dr inż. Paweł Ksieniewicz

Projektowanie algorytmów rozpoznawania wzorców dla zadania klasyfikacji trudnych danych

WNIOSEK O PRZEPROWADZENIE POSTĘPOWANIA W SPRAWIE NADANIA STOPNIA DOKTORA HABILITOWANEGO WRAZ Z ZAŁĄCZNIKAMI

Politechnika Wrocławska
Wybrzeże Wyspiańskiego 27
50-370 Wrocław
Dyscyplina naukowa: **Informatyka Techniczna i Telekomunikacja**
za pośrednictwem:
Rady Doskonałości Naukowej
pl. Defilad 1
00-901 Warszawa
(Pałac Kultury i Nauki, p. XXIV, pok. 2401)

Paweł Ksieniewicz
Politechnika Wrocławska
Wydział Informatyki i Telekomunikacji
Katedra Systemów i Sieci Komputerowych
Wybrzeże Wyspiańskiego 27
50-370 Wrocław

Wniosek

z dnia 27.08.2022

o przeprowadzenie postępowania w sprawie nadania stopnia doktora habilitowanego w dziedzinie **Nauk inżynieryjno-technicznych** w dyscyplinie[1] **Informatyka techniczna i telekomunikacja**.

> Określenie osiągnięcia naukowego będącego podstawą ubiegania się o nadanie stopnia doktora habilitowanego: cykl publikacji naukowych zatytułowany „**Projektowanie algorytmów rozpoznawania wzorców dla zadania klasyfikacji trudnych danych**".

Wnioskuję – na podstawie art. 221 ust. 10 ustawy z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce (Dz. U. z 2021 r. poz. 478 zm.) – aby komisja habilitacyjna podejmowała uchwałę w sprawie nadania stopnia doktora habilitowanego w głosowaniu **jawnym**[*2].

Zostałem poinformowany, że:
Administratorem w odniesieniu do danych osobowych pozyskanych w ramach postępowania w sprawie nadania stopnia doktora habilitowanego jest Przewodniczący Rady Doskonałości Naukowej z siedzibą w Warszawie (pl. Defilad 1, XXIV piętro, 00-901 Warszawa).
Kontakt za pośrednictwem e-mail: `kancelaria@rdn.gov.pl`*, tel. 22 656 60 98 lub w siedzibie organu.*
Dane osobowe będą przetwarzane w oparciu o przesłankę wskazaną w art. 6 ust. 1 lit. c) Rozporządzenia UE 2016/679 z dnia z dnia 27 kwietnia 2016 r. w związku z art. 220 - 221 orazart. 232 – 240 ustawy z dnia 20 lipca 2018 roku - Prawo o szkolnictwie wyższym i nauce, w celu przeprowadzenie postępowania o nadanie stopnia doktora habilitowanego oraz realizacji praw i obowiązków oraz środków odwoławczych przewidzianych w tym postępowaniu.
Szczegółowa informacja na temat przetwarzania danych osobowych w postępowaniu dostępna jest na stronie `www.rdn.gov.pl/klauzula-informacyjna-rodo.html`

.....
(podpis wnioskodawcy)

[1] Klasyfikacja dziedzin i dyscyplin wg. rozporządzenia Ministra Nauki i Szkolnictwa Wyższego z dnia 20 września 2018 r. w sprawie dziedzin nauki i dyscyplin naukowych oraz dyscyplin w zakresie sztuki (Dz. U. z 2018 r. poz. 1818).
[2*] Niepotrzebne skreślić.

Załączniki:

1. Dane wnioskodawcy
2. Kopia dokumentu potwierdzającego posiadanie stopnia doktora
3. Autoreferat wnioskodawcy
4. Wykaz osiągnięć naukowych
5. Deklaracje współautorów dotyczące wkładu pracy
6. Publikacje wchodzące w skład osiągnięcia naukowego „Projektowanie algorytmów rozpoznawania wzorców dla zadania klasyfikacji danych trudnych”

ZALACZNIK 1

Dane wnioskodawcy

1. Imię i Nazwisko: *Paweł Ksieniewicz*
2. Miejsce pracy: *Katedra Systemów i Sieci Komputerowych, Wydział Informatyki i Telekomunikacji, Politechnika Wrocławska*
3. Adres korespondencyjny: *ul. Xxxxxxxxx xxb/XXa, xx-xxx Wrocław*
4. Nr telefonu: *XXX XXX XXX*
5. Adres e-mail: *XXXXX.XXXXXXXXXXX@pwr.edu.pl*
6. Numer PESEL: *XXXXXXXXXXX*
7. Numer i seria dokumentu tożsamości w przypadku braku nadania numeru PESEL:

   .....

.....
(podpis wnioskodawcy)

ZALACZNIK 2

Kopia dokumentu potwierdzającego posiadanie stopnia doktora

Skan dyplomu poświadczającego uzyskanie stopnia naukowego doktora w dziedzinie nauk technicznych w dyscyplinie naukowej informatyka nadany uchwałą Rady Wydziału Elektroniki Politechniki Wrocławskiej z dnia 21 czerwca 2017 r.

DYPLOM

WYDANY W RZECZYPOSPOLITEJ POLSKIEJ

Politechnika Wrocławska

WYDZIAŁ ELEKTRONIKI
(nazwa jednostki organizacyjnej szkoły wyższej)

PAWEŁ KSIENIEWICZ
(imię lub imiona i nazwisko)

urodzony dnia 25 KWIETNIA 1989 roku w ŚWINOUJŚCIU

na podstawie przedstawionej rozprawy doktorskiej

„MULTIDIMENSIONAL DATA REPRESENTATION AND ANALYSIS"

(tytuł rozprawy doktorskiej)

oraz po złożeniu wymaganych egzaminów, uzyskał stopień naukowy

DOKTORA

w dziedzinie nauk TECHNICZNYCH

w dyscyplinie naukowej INFORMATYKA

nadany uchwałą Rady WYDZIAŁU ELEKTRONIKI

POLITECHNIKI WROCŁAWSKIEJ

z dnia 21 CZERWCA 2017 R.

Promotor w przewodzie doktorskim

PROF. DR HAB. INŻ. MICHAŁ WOŹNIAK

Kopromotor w przewodzie doktorskim

Recenzenci w przewodzie doktorskim PROF. DR HAB. INŻ. KATARZYNA STĄPOR

PROF. DR HAB. INŻ. KHALID SAEED

WROCŁAW 22 CZERWCA 2017 R.
(miejscowość i data wydania dyplomu)

(podpis Promotora)

Dziekan
Wydziału Elektroniki
Prof. dr hab. inż. Czesław Smutnicki

(pieczęć imienna i podpis Przewodniczącego rady jednostki organizacyjnej)

pieczęć urzędowa

REKTOR
Prof. Cezary Madryas

(pieczęć imienna i podpis Rektora)

nr 6224

PRK VIII
Kwalifikacja pełna na poziomie
Ramy Polskiej Ramy Kwalifikacji

ZALACZNIK 3

Autoreferat wnioskodawcy

1 *Imię i nazwisko*

Paweł Ksieniewicz

2 *Posiadane dyplomy, stopnie naukowe lub artystyczne – z podaniem podmiotu nadającego stopień, roku ich uzyskania oraz tytułu rozprawy doktorskiej*

2017 **Stopień doktora nauk technicznych w dyscyplinie Informatyka.**
Nadany uchwałą Rady Wydziału Elektroniki Politechniki Wrocławskiej.

Tytuł rozprawy:	*Multidimensional data representation and analysis*
Tytuł w języku polskim:	*Reprezentacja i analiza danych wielowymiarowych*
Promotor:	*prof. dr hab. inż. Michał Woźniak*

2017 **Tytuł zawodowy magistra inżyniera w dyscyplinie Informatyka.**
Wydział Elektroniki Politechniki Wrocławskiej.

Tytuł pracy magisterskiej:	*System wykrywania sztormów na Morzu Bałtyckim z wykorzystaniem metorogramów*
Promotorka:	*dr inż. Iwona Poźniak-Koszałka*

3 *Informacje o dotychczasowym zatrudnieniu w jednostkach naukowych lub artystycznych*

10.2017 – obecnie **Adiunkt**
Katedra Systemów i Sieci Komputerowych
Wydział Informatyki i Telekomunikacji
Politechnika Wrocławska
(do roku 2021 w Katedrze Systemów i Sieci Komputerowych Wydziału Elektroniki PWr)

5.2019 – 11.2021 **Specjalista ds. sztucznej inteligencji**
Uniwersytet Technologiczno-Przyrodniczy im. Jana i Jędrzeja Śniadeckich w Bydgoszczy

2.2015 – 10.2017 **Asystent**
Katedra Systemów i Sieci Komputerowych
Wydział Elektroniki
Politechnika Wrocławska

4 Omówienie osiągnięć, o których mowa w art. 219 ust. 1 pkt. 2 ustawy z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce (Dz. U. z 2021 r. poz. 478 z późn. zm.)

4.1 Tytuł osiągnięcia naukowego

W ramach niniejszego wniosku habilitacyjnego prezentowane jest osiągnięcie w formie cyklu powiązanych ze sobą tematycznie publikacji pod zbiorczym tytułem:

Projektowanie algorytmów rozpoznawania wzorców dla zadania klasyfikacji trudnych danych

4.2 Wykaz publikacji wchodzących w skład cyklu

Zamieszczony poniżej ciąg publikacji ułożony jest w odwrotnej kolejności chronologicznej, stanowiąc listę jedenastu artykułów opublikowanych w latach 2017–2022. Wszystkie podane przy nich wartości bibliometryczne oddają **stan na dzień 27. sierpnia 2022 r.** zgodnie z bazami publikacji naukowych:

WoS *Web of Science*

`https://www.webofscience.com/wos/author/record/1886494`

Sco *Scopus*

`https://www.scopus.com/authid/detail.uri?authorId=56206176100`

GSc *Google Scholar*

`https://scholar.google.com/citations?user=YSM30D8AAAAJ`

- Wpisy bibliograficzne dla artykułów w czasopismach zostały uzupełnione o informację o wartości czynnika *Impact Factor*[1] wydawnictwa z roku publikacji.
- Dla artykułów konferencyjnych podano informację o poziomie konferencji według rankingu CORE[2] w roku publikacji.
- Każdy wpis zawiera także informację o liczbie punktów *Ministerstwa Edukacji i Nauki* (MEiN) w roku publikacji.
- Do każdej pozycji ze spisu publikacji wchodzących w skład cyklu została dodana również informacja o moim wkładzie autorskim, zgodnie z wytycznymi wzorca CRediT (*Contributor Roles Taxonomy*)[3]. Określenia te pokrywają się z deklaracjami współautorów zawartymi w Załączniku 5 do wniosku.

[1] Journal Citation Reports — `https://clarivate.com/webofsciencegroup/solutions/journal-citation-reports`

[2] CORE Rankings Portal — `https://www.core.edu.au/conference-portal`

[3] CRediT author statement – `https://www.elsevier.com/authors/policies-and-guidelines/credit-author-statement`

[C1] Joanna Komorniczak, Paweł Zyblewski i Paweł Ksieniewicz. "Statistical Drift Detection Ensemble for batch processing of data streams". W: *Knowledge-Based Systems* 252 (2022), s. 109380. DOI: 10.1016/j.knosys.2022.109380

CRediT: **Conceptualization** **Software** **Validation** **Investigation** **Writing - Original Draft** **Writing - Review & Editing** **Visualization** **Supervision**

	WoS	Sco	GSc
l. cytowań	—	—	—

Szacowany udział	70%
Impact Factor	8.139
l. punktów MEiN	200

[C2] Pawel Ksieniewicz. "Processing data stream with chunk-similarity model selection". W: *Applied Intelligence* (lip. 2022). DOI: 10.1007/s10489-022-03826-4

CRediT: **Conceptualization** **Methodology** **Software** **Validation** **Formal Analysis** **Investigation** **Resources** **Data Curation** **Writing - Original Draft** **Writing - Review & Editing** **Visualization**

	WoS	Sco	GSc
l. cytowań	—	—	—

Szacowany udział	100%
Impact Factor	5.086
l. punktów MEiN	70

[C3] Joanna Komorniczak, Pawel Zyblewski i Pawel Ksieniewicz. "Prior Probability Estimation in Dynamically Imbalanced Data Streams". W: *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, lip. 2021. DOI: 10 . 1109 / ijcnn52387 . 2021.9533795

CRediT: **Conceptualization** **Validation** **Formal Analysis** **Investigation** **Resources** **Writing - Original Draft** **Writing - Review & Editing** **Visualization** **Supervision**

	WoS	Sco	GSc
l. cytowań	2	1	4

Szacowany udział	70%
Core	B
l. punktów MEiN	140

[C4] Paweł Ksieniewicz. "The prior probability in the batch classification of imbalanced data streams". W: *Neurocomputing* 452 (wrz. 2021), s. 309–316. DOI: 10.1016/j.neucom.2019.11.126

CRediT: **Conceptualization** **Methodology** **Software** **Validation** **Formal Analysis** **Investigation** **Resources** **Data Curation** **Writing - Original Draft** **Writing - Review & Editing** **Visualization**

	WoS	Sco	GSc
l. cytowań	2	2	4

Szacowany udział	100%
Impact Factor	5.719
l. punktów MEiN	140

[C5] Paweł Ksieniewicz, Paweł Zyblewski, Michał Choraś, Rafał Kozik, Agata Giełczyk, Michał Woźniak, *"Fake News Detection from Data Streams"*. W: *2020 International Joint Conference on Neural Networks (IJCNN)*. 2020, s. 1-8. DOI: 10.1109/IJCNN48605.2020.9207498

