczenie dotyczy.

Spis rysunków

Spis tabel