CRediT: **Conceptualization** **Methodology** **Software** **Validation** **Investigation** **Writing - Original Draft** **Writing - Review & Editing** **Visualization**

	WoS	Sco	GSc
l. cytowań	6	7	13

Szacowany udział	50%
Core	A
l. punktów MEiN	140

[C6] Pawel Ksieniewicz. "Combining Random Subspace Approach with smote Oversampling for Imbalanced Data Classification". W: *Hybrid Artificial Intelligent Systems*. Red. Hilde Pérez García i in. Cham: Springer International Publishing, 2019, s. 660–673. ISBN: 978-3-030-29859-3. DOI: 10.1007/978-3-030-29859-3_56

CRediT: **Conceptualization** **Methodology** **Software** **Validation** **Formal Analysis** **Investigation** **Resources** **Data Curation** **Writing - Original Draft** **Writing - Review & Editing** **Visualization**

	WoS	Sco	GSc
l. cytowań	4	5	5

Szacowany udział 100%
Core C
l. punktów MEiN 20

[C7] Paweł Ksieniewicz, Michał Woźniak, Bogusław Cyganek, Andrzej Kasprzak i Krzysztof Walkowiak. *"Data stream classification using active learned neural networks"*. W: *Neurocomputing 353 (2019)*, s. 74-82. DOI: 10.1016/j.neucom.2018.05.130

CRediT: **Methodology** **Software** **Validation** **Investigation** **Writing - Original Draft** **Writing - Review & Editing** **Visualization**

	WoS	Sco	GSc
l. cytowań	11	18	28

Szacowany udział 50%
Impact Factor 4.438
l. punktów MEiN 140

[C8] Pawel Ksieniewicz. "Undersampled Majority Class Ensemble for highly imbalanced binary classification". W: *Proceedings of the Second International Workshop on Learning with Imbalanced Domains: Theory and Applications*. Red. Luís Torgo i in. T. 94. Proceedings of Machine Learning Research. PMLR, paź. 2018, s. 82–94. URL: https://proceedings.mlr.press/v94/ksieniewicz18a.html

CRediT: **Conceptualization** **Methodology** **Software** **Validation** **Formal Analysis** **Investigation** **Resources** **Data Curation** **Writing - Original Draft** **Writing - Review & Editing** **Visualization**

	WoS	Sco	GSc
l. cytowań	—	—	10

Szacowany udział 100%
Core A
l. punktów MEiN 140

[C9] Paweł Ksieniewicz i Michał Woźniak. "Imbalanced Data Classification Based on Feature Selection Techniques". W: *Intelligent Data Engineering and Automated Learning – IDEAL 2018*. Red. Hujun Yin i in. Cham: Springer International Publishing, 2018, s. 296–303. DOI: 10.1007/978-3-030-03496-2_33

CRediT: **Methodology** **Software** **Validation** **Investigation** **Writing - Original Draft** **Writing - Review & Editing** **Visualization**

	WoS	Sco	GSc
l. cytowań	6	11	13

Szacowany udział 80%
Core B
l. punktów MEiN 15

[C10] Paweł Ksieniewicz, Bartosz Krawczyk i Michał Woźniak. "Ensemble of Extreme Learning Machines with trained classifier combination and statistical features for hyperspectral data". W: *Neurocomputing* 271 (2018), s. 28–37. DOI: 10 . 1016 / j . neucom.2016.04.076

CRediT: **Conceptualization** **Methodology** **Software** **Validation** **Writing - Original Draft** **Writing - Review & Editing** **Visualization**

	WoS	Sco	GSc
l. cytowań	15	15	18

Szacowany udział 60%
Impact Factor 4.072
l. punktów MEiN 30

[C11] Paweł Ksieniewicz i Michał Woźniak. "Dealing with the task of imbalanced, multidimensional data classification using ensembles of exposers". W: *Proceedings of the First International Workshop on Learning with Imbalanced Domains: Theory and Applications*. Red. Paula Branco Luís Torgo i Nuno Moniz. T. 74. Proceedings of Machine Learning Research. PMLR, 22 Sep 2017, s. 164–175. URL: https://proceedings.mlr.press/v74/ksieniewicz17a.html

CRediT: **Conceptualization** **Methodology** **Software** **Validation** **Formal Analysis** **Investigation** **Resources** **Data Curation** **Writing - Original Draft** **Writing - Review & Editing** **Visualization**

	WoS	Sco	GSc
l. cytowań	—	—	9

Szacowany udział 80%
Core A
l. punktów MEiN 140

4.3 Informacje naukometryczne

Podane poniżej wartości oddają **stan na dzień 27. sierpnia 2022 r.** zgodnie z bazami publikacji naukowych *Web of Science* (WoS), *Scopus* (Sco) oraz *Google Scholar* (GSc).

- Sumaryczny IF dla osiągnięcia 27,454
- Sumaryczne MEiN 1 175

	WoS	Sco	GSc
○ Sumaryczna liczba cytowań dla osiągnięcia	46	59	104
○ Indeks Hirscha autora	8	10	13
l. cytowań.	157	276	377
bez autocytowań	134	240	—
l. dokumentów	36	42	51

5 *Omówienie celu naukowego wyżej wymienionych prac, osiągniętych wyników oraz ich ewentualnego wykorzystania*

Duży odsetek publikacji w tematyce systemów nadzorowanego uczenia się maszyn rozpoczynany jest parafrazą zdania *"współczesny świat wypełniony jest danymi"*. Przyczynia się do tego wiele postępujących czynników, rozpoczynając od tak pozytywnych zjawisk jak masowa digitalizacja treści [Z1], automatyzacja procesów produkcyjnych [Z2] czy rosnąca rola systemów rekomendacyjnych [Z3].

Należy jednak zwrócić również uwagę na bardziej złożone społecznie zjawiska takie jak popularyzacja wykorzystania pojazdów autonomicznych, w tym w usługach kurierskich i transportowych redukujących czynnik ludzki [Z4], spadająca jakość produktów elektronicznych powodująca zwiększoną potrzebę automatycznych systemów diagnostyki sprzętu [Z5], modelowanie zachowań ludzkich i – szczególnie niepokojące – profilowanie konsumentów w świecie, w którym dominująca większość naszych zachowań pozostawia swój ślad cyfrowy [Z6]. Proces ten wzmacniany jest także przez trwającą nieprzerwanie od końca 2019 roku pandemię wirusa *SARS-CoV-2*, który w dniu finalnej redakcji tego dokumentu dotknął bezpośrednio już ponad pół miliarda mieszkańców Ziemi, stanowiąc dodatkowy, niezaprzeczalny argument za automatyzacją czynności i procedur, które dotychczas wykonywane były przez odpowiednio wyszkolonych do tego celu ludzi.

Już w latach siedemdziesiątych zauważone zostało, zarówno w krytycznym dla całego środowiska maszyn uczących się raporcie Lighthilla [Z7], jak i w merytorycznych ocenach ówczesnego stanu badań nad systemami sztucznej inteligencji [Z8], że indukcyjne modele predykcyjne najlepiej radzą sobie z problemami opisywanymi przez stosunkowo niewielką liczbę atrybutów. Przy silnie ograniczonej mocy obliczeniowej komputerów tamtej epoki [Z9], większość publikacji opierała zawarte w nich obserwacje na temat efektywności poszczególnych metod rozpoznawania na tzw. *toysetach* (zbiorach ilustracyjnych), ograniczonych do zaledwie kilku wymiarów i opisanych przez nie więcej niż kilkaset instancji. Ograniczenia te istotnie utrudniały generalizację wniosków na problemy rzeczywiste, opisywane przez potencjalnie nieskończone zbiory danych o wysokiej liczbie atrybutów.

Metody uczenia nadzorowanego, zgodnie z powszechnie przyjętą taksonomią, skupiają się na zadaniach regresji i klasyfikacji [Z10]. W pierwszym z tych przypadków rolą systemu rozpoznawania jest zyskanie zdolności do wyznaczania, najczęściej ciągłej, wartości zmiennej objaśnianej na podstawie zbioru odpowiednio anotowanych obserwacji. W zadaniu klasyfikacji – na którym skupia się większość prowadzonych przeze mnie badań – obiekty przypisywane są do jednej z predefiniowanych kategorii nazywanych klasami problemu. W wypadku rzeczywistych zbiorów danych dla zadania klasyfikacji stosunkowo rzadka jest sytuacja, w której dysponujemy równomierną reprezentacją każdej z tych kategorii, co przynosi dodatkową trudność w ich analizie [Z11]. Trudność ta jest na tyle silna, że pozwoliła na wyodrębnienie się istotnego nurtu badań nad klasyfikacją danych niezbalansowanych [Z12].

Modele klasyfikacji, często niezależnie od rodziny z której pocho-

[Z1] P. Sulima-Samujłło. *Kolekcje polskich i dalekowschodnich (Chiny, Japonia, Korea) bibliotek cyfrowych. Analiza porównawcza*. 2017

[Z2] W. Van der Aalst, M. Bichler i A. Heinzl. *Robotic process automation*. 2018

[Z3] S. Zhang i in. "Deep learning based recommender system: A survey and new perspectives". W: *ACM Computing Surveys (CSUR)* (2019)

[Z4] S. M. Shavarani i in. "Application of hierarchical facility location problem for optimization of a drone delivery system: a case study of Amazon prime air in the city of San Francisco". W: *The International Journal of Advanced Manufacturing Technology* (2018)

[Z5] D. Neupane i J. Seok. "Bearing fault detection and diagnosis using case western reserve university dataset with deep learning approaches: A review". W: *IEEE Access* (2020)

[Z6] S. Miller i in. "Machine learning, ethics and law". W: *Australasian Journal of Information Systems* (2019)

[Z7] J Lighthill. *Artificial intelligence: a general survey. Science Research Council*. 1973

[Z8] J. McCarthy, E. Feigenbaum i J. Lederberg. *Artificial Intelligence Project*. Spraw. tech. Progress Report, 1973

[Z9] R. Tadeusiewicz. *Krótka historia informatyki*. Wydawnictwo RM, 2019

[Z10] E. Alpaydin. *Machine learning*. MIT Press, 2021

[Z11] Y. Sun, A. Wong i M. Kamel. "Classification of imbalanced data: A review". W: *International journal of pattern recognition and artificial intelligence* (2009)

[Z12] B. Krawczyk. "Learning from imbalanced data: open challenges and future directions". W: *Progress in Artificial Intelligence* (2016)

dzą, wykazują silną tendencję do faworyzowania decyzji w kierunku klasy dominującej zbiór danych [Z13]. Przykładowo, reguły splitu drzew decyzyjnych, przypisując etykietę do wierzchołka budowanego grafu uwzględniają przede wszystkim proporcję pomiędzy etykietami zakreślonych jego parametrami obiektów [Z14]. Funkcje straty wykorzystywane w trenowaniu sieci neuronowych najczęściej tym samym kosztem obciążają błędne decyzje względem każdej z klas problemu [Z15]. Wreszcie, proste podejścia do rozpoznawania takie jak klasyfikatory minimalnoodległościowe, również wykazują tendencję do premiowania obiektów klas o większym prawdopodobieństwie *a priori*, co w każdym z wymienionych przypadków prowadzi do budowy modeli nadmiernie ograniczających obszar decyzyjny klasy mniejszościowej [Zyb19].

Najpowszechniej występujące współcześnie trudności przetwarzania danych powiązane są z czynnikiem ich masowości, dokładniej opisywanym przez model 4V Big Data [Z16]. Masowość danych wyrażać może się zarówno we wspomnianym już kontekście wielowymiarowości, jak też w niezwykle dużej liczności zbiorów. Pierwotną motywacją badań w tym zakresie były ograniczenia obliczeniowe i pamięciowe maszyn z drugiej połowy XX wieku, które nie pozwalały ani na jednoczesne składowanie pełnego zbioru danych, ani na przetworzenie wszystkich dostępnych wzorców w akceptowalnym czasie obliczeniowym [Z17]. Podstawowe rozwiązania w tym zakresie wprowadzają paradygmat tzw. uczenia inkrementalnego, dostępny zarówno dla prostych modeli takich jak *Naive Bayes* czy niektóre drzewa decyzyjne [Z18], jak i dla złożonych procedur optymalizacyjnych, wśród których największy nacisk należałoby położyć na sieci neuronowe [Z19].

Typowy przypadek uczenia inkrementalnego, podobnie jak bardziej ogólna koncepcja *i. i. d.* (ang. *independent and identically distributed*) stanowiąca kluczowe zagadnienie walidacji modeli rozpoznawania, opiera się na założeniu o stacjonarności koncepcji, tj. na niezmienności prawdopodobieństwa *a posteriori* problemu. Dzięki takiemu założeniu, algorytm uczenia pozwalający na aktualizację wiedzy otrzymuje w każdym kroku wsad z dostępnych danych uczących – w zależności od zastosowanego podejścia rozłączny lub losowy – i dokonuje korekty modelu na ich podstawie. Korekta ta uwzględnia fakt, że systemowi rozpoznawania dostarczone zostały już wcześniej pewne przykłady opisujące problem i z czasem rola nowych obiektów jest coraz mniejsza, pozwalając na konwergencję modelu [Z20].

Dziedzina przetwarzania strumieni danych rozwija paradygmat uczenia inkrementalnego o uwzględnienie czynnika czasu przez uznanie zbioru obiektów za uporządkowaną sekwencję. Daje to zarówno podstawę do właściwego spojrzenia na prędkość przetwarzania danych – ponieważ obserwacje opisywane są i w czasie i przestrzeni – jak i rozszerzenie pojęcia różnorodności poza heterogeniczność źródeł, przez zniesienie założenia o stacjonarności koncepcji. Procedury uczenia w takim środowisku muszą ulegać zmianie i nie dążyć już do analizy stabilnego zjawiska, ale do konsekwentnej modyfikacji założeń na jego temat, znajdujących odzwierciedlenie w modelu zdolnym do śledzenia dynamiki jego zmian [Z21].

[Z13] V. Ganganwar. "An overview of classification algorithms for imbalanced datasets". W: *International Journal of Emerging Technology and Advanced Engineering* (2012)

[Z14] S. Kotsiantis. "Decision trees: a recent overview". W: *Artificial Intelligence Review* (2013)

[Z15] P. Alaba i in. "Towards a more efficient and cost-sensitive extreme learning machine: A state-of-the-art review of recent trend". W: *Neurocomputing* (2019)

[Zyb19] P. Zyblewski, P. Ksieniewicz i M. Woźniak. "Classifier Selection for Highly Imbalanced Data Streams with Minority Driven Ensemble". W: *Artificial Intelligence and Soft Computing*. Springer International Publishing, 2019, s. 626–635

[Z16] S. Sagiroglu i D. Sinanc. "Big data: A review". W: *2013 international conference on collaboration technologies and systems (CTS)*. IEEE. 2013

[Z17] T. Chan, G. Golub i R. LeVeque. "Updating formulae and a pairwise algorithm for computing sample variances". W: *COMPSTAT 1982 5th Symposium held at Toulouse 1982*. Springer. 1982

[Z18] V. da Costa, A. de Leon Ferreira, S. Junior i in. "Strict very fast decision tree: a memory conservative algorithm for data stream mining". W: *Pattern Recognition Letters* 116 (2018), s. 22–28

[Z19] Ian Goodfellow, Yoshua Bengio i Aaron Courville. *Deep learning*. MIT press, 2016

[Z20] Y. Wu i in. "Large scale incremental learning". W: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, s. 374–382

[Z21] B. Krawczyk i in. "Ensemble learning for data stream analysis: A survey". W: *Information Fusion* (2017)

Zmiany tego rodzaju zwykło się w literaturze nazywać dryfami lub dryftami koncepcji (ang. *concept drift*) [Z22]. W uproszczonej taksonomii możemy wyróżnić dryfy wirtualne – zmieniające co prawda rozkład klas w przestrzeni, ale nie mające istotnego wpływu na zmiany we właściwej granicy decyzyjnej oraz dryfy rzeczywiste – stanowiące podstawowe źródło czasowej degeneracji modeli i najczęściej spotykane w strumieniach rzeczywistych. Degeneracja modelu może być również tymczasowym spadkiem jego zdolności generalizacyjnej, w wypadku strumieni o koncepcjach nawracających (ang. *recurrent drift*), typowych dla danych opisujących zjawiska okresowe o uporządkowanej (cykl dobowy, cykl roczny) lub nieuporządkowanej (pogoda, rynek giełdowy) naturze.

[Z22] J. Gama i in. "A survey on concept drift adaptation". W: *ACM computing surveys (CSUR)* (2014)

Podział cyklu publikacji na obszary tematyczne

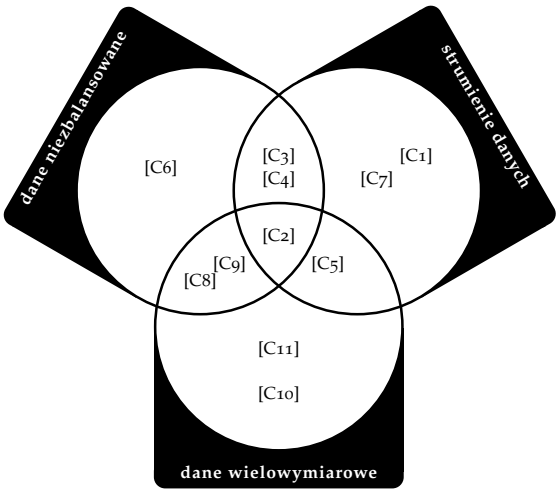
Od roku złożenia rozprawy doktorskiej, która podejmowała tematykę metod reprezentacji i klasyfikacji danych wielowymiarowych, w swoich pracach badawczych stale rozwijam metody pozwalające na efektywne przetwarzanie danych obciążonych wymienionymi wyżej trudnościami. Biorąc je pod uwagę, do typowego dla wstępów do prac z dziedziny stwierdzenia o tym, że *świat wypełniony jest danymi*, warto byłoby dodać informację, iż dane te – tak samo jak świat, który opisują – ulegają ciągłym zmianom, a efektywne systemy rozpoznawania powinny być zdolne do ich rejestrowania i nadążania za nimi.

W kolejnych sekcjach opiszę zaproponowane przeze mnie metody oraz towarzyszące im publikacje, wpisujące się w opisaną tematykę i będące podstawą osiągnięcia naukowego. Listę jedenastu artykułów naukowych stanowiących opisywany cykl wpisałem w trzy przenikające się kategorie:

I Algorytmy przetwarzania danych wielowymiarowych.

II Algorytmy przetwarzania danych niezbalansowanych.

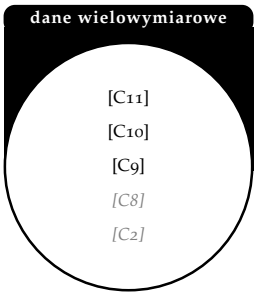
III Algorytmy przetwarzania strumieni danych.



Rysunek 3.1: Przypisanie prac z cyklu publikacji do zakresu tematycznego klasyfikacji danych trudnych. Podane w nawiastach kwadratowych indentyfikatory odwołują się do spisu zawartego w P4.

Przypisanie poszczególnych prac do kategorii zostało zilustrowane w Rysunku 3.1 zawierającym nachodzące na siebie obszary badawcze wraz ze stosownymi odniesieniami literaturowymi.

I – Algorytmy przetwarzania danych wielowymiarowych



Po prawej stronie diagram zaznaczający opisywane w ramach wątku przetwarzania danych wielowymiarowych, wliczający prace opisane w innych wątkach dominujących.

Najsilniej rozwijającą się w ostatnim dziesięcioleciu odpowiedzią środowiska naukowego na stale rosnącą wymiarowość danych jest bez wątpienia nurt uczenia reprezentacji (ang. *representation learning*). Dedykowany jest on głównie wielowymiarowym danym sygnałowym, w których najczęściej spotykamy bardzo silne, ale jednocześnie zmienne, relacje pomiędzy atrybutami, stanowiącymi najczęściej wartości poszczególnych elementów obrazu.

Niemniej jednak, nadal interesującą alternatywą są tutaj metody klasyczne, uwzględniające element manualnej inżynierii cech. Pozwalają one na wstępną redukcję atrybutów, które łatwiej jest opisać przez podprzestrzenną dywersyfikację nowej przestrzeni, konstruując tak już dwupoziomowy system rozpoznawania zdolny do budowy złożonych granic decyzyjnych nawet przy użyciu prostych klasyfikatorów bazowych.

Szczególnym przypadkiem obrazów cyfrowych są *obrazy nadwidmowe*[4] (ang. *hyperspectral images*). Stanowią one efekt próbkowania promieniowania elektromagnetycznego zarówno w dwóch wymiarach przestrzennych – jak standardowe obrazy – jak i w wymiarze spektralnym, dzięki zastosowaniu pryzmatu rozszczepiającego promieniowanie elektromagnetyczne na wąskie, rozłączne pasma opisujące zakres częstotliwości przekraczający zdolność widzenia ludzkiego[5]. W sytuacji, w której liczba takich pasm przekracza setkę, zbiór atrybutów dla każdego piksela przestaje być pulą luźno powiązanych ze sobą czynników (jak w reprezentacji RGB czy obrazach wielowidmowych) i staje się już typową dla obrazu nadwidmowego sygnaturą spektralną.

Stanowi to niezwykle interesujący przypadek danych rzeczywistych, w których możemy mówić o ciągłości informacji w każdym z wymiarów obrazu. Sygnatura spektralna, którą najczęściej wykorzystujemy w badaniach jako bazową reprezentację wzorca, jest jednowymiarowym sygnałem cyfrowym zbliżonym w swoim przebiegu do wyraźnie zaszumionego wykresu funkcji wielomianowej. Szum ten najczęściej ma charakter globalny w skali obrazu i typowy dla wykorzystanego sensora.

Dane nadwidmowe reprezentowane są przez tzw. *kostki hiperspektralne* (ang. *hyperspectral cubes*) będące trójwymiarowymi tensorami, które dla zadania klasyfikacji uzupełnione są o dwuwymiarową, dyskretną mapę *ground truth*, tj. etykiet każdego piksela. Prezentują one najczęściej zdjęcia lotnicze pól uprawnych i miast, budując wieloklasowe problemy niezbalansowane, choć zakres ich zastosowań wykracza też poza tę specyfikę i odnajduje się – przykładowo – w obrazowaniu palimpsestów i kontroli jakości produkcji.

[4] W j. polskim brak jest powszechnie uznanej taksonomii dla danych o ciągłym wymiarze spektralnym, w związku z czym posługuję się tutaj tłumaczeniem wykorzystywanym w raportach sprawozdawczych, zaproponowanym w 2014 roku.
[5] Na przykładzie sensora AVIRIS.

W [C10] zaproponowałem autorską metodę budowy systemu rozpoznawania dla obrazów nadwidmowych, opartą o ręczną inżynierię atrybutów i zespołową integrację szybkich sieci neuronowych *Extreme Learning Machines* (ELM).

[C10]

Paweł Ksieniewicz, Bartosz Krawczyk i Michał Woźniak. "Ensemble of Extreme Learning Machines with trained classifier combination and statistical features for hyperspectral data". W: *Neurocomputing* 271 (2018), s. 28–37. DOI: 10.1016/j.neucom.2016.04.076

Silna zależność pomiędzy cechami zawartymi w sygnaturze spektralnej powiązana jest z równie silną nadmiarowością zapisanej w niej informacji. Strategią przeciwdziałającą redundancji jest zerwanie ciągłości widmowej danych, którą możemy uzyskać przez selekcję atrybutów lub ich ekstrakcję. W selekcji jest to często operacja silnie stratna i wzmacniająca rolę szumu w pojedynczej próbce, a w ekstrakcji i uczeniu reprezentacji – gubiąca interpretowalność atrybutów. Alternatywą jest tutaj, wzorowana na właściwościach fotoreceptorów, ręczna inżynieria atrybutów, gdzie każda nowa cecha stanowi liniową kombinację elementów sygnatury bazowej, posiada swoją semantyczną interpretację i wykazuje obniżoną zależność od pozostałych.

Zaproponowałem tu czternaście metryk statystycznych rozpinających się od prostych miar, takich jak minimum, średnia czy mediana sygnatury, przez metryki zróżnicowania sygnału reflektancji, interpretowanego jako zmienna losowa po pseudo kanały barwne uzyskane z projekcji całego obrazu do modelu HSV.

Należy mieć tu na uwadze, że inżynieria atrybutów oparta na miarach statystycznych jest nadal silnie czuła na błędy pomiarów, które są szczególnie widoczne w rzeczywistych obrazach nadwidmowych. Do przeciwdziałania ich negatywnym skutkom wykorzystałem, zaproponowany przeze mnie w jednej z wcześniejszych prac, *Entropodynamiczny Filtr Percentylowy* [Ksi18e], który pozwala na detekcję impulsywnych zmian sygnału i tym samym oznaczenie pasm sygnatury obciążonych wysokim błędem – zakłócającym wartości metryk statystycznych.

[Ksi18e] Paweł Ksieniewicz. "Entropodynamiczny filtr percentylowy". W: *Edukacja – Technika – Informatyka*. Wydawnictwo Uniwersytetu Rzeszowskiego, 2018

Kluczowym zagadnieniem pracy było odpowiednie wykorzystanie potencjału algorytmu ELM. Jego model neuronowy jest silnie oparty na losowości, pozwalając na znacznie krótszy proces uczenia niż MLP, jednak kosztem znacznie wyższej wariancji modelu. Stabilizacja takiego rozwiązania często realizowana jest przez wykorzystanie uczenia zespołowego (ang. *ensemble learning*), czego skuteczność została już potwierdzona w wielu publikacjach [Z23]. Krytyczny dla zespołu aspekt dywersyfikacji jego puli najczęściej opiera się na losowości wag wejściowych, co przy szerokim i silnie zależnym wewnętrznie wektorze atrybutów zdecydowanie nie jest rozwiązaniem optymalnym. Literatura pokazuje, że dla danych tego typu znacznie lepszą strategią jest dywersyfikacja projekcyjna [Z24] lub podprzestrzenna [Sul21], która możliwa jest do efektywnego zastosowania – przykładowo – po zaproponowanej ręcznej inżynierii cech.

[Z23] Guang-Bin Huang, Qin-Yu Zhu i Chee-Kheong Siew. "Extreme learning machine: theory and applications". W: *Neurocomputing* 70.1-3 (2006), s. 489–501

[Z24] Juan José Rodriguez, Ludmila I Kuncheva i Carlos J Alonso. "Rotation forest: A new classifier ensemble method". W: *IEEE transactions on pattern analysis and machine intelligence* (2006)

[Sul21] Sułot D, P. Zyblewski i P. Ksieniewicz. "Analysis of Variance Application in the Construction of Classifier Ensemble Based on Optimal Feature Subset for the Task of Supporting Glaucoma Diagnosis". W: *Computational Science – ICCS 2021*. Springer International Publishing. DOI: 10.1007/978-3-030-77967-2_10

W wypadku dywersyfikacji podprzestrzennej (ang. *random subspace*) musimy pamiętać o zapewnieniu pełnego pokrycia oryginalnej przestrzeni atrybutów. Prawdopodobieństwo pokrycia możemy wyliczyć przez wzór:

$$P(coverage) = 1 - (1 - \frac{r}{d})^L, \tag{3.1}$$

gdzie r to wielkość podprzestrzeni, d – liczba atrybutów, a L to wielkość puli. Naturalnie, przy zachowaniu stałej wielkości podprzestrzeni r, wraz ze wzrostem liczby wymiarów problemu powinniśmy też proporcjonalnie zwiększać pulę modeli, pamiętając jednak o tym, że po

przekroczeniu pewnego limitu będzie ona wpływać negatywnie na różnorodność.

Przy wykorzystaniu 14. atrybutów statystycznych zamiast ponad 200. spektralnych[6] możliwe jest więc zbudowanie mniejszego zespołu klasyfikatorów dywersyfikowanego przez RSM pozwalającego na pełne pokrycie przestrzeni atrybutów.

[6] Najczęściej wykorzystywany w danych benchmarkowych sensor AVIRIS opisuje 224 pasma w widmie od 400 do 2 500 nanometrów.

Zespoły ELM domyślnie integruje się przez reguły głosowania większościowego, co daje duży potencjał do modyfikacji mających istotny wpływ na jakość docelowego modelu. W pracy proponuję więc komplementarnie ważoną integrację zespołu, gdzie za wyliczenie optymalnych wag – dla każdej klasy indywidualnie – odpowiada pojedynczy perceptron.

Uzyskane rezultaty badań pozwoliły na uprawdopodobnienie hipotezy stanowiącej o tym, że zespołowy model ELM dywersyfikowany przez RSM z wykorzystaniem atrybutów statystycznych, integrowany z wykorzystaniem wyuczalnej reguły decyzyjnej okazuje się statystycznie istotnie lepszy od podejść standardowych dla większości analizowanych zbiorów danych. Możemy dzięki nim zaobserwować też, że odpowiednia selekcja atrybutów, bez wykorzystywania metod wbudowanych czy resamplingu, może prowadzić do istotnego statystycznie ulepszenia modelu klasyfikacji nawet w wieloklasowych danych niezbalansowanych.

[C9]

Paweł Ksieniewicz i Michał Woźniak. "Imbalanced Data Classification Based on Feature Selection Techniques". W: *Intelligent Data Engineering and Automated Learning – IDEAL 2018*. Red. Hujun Yin i in. Cham: Springer International Publishing, 2018, s. 296–303. DOI: 10.1007/978-3-030-03496-2_33

Obserwacja ta stała się podstawą do badań przedstawionych w [C9]. Stanowią one analizę odchodzącą już od specyfiki obrazów nadwidmowych, gdzie ewaluacja przeprowadzona została na kolekcji 35 tabelarycznych zbiorów o skali niezbalansowania sięgającej 1:41.

Należy zaznaczyć, że punktem wyjściowym badań było wykorzystanie problemów o wstępnie zredukowanej wymiarowości – przyjmując przestrzenie od 8 do 19 atrybutów – i analiza potencjału metod optymalizacyjnych w selekcji podprzestrzeni do ulepszenia efektywności w klasyfikacji niezbalansowanej.

W artykule zaproponowałem autorską metodę hybrydową wykorzystującą techniki selekcji atrybutów do budowy puli klasyfikatorów na potrzeby zespołu. Celem uniknięcia przeuczenia zastosowałem techniki regularyzacyjne zapewniające maksymalizację różnorodności zespołu przy jednocześniej minimalizacji wielkości podprzestrzeni dla każdego klasyfikatora bazowego. Zostały one wprowadzone do procedury optymalizacyjnej przez konstrukcję następującej funkcji kryterialnej:

$$Q(\Pi) = BAC(\Pi) - \alpha * \frac{NF(\Pi)}{d} + \beta * \frac{AH(\Pi)}{d}, \tag{3.2}$$

gdzie $BAC(\Pi)$ wskazuje na zbalansowaną dokładność zespołu, którego pulę definiujemy jako Π, d wskazuje na liczbę bazowych atrybutów problemu, a dalsze składowe definiują czynniki regularyzacyjne:

NF – liczbę atrybutów dostępnych w puli, która podzielona przez d daje skalę pokrycia oryginalnej przestrzeni problemu przez zespół – sterowana przez hiperparametr α.

AH – średnią odległość Hamminga pomiędzy słowami zakodowanymi jako binarne maski podprzestrzeni osobników – sterowana przez hiperparametr β.

Szczególnie istotny wydaje się tutaj czynnik AH stanowiący rodzaj metryki różnorodności, który w ustawieniu odwrotnej proporcjonalności do modułu atrybutów powinien pozwalać na eliminację klasyfikatorów o wysokiej zależności statystycznej.

Procedura optymalizacyjna rozpoczyna się od wygenerowania początkowej populacji puli klasyfikatorów

$$Population = \{\Pi_1, \Pi_2, \ldots, \Pi_S\}, \tag{3.3}$$

gdzie parametr S, jako czynnik arbitralny, określa wielkość populacji. Jego zwiększanie wiąże się z wcześniejszą konwergencją optymalizacji, ale generuje też wykładniczy narzut obliczeniowy. Osobniki oceniane są przez kryterium określone w Równaniu 3.2. Dla uproszczenia obliczeń algorytm stosuje prostą strategię przeszukiwania, w każdym kroku pozostawiając w puli jedynie najbardziej rokujący zespół, integrowany przez jedną z trzech zaproponowanych strategii:

R – prosta akumulacja wsparć bez ważenia klasyfikatorów bazowych,

W – ważona akumulacja wsparć z wagami proporcjonalnymi do zbalansowanej dokładności osiąganej przez klasyfikatory bazowe,

N – ważona akumulacja wsparć z wagami stanowiącymi znormalizowaną zbalansowaną dokładność osiąganą przez klasyfikatory bazowe.

Otrzymane wyniki uprawdopodobniają hipotezę stanowiącą o tym, że selekcja atrybutów odgrywa krytyczną rolę w klasyfikacji danych niezbalansowanych. W większości przypadków zaproponowana przeze mnie metoda osiągała istotnie wyższe rezultaty niż modele wyuczone na pełnej dostępnej reprezentacji.

Warto zauważyć, że struktura tożsama logicznie z obrazami wielo- i nadwidmowymi może zostać uznana za reprezentację nie tylko pojedynczego zdarzenia, ale i całej koncepcji. Praca [C11] rozwija koncepcję takiej właśnie reprezentacji, nazywanej *ekspozerem*.

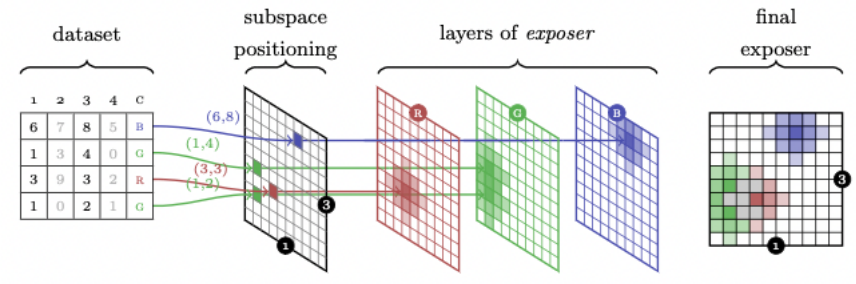
Podstawowym założeniem ekspozera jest to, że wymiary przestrzenne dyskretnej reprezentacji sygnałowej mogą stanowić odwzorowanie planarnej projekcji problemu, a wymiar spektralny – odpowiadać za niezależne próbkowanie go w klasach. Istotne zredukowanie jej rozdzielczości wraz z zaproponowaniem odpowiedniej reguły predykcyjnej pozwala na wykorzystanie jej jako efektywnego czasowo i jakościowo modelu rozpoznawania niepozostającego w silnej zależności z żadną wykorzystywaną powszechnie metodą indukcji. W wypadku ciągłego próbkowania wymiaru spektralnego stosowny model miałby potencjał do rozwiązywania zadania regresji, a przy podejściu dyskretnym – możliwy byłby do zastosowania w zadaniu klasyfikacji.

Procedura budowy ekspozera inspirowana jest procesem naświetlania kliszy fotograficznej. Dlatego też, parametrami kontrolującymi

[C11]

Paweł Ksieniewicz i Michał Woźniak. "Dealing with the task of imbalanced, multidimensional data classification using ensembles of exposers". W: *Proceedings of the First International Workshop on Learning with Imbalanced Domains: Theory and Applications*. Red. Paula Branco Luís Torgo i Nuno Moniz. T. 74. Proceedings of Machine Learning Research. PMLR, 22 Sep 2017, s. 164–175. URL: https://proceedings.mlr.press/v74/ksieniewicz17a.html

indukcje są (a) ziarno filmu – odpowiadające za gęstość próbkowania przestrzennego i (b) czynnik dyspersji światła – odpowiedzialny za płynne złamanie zasady rozłączności kubełków histogramu. W przeciwieństwie do fizycznego pierwowzoru, struktura taka *naświetlana* jest przez wiązkę wzorców zorganizowanych przestrzennie przez projekcję do dwuwymiarowej podprzestrzeni. Wynikiem takiej procedury jest nieskwantyzowany obraz cyfrowy, gdzie intensywność każdego punktu stanowi zakumulowaną gęstość rozkładu wzorców oddziałujących na jego sąsiedztwo.



Rysunek 3.2: Ekspozycja czterech obiektów opisujących trzy klasy w dwuwymiarowej podprzestrzeni czterowymiarowego problemu.

Wizualizacja procesu ekspozycji dla czterech obiektów opisujących trzy klasy w dwuwymiarowej podprzestrzeni czterowymiarowego problemu przedstawiona została na Rysunku 3.2. W pierwszym kroku odbywa się tutaj pozycjonowanie podprzestrzenne, w którym każdy z obiektów zawartych w zbiorze uzyskuje nowe współrzędne w skończonej, planarnej projekcji na siatce 10×10 (parametr ziarna). Ekspozer składa się z tylu warstw, ile klas zawiera się w zbiorze danych, a każda z nich czuła jest jedynie na obiekty przynależącej do niej klasy.

Przyjmijmy, że $\mathcal{DS}$ określa zbiór n instancji, z których każda (x_k) reprezentowana jest przez d-wymiarowy wektor atrybutów oraz etykietę i_k ze skończonego zbioru etykiet $\mathcal{M}$.

$$\begin{aligned} \mathcal{X} &\subseteq \Re^d \\ \mathcal{M} &= \{1, 2, \ldots, M\} \\ x_k &= [x_k^{(1)}, x_k^{(2)}, \ldots, x_k^{(d)}]^T, \qquad x_k \in \mathcal{X} \\ \mathcal{DS} &= \left\{(x_1, i_1), (x_2, i_2), \ldots, (x_n, i_n)\right\} \end{aligned} \tag{3.4}$$

Różnorodność możliwych do zbudowania ekspozerów wyraża się przez zbiór Λ, zawierający $\binom{d}{s}$ kombinacji λ_i, gdzie s stanowi wybraną wymiarowość ekspozera, w planarnym wypadku wynoszącą 2.

$$\begin{aligned} \Lambda &= \{\lambda_1, \lambda_2, \ldots, \lambda_L\}, \quad |\Lambda| = L = \binom{d}{s} \\ \lambda_i &= [l_1, l_2, \ldots, , l_s], \qquad l_j \in \{1, 2, \ldots, d\}, \quad l_1 \neq l_2 \neq \ldots \neq l_s, \end{aligned} \tag{3.5}$$

Przy takich założeniach, reprezentacją ekspozera $(\mathcal{E})$ jest s-wymiarowa kostka danych:

$$\begin{aligned} \mathcal{E}_m &\in G^s = \underbrace{G \times G \times \ldots \times G}_{s} \\ \mathcal{E} &= \{\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_M\}, \end{aligned} \tag{3.6}$$

gdzie każda komórka adresowana przez loc zawiera wektor wartości

$$\begin{aligned} \mathcal{E}^{(loc)} &= [v_1, v_2, \ldots, v_M]^T \\ loc &= [loc_1, loc_2, \ldots, loc_s]^T \end{aligned} \tag{3.7}$$

Pojedyncza wartość jest sumą wszystkich pozytywnych różnic zadanego promienia oddziaływania r z dystansem pomiędzy punktem centralnym komórki (loc) i każdym punktem loc_k dla którego $i_k = m$.

$$\begin{aligned} \mathcal{E}_m^{(loc)} &= \sum_{k=1}^{n} \Big[d(loc, loc_k) < r \ \wedge \ i_k = m \Big] \cdot \Big(r - d(loc, loc_k) \Big) \\ loc_k &= [x_k^{(\lambda_1)}, x_k^{(\lambda_2)}, \ldots, x_k^{(\lambda_s)}]^T \end{aligned} \tag{3.8}$$

Ekspozer $\mathcal{E}$, zbudowany na podprzestrzeni λ zbioru uczącego $\mathcal{LS}$ może zostać wykorzystany do inferencji dla obiektu testowego x_k ze zbioru testowego $\mathcal{TS}$ przez spróbkowanie go po projekcji do wspólnej przestrzeni

$$\Psi(x_k) = \underset{m \in \mathcal{M}}{argmax} \left(\mathcal{E}_m^{(loc_k)} \right). \tag{3.9}$$

Podobnie jak w przypadku zastosowania *Extreme Learning Machines* do klasyfikacji atrybutów statystycznych uzyskanych z sygnatur spektralnych, i tutaj do konstrukcji właściwego systemu rozpoznawania wykorzystywany jest nie pojedynczy, podprzestrzenny model, a zespół ekspozerów Π zbudowany na zbiorze kombinacji Λ' stanowiącym podzbiór wszystkich możliwych s-wymiarowych podprzestrzeni Λ

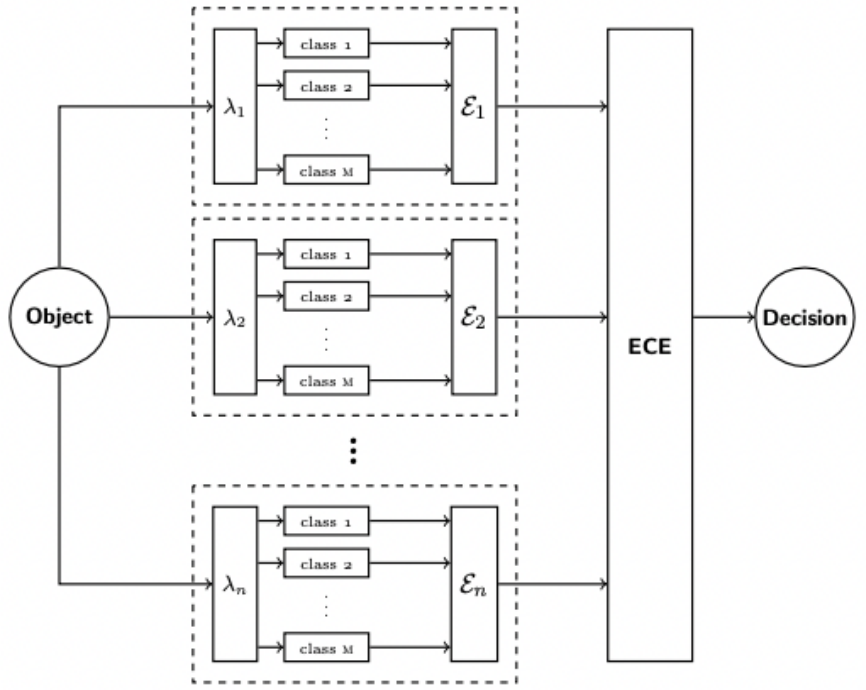
$$\begin{aligned} \Lambda' &\subset \Lambda, \qquad |\Lambda'| = N, \qquad N < L \\ \Pi &= \{\Psi_1, \Psi_2, \ldots, \Psi_N\}, \quad \Psi : \mathcal{X} \leftarrow \mathcal{M} \end{aligned} \tag{3.10}$$

Końcowy schemat przetwarzania stanowi trzypoziomowy zespół klasyfikatorów przedstawiony na Rysunku 3.3. Jego najniższy poziom to zbiór monochromatycznych warstw, budujących drugi poziom, klasyfikatorów bazowych zespołu dla podprzestrzeni λ_i, integrowanych na ostatnim poziomie we właściwy system wieloklasyfikatorowy *Exposer Classifier Ensemble* (ECE).

Zaproponowana w pracy metoda została poddana ewaluacji z wykorzystaniem pakietu *Weles*[7] opracowanego w ramach bieżących prac *Zespołu Uczenia Maszynowego, Katedry Systemów i Sieci Komputerowych*. Jak można zauważyć dzięki przeprowadzonej analizie, ECE często osiąga szczyt swojej efektywności przy relatywnie niskich wartościach hiperparametrów, jakkolwiek trudniejsze, wielowymiarowe i silnie niezbalansowane problemy przesuwają tę granicę wykazując, że wzrost próbkowania może mieć pozytywny wpływ na jakość klasyfikacji. W większości przypadków proponowane rozwiązanie osiąga rezultaty istotnie lepsze od pozostałych modeli bazowych, w żadnym przypadku nie okazując się najsłabszą metodą w stawce.

[7] Weles – Collection of various pattern recognition methods and experimental tools made by ML Group of Wrocław University of Science and Technology. — `https://github.com/w4k2/weles`

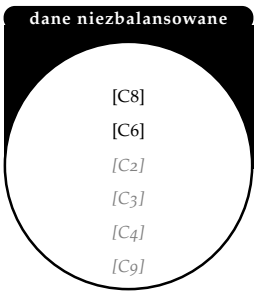
Algorytm ECE wykorzystując informację o rozkładzie klas wykazuje przewagi typowe z klasyfikatorów bayesowskich, osiągając najlepsze rezultaty dla danych niezbalansowanych. Dodatkowo, dzięki strategii próbkowania podprzestrzennego pozostaje on w dużym stopniu



Rysunek 3.3: Schemat zespołu ekspozerów.

odporny na zjawisko klątwy wielowymiarowości, ponieważ obiekty interpretowane są w nim jedynie w niskowymiarowych reprezentacjach. Łącząc te przewagi z regułą inferencyjną typową dla klasyfikatorów minimalnoodległościowych okazuje się on stabilnym rozwiązaniem dla trudnych przypadków danych jednocześnie niezbalansowanych i wielowymiarowych.

II – Algorytmy przetwarzania danych niezbalansowanych



Drugim istotnym wątkiem moich prac realizowanych po uzyskaniu stopnia naukowego doktora jest przetwarzanie danych niezbalansowanych. Elementy tego rodzaju przetwarzania obecne są już w pracach przedstawionych we wcześniejszej sekcji, ale wyodrębniłem też w badaniach jednoznaczny wątek traktujący ten problem jako podstawowe zagadnienie analiz.

Po prawej stronie diagram zaznaczający opisywane w ramach wątku przetwarzania danych niezbalansowanych, wliczający prace opisane w innych wątkach dominujących.

[C6]

Pawel Ksieniewicz. "Combining Random Subspace Approach with smote Oversampling for Imbalanced Data Classification". W: *Hybrid Artificial Intelligent Systems*. Red. Hilde Pérez García i in. Cham: Springer International Publishing, 2019, s. 660–673. ISBN: 978-3-030-29859-3. DOI: 10.1007/978-3-030-29859-3_56

Przykładem takiego podejścia jest praca [C6], proponująca hybrydowe rozwiązanie łączące dywersyfikację podprzestrzenną z nadpróbkowywaniem syntetycznym. Strategia taka pozwala na budowę zespołu klasyfikatorów integrującego oversampling w procedurze konstrukcji systemu wieloklasyfikatorowego w miejsce stosowanego standardowo potoku rozłącznych akcji *preprocessing* → *modelowanie*. W typowym podejściu do konstrukcji tego rodzaju systemów, faza preprocessingu nie jest specyficzna dla dywersyfikowanych modeli, a dokonuje się przed

uróżnorodnieniem puli.

Konstrukcja dowolnego zespołu klasyfikatorów wiąże się z dwiema podstawowymi trudnościami. Pierwszą jest zapewnienie różnorodnej puli klasyfikatorów pozwalającej na realizację niezależnych predykcji. Możemy osiągnąć to przez wykorzystywanie różnych modeli – w ramach dywersyfikacji heterogenicznej – lub też przez modyfikację zestawu danych treningowych – przy bardziej czytelnej w integracji dywersyfikacji homogenicznej. Proponowana metoda wykorzystuje to drugie podejście, w którym każdy klasyfikator jest trenowany na losowej podprzestrzeni zbioru uczącego. Redukcja przestrzenności problemu, poza zapewnieniem różnorodności, stanowi również czynnik różnicujący efekt nadpróbkowywania. Standardowe metody syntetycznego oversamplingu, takie jak SMOTE [Z25] czy ADASYN, opierają się najczęściej na ocenie podobieństwa wzorców określanego przez metryki dystansowe, którego struktura zmienia się wraz z każdą modyfikacją zbioru uczącego inną niż losowy obrót przestrzeni problemu [Z24].

[Z25] Nitesh V Chawla i in. "SMOTE: synthetic minority over-sampling technique". W: *Journal of artificial intelligence research* 16 (2002), s. 321–357

[Z24] Juan José Rodriguez, Ludmila I Kuncheva i Carlos J Alonso. "Rotation forest: A new classifier ensemble method". W: *IEEE transactions on pattern analysis and machine intelligence* (2006)

Drugą trudnością typową dla projektowania zespołu klasyfikatorów jest zapewnienie odpowiedniej reguły decyzyjnej, integrującej predykcje modeli dostępnych w puli. Najbardziej obiecującym rozwiązaniem w tym wypadku – pozwalającym na przekroczenie ograniczenia abstrakcyjnej reguły decyzyjnej Wyroczni[8] stawianego regułom opartym na głosowaniu większościowym – jest akumulacja wsparć klasyfikatorów bazowych. Należy jednak pamiętać, że wymaga ona wykorzystania wyłącznie klasyfikatorów udostępniających funkcję decyzyjną, klasyfikatorów problabilistycznych lub modeli o interpretacji probabilistycznej. Regułę taką można dodatkowo rozbudować przez wprowadzenie ważenia, które w rozważanym przypadku oparte było na wyznaczonej przez metrykę F_1 jakości osiąganej przez każdy z modeli wchodzących w skład puli.

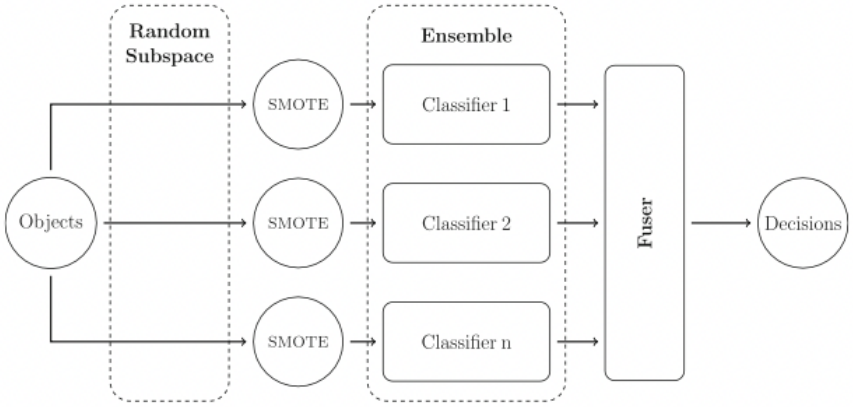
[8] Wyroczna – w zakresie uczenia zespołowego – stanowi abstrakcyjną regułę decyzyjną o nieuprawnionym dostępie do etykiet, podejmującą słuszną decyzję zawsze, kiedy skłania się ku niej co najmniej jeden z klasyfikatorów w puli.

Pełen schemat przetwarzania zaproponowanej metody przedstawiony został na Rysunku 3.4. Zgodnie z przedstawioną procedurą, dostępny zbiór uczący dzielony jest na podzbiory jego atrybutów, a następnie obiekty klasy mniejszościowej każdej zredukowanej tak reprezentacji są syntetyzowane do poziomu zbalansowania przez algorytm SMOTE, aby na każdej lokalnie zrównoważonej reprezentacji zbudować nowy model. Tak skonstruowana pula klasyfikatorów integrowana jest później przez akumulację wsparć ważoną przez wartość metryki F_1 osiągniętej na pełnym zbiorze uczącym.

Zaproponowana metoda została poddana ewaluacji eksperymentalnej z wykorzystaniem 30 zbiorów danych o różnej, wysokiej skali niezbalansowania, dostępnych w repozytorium KEEL[9]. W ramach ewaluacji przetestowano efektywność rozwiązania dla trzech klasyfikatorów o interpretacji probabilistycznej: *Gaussian Naive Bayes*, *Logistic Regression* i *Support Vector Machine*. Analiza miała na celu porównanie ze sobą sześciu podejść:

[9] Knowledge Extraction Evolutionary Learning – dataset repository `https://sci2s.ugr.es/keel/datasets`

- uczenia na oryginalnym zbiorze,
- globalnego oversamplingu SMOTE,
- dywersyfikacji przez losowe podprzestrzenie,



Rysunek 3.4: Schemat architektury zespołu klasyfikatorów zaproponowanego w pracy *Combining Random Subspace Approach with* SMOTE *Oversampling for Imbalanced Data Classification*.

- połączenia dywersyfikacji przez losowe podprzestrzenie z lokalnym oversamplingiem SMOTE,
- ważonego przez metrykę *F1-score* zespołu o surowej dywersyfikacji przez losowe podprzestrzenie,
- ważonego przez metrykę *F1-score* zespołu łączącego dywersyfikację przez losowe podprzestrzenie z lokalnym oversamplingiem SMOTE.

Analiza uzyskanych wyników pozwala zaobserwować, że niezależnie od wykorzystanego klasyfikatora bazowego, użycie globalnego oversamplingu SMOTE wpływa pozytywnie na jakość rozpoznawania, z reguły ulepszając model bazowy.

Patrząc na drugi istotny czynnik przetwarzania, wykorzystanie jedynie dywersyfikacji przez losowe podprzestrzenie, przy warunkach wysokiego niezbalansowania, prowadzi do bardzo niskich rezultatów, pogarszających nawet jakość klasyfikatora bazowego. Wprowadzenie ważenia zespołu ulepsza taki model, prowadząc do rezultatów lepszych niż bazowy, ale nadal gorszych niż globalny SMOTE.

Wykorzystanie obydwu tych strategii, w formie zespołu z lokalnym nadpróbkowywaniem, prowadzi do rezultatów odrobinę lepszych niż globalny SMOTE, co może sugerować, że pozytywny wpływ obu tych czynników jest niezależny od siebie i potencjalnie może okazać się komplementarny. Potwierdzają to rezultaty osiągane przez pełną propozycję rozwiązania, ważony zespół łączący dywersyfikację przez losowe podprzestrzenie z lokalnym oversamplingiem, którego jakość jest jednoznacznie najlepsza w puli analizowanych przykładów.

Praca uprawdopodabnia stwierdzenie, że wspólne wykorzystanie metod balansowania i dywersyfikacji, przy kontroli wpływu każdej podprzestrzeni na finalną predykcję przez ważenie zgodne z efektywnością w problemie niezbalansowanym, może prowadzić do zbliżenia się do optymalnego rozmieszczenia obiektów uczących w przestrzeni problemu i tym samym prowadzić do budowy modeli o wysokiej zdolności dyskryminacyjnej.

[C8]

Pawel Ksieniewicz. "Undersampled Majority Class Ensemble for highly imbalanced binary classification". W: *Proceedings of the Second International Workshop on Learning with Imbalanced Domains: Theory and Applications*. Red. Luís Torgo i in. T. 94. Proceedings of Machine Learning Research. PMLR, paź. 2018, s. 82–94. URL: https://proceedings.mlr.press/v94/ksieniewicz18a.html

Wykorzystanie metod resamplingu w konstrukcji systemów rozpoznawania dla danych niezbalansowanych nierozerwalnie niesie ze sobą ryzyko utraty informacji o potencjalnie istotnych obiektach klasy większościowej. Możliwa jest jednak mitygacja tego problemu, przez odpowiednie wykorzystanie podejścia zespołowego, które pozwoli na pełne wykorzystanie dostępnych danych bez konieczności syntetyzacji nowych wzorców lub redukcji obiektów istniejących. Propozycję algorytmu tego rodzaju przedstawiłem w pracy [C8].

Metoda ta dedykowana jest jedynie problemom binarnym o wysokim stopniu niezbalansowania, gdzie *imbalanced ratio* (IR) wynosi 1:9 i więcej. Złożone metody resamplingu, takie jak SMOTE czy ADASYN, pomimo wysokiej efektywności w wielu niezbalansowanych problemach rozpoznawania, nie odnajdują swojego zastosowania w scenariuszach skrajnych, gdzie klasa mniejszościowa reprezentowana jest jedynie przez kilka przykładów pośród których nie jest możliwa analiza sąsiedztwa pozwalająca na syntetyzację nowego obiektu. Teoretycznie możliwe jest w takiej sytuacji zastosowanie undersamplingu, ale jak pokazują badania, rozwiązania takie w sytuacjach silnego niezbalansowania nie pozwalają na uzyskanie stabilnego systemu rozpoznawania.

Popularnym rozwiązaniem w takiej sytuacji jest wykorzystanie standardowych zespołów opartych na *Baggingu* lub *Boostingu* – opartych na losowym próbkowaniu ze zwracaniem, które pozwalają na rozbicie dużego problemu w serię mniejszych. W pracy proponuję prostą obliczeniowo metodę, która również dokonuje rozbicia problemu rozpoznawania, zapewniając jednak obecność wszystkich obiektów dostępnych w zbiorze uczącym, eliminując również ryzyko nakładania się klas.

Model *Undersampled Majority Class Ensemble* (UMCE) buduje się zgodnie z następującymi krokami:

1. Podziel $\mathcal{DS}$ na zbiór mniejszościowy *MinC* i większościowy *MajC*.
2. Wyznacz współczynnik niezbalansowania IR jako proporcję liczności zbiorów *MinC* i *MajC*.
3. Wyznacz współczynnik k jako najbliższe całkowite zaokrąglenie IR.
4. Zrealizuj k-foldowy podział zbioru *MajC*, aby wytworzyć zbiór podzbiorów $MajC_1, MajC_2, \ldots, MajC_k$.
5. Dla każdego i w zakresie do k:
    6. Złącz zbiory $MajC_i$ i *MinC* w zbalansowany zbiór $\mathcal{TS}_i$.
    7. Zbuduj model Ψ_i na podstawie przykładów $\mathcal{TS}_i$ i dodaj go do zespołu.

Opcjonalnym elementem przetwarzania jest uzupełnienie puli modeli o dodatkowy klasyfikator zbudowany z wykorzystaniem typowego zastosowania algorytmu SMOTE. Został on włączony do metody aby

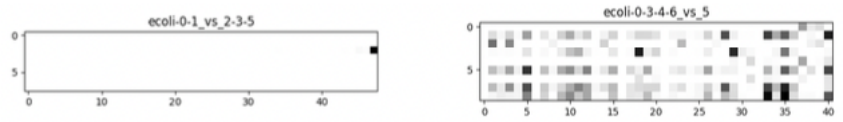
umożliwić weryfikację różnic w zyskach z obu strategii, podobnie jak miało to miejsce w poprzednio opisanej pracy.

Klasycznie już, istotnym elementem metody zespołowej jest również reguła integracji. W wypadku algorytmu UMCE zdecydowałem się odejść od prostych paradygmatów fuzji wsparć, rozwijając strategię podstawową (REG) i ważoną proporcjonalnie do jakości (WEI) o trzy dodatkowe propozycje:

nor Normalizowana przedziałowo akumulacja ważona – pozwalająca na istotne wzmocnienie najlepszego klasyfikatora w puli i całkowitą redukcję wpływu klasyfikatora najsłabszego.

con Dynamiczna akumulacja wsparć. Aby zapewnić lepsze wykorzystanie klasyfikatorów o większej "pewności"decyzji względem danego obiektu, decyzja dla każdego wzorca ważona jest przez bezwzględną różnicę pomiędzy wsparciami, która na potrzeby badań została nazwana *kontrastem*. Ilustracją tego podejścia jest Rysunek 3.5, na osiach X prezentujący próbki, a na osiach Y – klasyfikatory dostępne w puli. Białe punkty pokazują kontrast 1, a więc decyzję pewną, podczas gdy czarne powiązane są z kontrastem 0, a więc absolutnym brakiem pewności, czyli wzorcem leżącym bezpośrednio na granicy decyzyjnej.

nci Akumulacja wsparć ważona przez iloczyn nor i con.



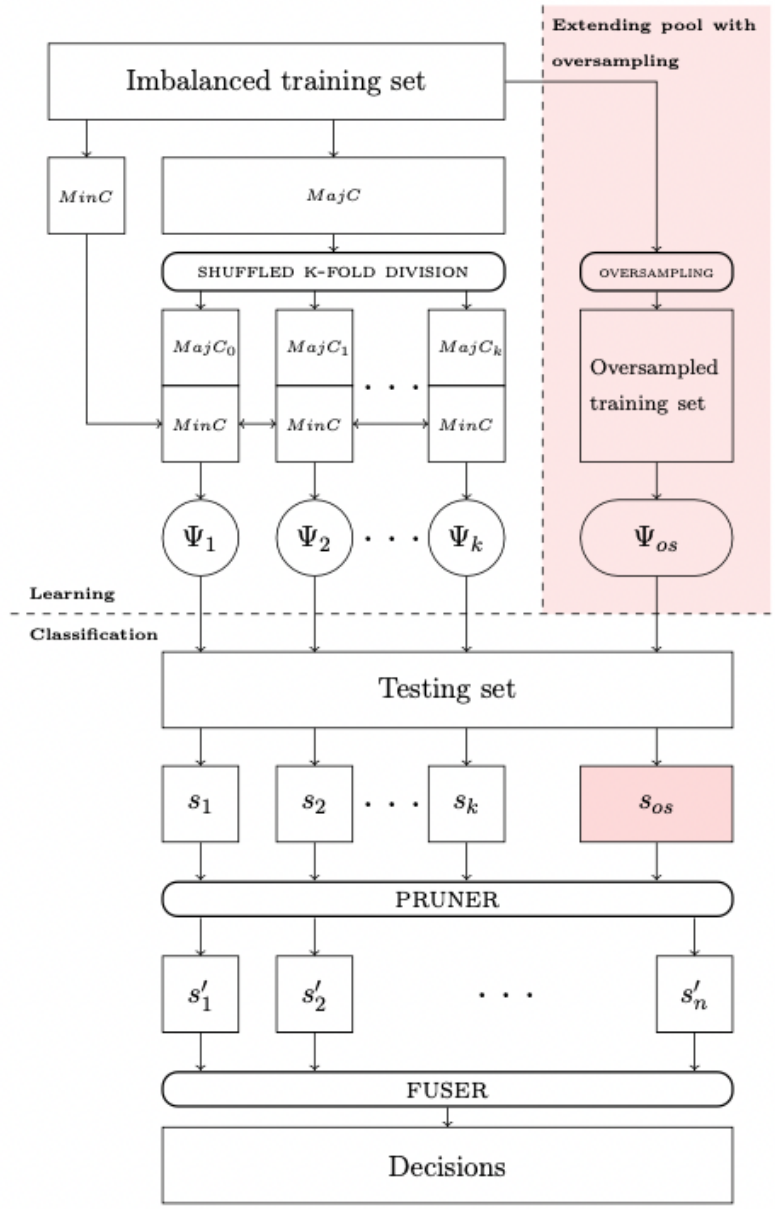
(*a*) Example of a *"sure"* ensemble (*b*) Example of *"unsure"* ensemble

Rysunek 3.5: Ilustracja rozkładu *kontrastu* w zespołach klasyfikatorów zbudowanych na dwóch zbiorach danych.

Należy pamiętać, że zaproponowana metoda budowy zespołu uzależnia jego wielkość od poziomu niezbalansowania, co dla danych bardzo silnie niezbalansowanych (przykładowo o IR 1:100) prowadzić będzie do konstrukcji bardzo dużego modelu hybrydowego. W związku z tym, zaproponowana została również metoda przycinania zespołu klasyfikatorów.

Typowe metody przycinania zespołów działają w trybie statycznym, analizując jakość modeli członkowskich i wycinając pulę klasyfikatorów o najniższym potencjale dyskryminacyjnym. W pracy zaproponowałem jednak metodę przycinania dynamicznego, dostosowującego skład zespołu do zbioru testowego. Zespół, otrzymując na wejściu zbiór testowy, generuje wektory wsparć (s_i) dla każdego klasyfikowanego obiektu, co pozwala zinterpretować wsparcia dla danej klasy z danego klasyfikatora jako zmienne losowe możliwe do analizy wzajemnej zależności statystycznej. W propozycji, wykorzystując test rankingowy, dokonuję klasteryzacji puli k modeli do n grup ($n \leq k$) celem uśrednienia wsparć i wag w obrębie każdej grupy przed końcową integracją.

Pełen schemat organizacji metody UMCE z rozbiciem na blok indukcyjny i predykcyjny przedstawiony został na Rysunku 3.6.

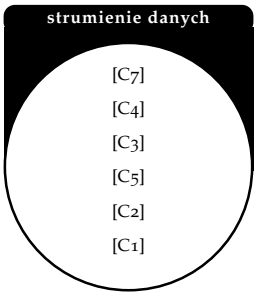


Rysunek 3.6: Schemat organizacji metody *Undersampled Majority Class Ensemble*.

Ewaluacja eksperymentalna metody UMCE została zrealizowana na 40 binarnych problemach klasyfikacji o IR większym niż 1:9. Aby umożliwić właściwe porównanie z metodami referencyjnymi zastosowano klasyczną, pięciofoldową walidację krzyżową, a w ocenie jakości modeli zastosowana została metryka zbalansowanej dokładności.

Metoda bazowa osiągnęła przewagę nad konkurencją jedynie w trzech przypadkach, podczas gdy standardowe metody over- i undersamplingu okazały się najlepsze w nie więcej niż dwóch przypadkach, niezależnie od zastosowanego klasyfikatora bazowego. Jednak, zarówno rozszerzenie puli klasyfikatorów UMCE o model wykorzystujący SMOTE, jak i zaproponowana metoda pruningu oraz mieszana metoda ważenia NCI prowadzą do jednoznacznie najlepszych rezultatów w porównaniu. Należy podkreślić skuteczność metody UMCE tym bardziej, że nawet najprostsza forma jej architektury, pozbawiona przycinania i ważenia zespołu, osiąga wyniki istotnie lepsze niż metody *state-of-the-art* dziedziny.

III – Algorytmy przetwarzania strumieni danych



Po prawej stronie diagram zaznaczający opisywane w ramach wątku przetwarzania strumieni danych, wliczający prace opisane w innych wątkach dominujących.

Istnieje wyraźne podobieństwo pomiędzy wsadowym podejściem do predykcji, wykorzystywanym w procedurze przycinania algorytmu UMCE, a typowymi strategiami przetwarzania strumieni danych, które swoją ewaluację opierają na protokole *Test-Then-Train*. Środowisko tego rodzaju pozwala nie tylko na wnioskowanie względem pojedynczego predykowanego obiektu, ale też udostępnia kontekst, w którym obiekt ten pojawia się w toku przetwarzania. Rozwiązania takie stanowią trzeci, kluczowy wątek prac realizowanych przeze mnie w ramach omawianego cyklu publikacji podejmującego tematykę klasyfikacji danych trudnych.

[C7]

Paweł Ksieniewicz i in. "Data stream classification using active learned neural networks". W: *Neurocomputing* 353 (2019), s. 74–82. DOI: 10.1016/j.neucom.2018.05.130

Pierwszą omawianą pracą z tej kategorii jest artykuł [C7]. Podejmuje on tematykę wykorzystania efektu *catastrophic forgetting* – typowego dla sieci neuronowych – w adaptacyjnej klasyfikacji strumieni danych podatnych na zjawisko dryfu koncepcji przy redukcji zaangażowania ekspertów w uzyskiwaniu oznaczenia danych.

Strategie uczenia się przy ograniczonej etykietyzacji dla problemów stacjonarnych, często opierają się na klasteryzacji dostępnych obiektów pozwalającej na transdukcyjną identyfikację prototypów, które po oznaczeniu przez eksperta stanowią później przypadki reprezentatywne dla wydzielonych podzbiorów zbioru uczącego. W wypadku uczenia inkrementalnego i strumieniowego częściej jednak wykorzystywany jest paradygmat uczenia aktywnego, w którym pewna wiedza – zakumulowana już przez przyrostowy model – może być wykorzystywana do identyfikacji tzw. przypadków trudnych w obrębie nowego, nieoznaczonego jeszcze wsadu.

Podstawowym zagadnieniem poruszanym w pracy jest istotna redukcja kosztów konstrukcji systemu rozpoznawania w środowisku strumieniowym przy jednoczesnym zachowaniu efektywności, którą uzyskałby przy pełnym dostępie do etykiet. W wielu zadaniach praktycznych niemożliwe jest pozyskanie odpowiedniego zbioru etykiet w racjonalnym czasie, a jednocześnie niemal zawsze wymaga ona wykorzystania czynnika ludzkiego, który jest zarówno kosztowny, jak i omylny – szczególnie jeżeli dotyczy oznaczania danych masowych i napływających z wysoką częstotliwością. Sprawia to, że projektowanie metod klasyfikacji, które zdolne są do budowy rzetelnych systemów rozpoznawania przy jedynie częściowej etykietyzacji jest jednocześnie dużym wyzwaniem, jak i nadal bardzo pożądanym celem badań. W ramach pracy proponowany jest hybrydowy model uczenia aktywnego, łączący podejścia typowe dla uczenia online wraz ze strategiami okna przesuwnego.

Przyjmujemy, że jeżeli decyzja dla zadanego obiektu wynika z wysokiej wartości wsparcia, leży on daleko od granicy decyzyjnej, a więc wykorzystanie go w budowie modelu nie będzie mieć istotnego wpływu na zmianę estymowanego prawdopodobieństwa *a posteriori*. W wypadku modeli interpretowanych probabilistycznie wysoka wartość funkcji

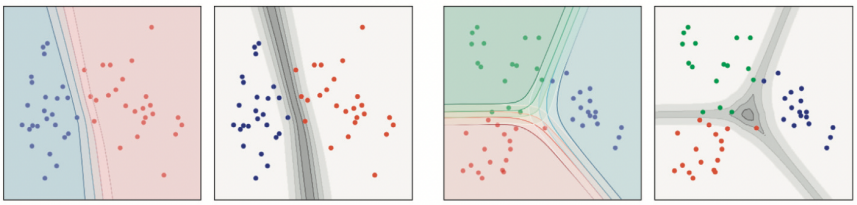
wsparcia oznacza też niewielkie prawdopodobieństwo błędu klasyfikacji. Stąd znacznie bardziej "*interesujące*" dla procedury modelowania są przypadki trudne, o niepewnej decyzji – rozumianej jako niewielka różnica pomiędzy wsparciami dla kategorii konkurujących między sobą o decyzję eksperta.

Aby sformalizować to założenie, zaproponowana została funkcja *RSFD* (*Relative Support Function Difference*), mierząca średnią różnicę pomiędzy największym prawdopodobieństwem i każdym z pozostałych elementów wektora wsparć dla zadanej obserwacji x

$$RSFD(x) = \frac{\sum_{i=1}^{M}[\max_{k \in \mathcal{M}}(F_k(x)) - F_i(x)]}{M-1}, \tag{3.11}$$

gdzie $Fi(x)$ to wartość wsparcia klasyfikatora i dla obiektu x.

Graficzna interpretacja funkcji *RSFD* dla przypadku dwu- i trzyklasowego zaprezentowana jest również na Rysunku 3.7. Barwy na ilustracji prezentują wsparcia dla klas, a spadek wartości w podglądzie monochromatycznym – trudność przypadków objętych obszarem.



Rysunek 3.7: Przykład rozkładu wsparć i wartości funkcji RSFD dla dwuwymiarowych zbiorów o dwóch i trzech klasach problemu.

Klasycznie, podejście takie wykorzystywane jest w *klasyfikacji z opcją odrzucenia* (ang. *classification with reject option* [Z26]), gdzie decyzje podejmowane są jedynie dla przypadków względem których klasyfikator jest w stanie podjąć dostatecznie pewną decyzję. W proponowanym przetwarzaniu zasada ta jest jednak odwracana i fakt *odrzucenia* danego obiektu jest powiązany z identyfikacją go jako obserwacji interesującej i skierowaniem do oznaczenia przez eksperta. Celem umożliwienia kontroli tego procesu wprowadzone zostały dwa hiperparametry metody:

- *próg* – wprowadzający wartość graniczną średniej różnicy wsparć, poniżej której obiekt uznawany jest za interesujący,
- *budżet* – określający jaki największy odsetek wsadu uczącego może zostać skierowany do etykietyzacji.

Całość zrealizowanej ewaluacji eksperymentalnej, jak w większości wchodzących w skład cyklu prac podejmujących przetwarzanie strumieniowe, wykonana została w oparciu o model *Multilayer Perceptron*. Efekty przeprowadzonej analizy pozwoliły na uprawdopodobnienie hipotezy o możliwości znaczącej redukcji kosztu etykietyzacji przy jednoczesnym zachowaniu zdolności dyskryminacyjnej modeli strumieniowych. Wstępne badania wykazały, że proste podejście budżetowe, pomijające całkowicie aspekt aktywnego doboru wzorców do etykietyzacji, liniowo degraduje krzywą uczącą, istotnie wydłużając czas konwergencji

[Z26] V. Franc, D. Prusa i V. Voracek. *Optimal strategies for reject option classifiers*. 2021. DOI: 10.48550/ARXIV.2101.12523

modelu. Wykorzystanie tej samej, lub nawet niższej liczby wzorców, dobranych jednak zgodnie z sugestiami dotychczasowego modelu, niweluje ten efekt, często pozwalając nawet na eliminację zapotrzebowania na eksperta w stabilnej fazie koncepcji.

Co szczególnie interesujące i możliwe do zaobserwowania w eksperymentach rozszerzonych, osiągany efekt w niektórych przypadkach pozwala nie tylko na zachowanie zdolności dyskryminacyjnych modelu o pełnej etykietyzacji, ale też wykazuje zdolność do eliminacji obiektów o negatywnym wpływie na model i prowadzi do osiągnięcia wcześniejszej zbieżności sieci neuronowej. Pokazuje to, że dziedzina przetwarzania strumieni danych nadal jest obszarem o dużym potencjale rozwijania badań, w szczególności tych dotyczących budowy modeli neuronowych w środowiskach o zmiennych koncepcjach.

Najpowszechniej podejmowanym w literaturze obszarem badań z zakresu strumieni danych jest analiza zdolności systemów uczących się do adaptacji do zmian w prawdopodobieństwie a posteriori problemu, a więc do dryfu koncepcji. Interesującym badawczo obszarem jest jednak również punkt styku pomiędzy tą tematyką, a problemem niezbalansowania danych, zdefiniowanym jako jeden z trzech głównych obszarów tematycznych prezentowanego cyklu.

Analizując to zagadnienie, w artykule [C4] wykazałem, że możliwe jest uwzględnienie estymowanego prawdopodobieństwa a priori strumienia danych o stałym stopniu niezbalansowania w istotnym statystycznie polepszeniu mocy generalizacyjnej modeli rozpoznawania. Podstawą analizy była weryfikacja potencjału przetwarzania końcowego (*postprocessingu*) w ulepszaniu modelu klasyfikacji strumieni niezbalansowanych statycznie bez ingerowania w procedurę budowy modelu.

W ramach pracy zaproponowana została metoda *Prior Imbalance Compensation* (PIC), generująca stosunkowo minimalny narzut obliczeniowy i zaprojektowana dla strumieni, w których nowe instancje pojawiają się z wysoką częstotliwością i w dużych ilościach lub dla przypadków o bardzo dużej skali niezbalansowania (w której odsetek klasy mniejszościowej stanowi mniej niż 5% ogółu dostępnych danych). Tak silne zaburzenie proporcji pomiędzy klasami, jak wskazane zostało już w opisie metody UMCE, uniemożliwia syntetyzację wzorców metodą SMOTE, a więc wymaga alternatywnych strategii przetwarzania.

Algorytm PIC, na podstawie dotychczas etykietowanych przypadków, przy najbardziej typowym dla literatury założeniu o stabilności niezbalansowania, z rosnącą precyzją estymuje prawdopodobieństwa *a priori* przetwarzanego problemu. Estymacja ta nie jest wykorzystywana przy budowie modelu, która odbywa się w klasyczny sposób, ale stanowi podstawę dla korekty uwzględnianej przy predykcji wsadowej. Jest ona możliwa do realizacji dla dowolnego modelu o interpretacji probabilistycznej lub o ciągłych funkcjach decyzyjnych, a więc przetestowana została w oparciu o klasyfikatory *Naive Bayes*, *k-Nearest Neighbors*, *Random Trees* oraz *Support Vector Machine*.

Zrealizowana ewaluacja eksperymentalna oceniała efektywność klasyfikacji strumieni niezbalansowanych (przetestowano problemy od

[C4]

Paweł Ksieniewicz. "The prior probability in the batch classification of imbalanced data streams". W: *Neurocomputing* 452 (wrz. 2021), s. 309–316. DOI: 10.1016/j.neucom.2019.11.126

zbalansowanych do niezbalansowanych w skali 1:20) zgodnie ze wskazaniami metryk zbalansowanej dokładności i *F1-score*. Kompensacja aprioryczna PIC pozwoliła na uzyskanie istotnego statystycznie polepszenia zbalansowanej dokładności klasyfikacji z niewielkim spadkiem metryki *F1-score* w każdym z testowanych scenariuszy. Szczególnie wyraźne jest to dla modeli opartych o *Support Vector Machine*.

Korekta aprioryczna ma też pewien wpływ na klasyfikację strumieni zbalansowanych, ale stanowi ona wtedy tylko niewielkie zaburzenie w jakości. Wraz ze zwiększaniem się skali niezbalansowania, przy zachowaniu tych samych analizowanych koncepcji, widać wyraźną degenerację modeli bazowych w domyślnej interpretacji predykcji przy jednoczesnym zachowaniu wysokiej sprawności przy zastosowaniu PIC. Należy zaznaczyć, że w obu przypadkach wykorzystywany jest dokładnie ten sam model, poddany jednak zmienionej interpretacji funkcji decyzyjnej.

Zbliżone obserwacje zostały dokonane zarówno na dryfach gradualnych, jak i nagłych. Algorytm PIC jest zdolny do ulepszenia jakości rozpoznawania dla każdego z rozważanych modeli klasyfikacji, ale największą odporność na niezbalansowania uzyskują dzięki niemu *Support Vector Machine*.

Rozszerzone badania nad potencjałem metody PIC pozwoliły na wstępną identyfikację nowego obszaru badawczego w zakresie przetwarzania niezbalansowanych strumieni danych, które pogłębiłem w dwóch kolejnych publikacjach prezentowanych w ramach konferencji *IEEE World Congress on Computational Intelligence 2021'* [C3] i *2022'* [Kom22], z których pierwsza – z racji na istotne rozwinięcie tej koncepcji – włączona została do opisywanego cyklu.

W pracy rozwijana jest analiza dotycząca przetwarzania niezbalansowanych strumieni danych z uwzględnieniem potencjalnej dynamiki tych zmian. Wraz ze współautorami proponujemy tam taksonomię zjawisk niezbalansowania strumieni danych, wydzielającą następujące kategorie:

BS **Strumienie zbalansowane (*balanced streams*)**, gdzie globalne prawdopodobieństwo *a priori* i każde z prawdopodobieństw lokalnych jest proporcjonalne i wzajemnie zależne.

SIS **Strumienie niezbalansowane statyczne (*statically imbalanced streams*)**, gdzie globalne prawdopodobieństwo *a priori* i prawdopodobieństwa dla każdego kolejnego okna sąsiadujących obiektów jest nieproporcjonalne, ale wzajemnie zależne.

DIS **Strumienie niezbalansowane dynamicznie (*dynamically imbalanced streams*)**, wśród których można wydzielić dwie kategorie:

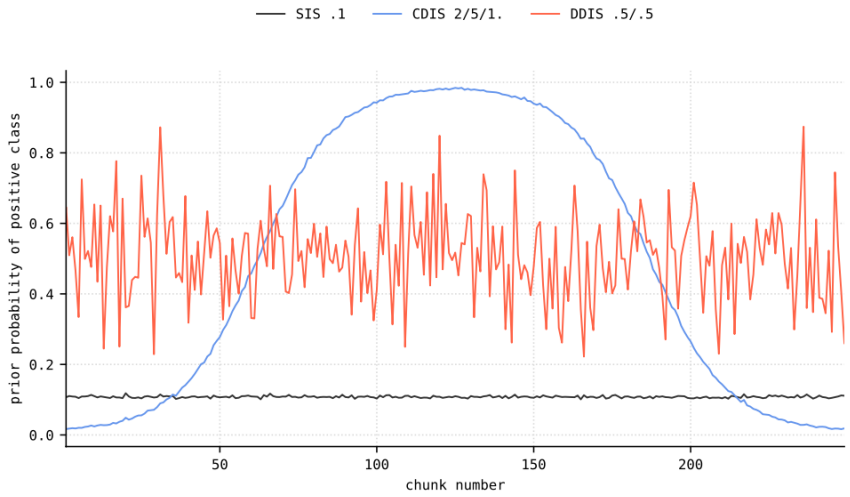
CDIS **Strumienie niezbalansowane dynamicznie w sposób ciągły (*continous dynamically imbalanced streams*)**, gdzie globalne prawdopodobieństwo *a priori* może różnić się od prawdopodobieństwa dla każdego kolejnego okna sąsiadujących obiektów, które nie są wzajemnie zależne, ale zmieniają się w sposób ciągły, umożliwiając obserwację trendów zmian.

[C3] Joanna Komorniczak, Pawel Zyblewski i Pawel Ksieniewicz. "Prior Probability Estimation in Dynamically Imbalanced Data Streams". W: *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, lip. 2021. DOI: 10.1109/ijcnn52387.2021.9533795

[Kom22] Joanna Komorniczak i Paweł Ksieniewicz. "Imbalanced Data Stream Classification Assisted by Prior Probability Estimation". W: *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022

DDIS **Strumienie niezbalansowane dynamicznie w sposób dyskretny (*discrete dynamically imbalanced streams*)**, gdzie globalne prawdopodobieństwo *a priori* może różnić się od prawdopodobieństwa dla każdego kolejnego okna sąsiadujących obiektów, które nie są wzajemnie zależne i zmieniają się dyskretnie, uniemożliwiając obserwację trendów zmian.

Przykładowe przebiegi prawdopodobieństwa wystąpienia klasy mniejszościowej dla przyjętej taksonomii zaprezentowane są na Rysunku 3.8. Jak można zauważyć, klasyczny przypadek strumienia niezbalansowanego, typowy dla literatury (SIS) charakteryzuje się stabilną proporcją o niewielkiej wariancji na całym przebiegu przetwarzania. Strumień niezbalansowany dynamicznie w sposób dyskretny (DDIS) również wykazuje pewną wartość oczekiwaną niezbalansowania, ale wariancja estymacji prawdopodobieństwa jest już w nim na tyle duża, że może mieć istotny wpływ na jakość modelu. Strumień niezbalansowany dynamicznie w sposób ciągły (CDIS) wykazuje charakter pośredni pomiędzy SIS i DDIS, cechując się niską lokalną wariancją, przy wysokiej wariancji globalnej.

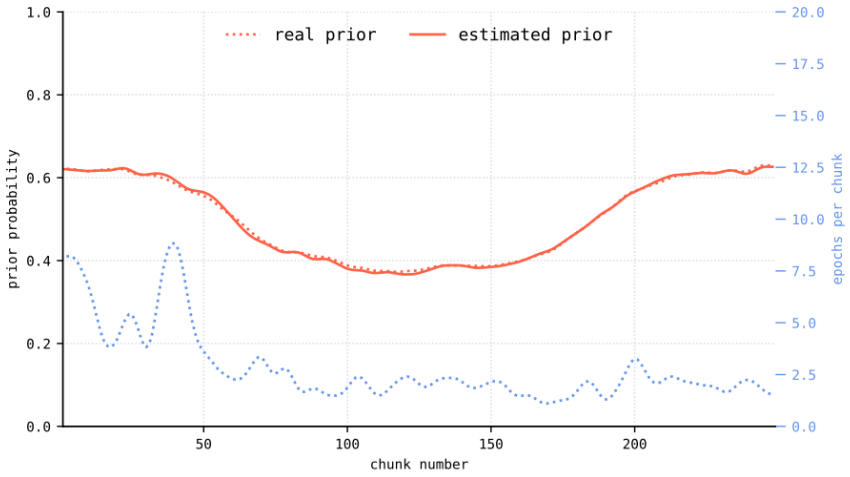


Rysunek 3.8: Prawdopodobieństwo *a priori* klasy pozytywnej dla kolejnych wsadów strumieni z każdej rozważanej kategorii strumieni niezbalansowanych.

Opisywana praca – poza taksonomią problemów – proponuje również pulę rozwiązań standardowych oraz metodę nienadzorowanej estymacji prawdopodobieństwa *a priori* zdolną do efektywnego rozpoznawania w – szczególnie trudnych – strumieniach o niezbalansowaniu dyskretnie-dynamicznym (DDIS). Algorytm *Dynamic Statistical Concept Analysis* (DSCA) wprowadza reprezentację koncepcji, jako zbioru jej standardowych miar statystycznych możliwych do wyznaczenia bez nadzoru etykiet, wraz z prostym zespołem regresorów sieci neuronowych. Modele regresji – budowane dla każdej klasy problemu – przy każdym kroku optymalizacji otrzymują tę samą reprezentację wsadu, jako zmienną objaśnianą przyjmując liczbę obiektów[10] przypisanej im klasy. Zespół taki integrowany jest do wektora prawdopodobieństwa *a priori* przez komplementarne skalowanie predykcji, pozwalając na późniejsze uwzględnienie w korekcie apriorycznej PIC.

[10] Wykorzystanie zespołu regresorów przyjmujących jako zmienną objaśnianą liczbę obiektów i integrującego końcową odpowiedź systemu przez proporcję – w toku badań – okazało się znacznie bardziej obiecującą strategią niż wyuczanie skali niezbalansowania. Podejście takie daje zarówno lepsze rezultaty, jak i umożliwia zastosowanie metody w klasyfikacji wieloklasowej.

Dodatkowym elementem metody DSCA jest dynamiczna kalibracja liczby epok uczenia, dostosowująca się do aktualnego poziomu błędu predykcji. W początkowej fazie przetwarzania wykonywanych jest więcej kroków optymalizacji – celem szybkiego zyskania zbieżności modelu – a z czasem wartość ta redukuje się do pojedynczych, korygujących aktualizacji (Rysunek 3.9).



Rysunek 3.9: Estymowane i rzeczywiste prawdopodobieństwo *a priori* modelu DSCA zestawione z liczbą epok uczących regresory na wsad przetwarzania.

Ewaluacja eksperymentalna metody przeprowadzona została z wykorzystaniem autorskiego generatora strumieni danych o dynamicznym niezbalansowaniu, który został zintegrowany z pakietem *stream-learn*. W ramach testowanej hipotezy, badania weryfikowały możliwość zbudowania takiego modelu predykcji prawdopodobieństwa *a priori*, który zdolny będzie do przewidywania w strumieniach niezbalansowanych dynamicznie w sposób dyskretny, w których standardowe podejścia będą wysoce niewydolne. Rezultaty pozwalają zaobserwować, że o ile w wypadku strumieni niezbalansowanych statycznie nie występują istotne różnice w osiąganych rezultatach, to w niemal każdym ze scenariuszy dynamicznie-ciągłego niezbalansowania, metoda DSCA okazuje się istotnie najlepszą w stosunku do wybranych metod referencyjnych.

Szczególnie interesujący jest jednak przypadek wszystkich scenariuszy DDIS, w których każda z metod referencyjnych wykazuje bardzo duży błąd – proporcjonalny do wariancji w niezbalansowaniu. Algorytm DSCA w takich przypadkach cechuje się bardzo wysoką efektywnością, podobną do osiąganej na pozostałych scenariuszach, okazując się jedyną dostępną obecnie w literaturze metodą odporną na niefunkcyjne zmiany w stopniu niezbalansowania.

Dalsze prace z zakresu predykcji prawdopodobieństwa *a priori*, wykorzystujące zaproponowany model DSCA, pozwoliły na uogólnienie obserwacji poczynionych wstępnie dla algorytmu PIC, względem ogólnego modelu *Multilayer Perceptron* w inkrementalnym przetwarzaniu danych.

[C5]

Paweł Ksieniewicz i in. "Fake News Detection from Data Streams". W: *2020 International Joint Conference on Neural Networks (IJCNN)*. 2020, s. 1–8. DOI: 10.1109/IJCNN48605.2020.9207498

Badania z zakresu przetwarzania strumieni danych często odnajdują również swoje zastosowania praktyczne. Przykładem takiego zastosowania zajmuje się praca [C5], która stanowi pierwszą podjętą w literaturze przedmiotu analizę wykorzystania metod strumieniowych w zagadnieniu detekcji źródeł dezinformacji, a precyzyjnie, wykrywania treści wpisujących się w zjawisko fake news.

Środowisko naukowe nie przyjęło jeszcze jednoznacznej i powszechnej definicji zjawiska *fake news* [Cho21]. W związku z tym, w badaniach podejmujących ten temat opieram się na określeniu go jako: *treści niezgodnej z konsensusem przyjętym w określonej grupie społecznej, mającej za zadanie zmienić ten konsensus w sposób działający na niekorzyść tej grupy*.

[Cho21] Michał Choraś i in. "Advanced Machine Learning techniques for fake news (online disinformation) detection: A systematic mapping study". W: *Applied Soft Computing* 101 (2021), s. 107050. DOI: 10.1016/j.asoc.2020.107050

Opisywana praca dokonuje przeglądu standardowych modeli rozpoznawania dla środowisk strumieniowych (*Hoeffding Trees*, *Multilayer Perceptron*, *Naive Bayes*), w kontekście przetwarzania z wykorzystaniem trzech standardowych strategii budowy modeli. Analizowana jest w niej jakość (*a*) modeli pojedynczych, (*b*) klasycznych zespołów zbudowanych zgodnie z procedurą SEA[11] oraz (*c*) modelu *Online Bagging*, dobierającego wagi dla wzorców na podstawie rozkładu Poissona.

[11] *Streaming Ensemble Algorithm* — podstawowy, prosty model zespołu klasyfikatorów o ustalonym limicie puli i jakościowym kryterium jej przycinania.

Ewaluacja eksperymentalna podjęta została w oparciu o zbalansowany zbiór danych *Getting Real about Fake news*, zinterpretowany jako sekwencja obiektów dzięki wykorzystaniu dostępnych w nim stempli czasowych. Należy pamiętać, że typowe modele NLP, w procedurze wektoryzacji generują najczęściej wielowymiarowe, rozproszone macierze, które wymagają odpowiedniej redukcji wymiarowości, aby umożliwić modelowi właściwe wnioskowanie. Jest to szczególnie istotne w środowisku strumieniowym, w którym jednym z kluczowych kryteriów jest czas przetwarzania. Każdy z modeli standardowych budowany był więc dodatkowo na zredukowanej reprezentacji problemu, wykorzystującej bazową metodę ekstrakcji w parze z (a) przycinaniem prostym po częstotliwości występowania n-gramu, (b) selekcją tokenów dokonaną przez metodę filtrową Chi^2 oraz (c) ekstrakcją atrybutów uzyskaną dzięki algorytmowi *Principal Components Analysis*.

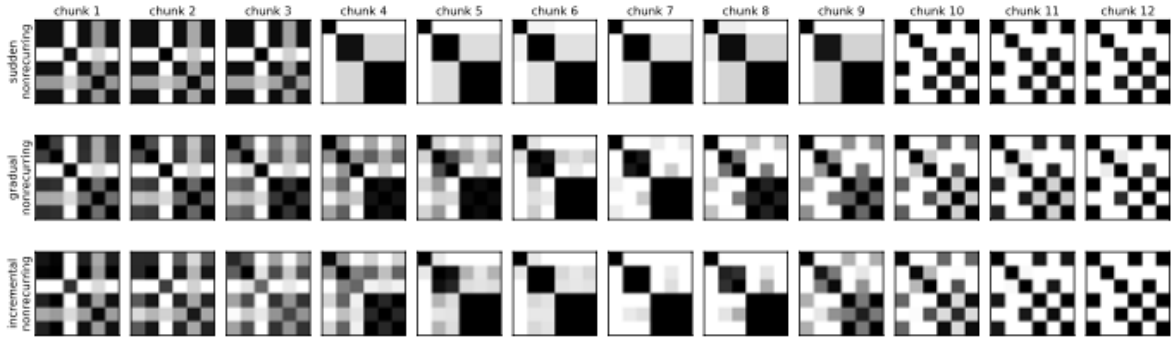
Przeciętnie w strumieniowym przetwarzaniu języka naturalnego prezentują się rezultaty osiągane przez pochodne algorytmu *Hoeffding Tree*. Niewielkie przewagi statystyczne uzyskują one jedynie w niskowymiarowych reprezentacjach, tracąc swoją zdolność dyskryminacyjną zanim w dostępnej przestrzeni problemu pojawi się dostatecznie wiele informacji, aby uzyskać model o maksymalnej mocy. Najlepszym uzyskanym modelem przetwarzania okazał się być *Multilayer Perceptron* poprzedzony analizą składowych głównych, redukującą dostępne tokeny do tysiąca niezależnych projekcji problemu, konstruujący zespół zgodnie z regułami *Online Bagging*. Przy tej samej konfiguracji przetwarzania, zarówno *Hoeffding Tree*, jak i podstawowy model *Naive Bayes* raportowały wynik nieznacznie przekraczający poziom klasyfikatora losowego. Stanowi to dodatkowe uprawdopodobnienie hipotezy o wysokim potencjale sieci neuronowych w przetwarzaniu strumieni danych w problemach trudnych, wykraczających poza standardowe zbiory benchmarkowe o atrybutach kategorycznych.

[C2]

Pawel Ksieniewicz. "Processing data stream with chunk-similarity model selection". W: *Applied Intelligence* (lip. 2022). DOI: 10.1007/s10489-022-03826-4

Właśnie ta hipoteza stanowi główny przyczynek do analizy podjętej w przedostatniej pracy wchodzącej w skład cyklu [C2]. Podejmuje ona temat nienadzorowanej analizy strumieni o atrybutach ilościowych, pozwalającej na tzw. identyfikację koncepcji.

Pierwszy istotny element pracy to propozycja *sygnatury koncepcji*, czerpiąca w swojej logice z dziedziny przetwarzania sygnałów wielowymiarowych. Rysunek 3.10 prezentuje przykładowe reprezentacje tego rodzaju, wyznaczone dla sześciu typowych strumieni danych zawierających dryfy koncepcji. Każdy strumień zawiera tu dwa dryfy, których punkty centralne znajdują się po trzecim i po dziewiątym wsadzie. Dla czytelności wizualizacji, strumienie zostały uproszczone do sześciu mocno skorelowanych ze sobą wymiarów, a same mapy cieplne prezentowane są jako ośmiobitowe obrazy znormalizowane odchyleniem standardowym sygnatur historycznych.



Rysunek 3.10: Wizualizacja macierzy auto-kowariancji wyznaczonych na dwunastu następujących po sobie wsadach sześciu strumieni danych zawierających dryfy koncepcji typowe dla literatury.

Jak można zaobserwować, struktura zależności pomiędzy atrybutami problemu, mierzona jako ich wariancja i kros-wariancja, zmienia się proporcjonalnie do dynamiki dryfów koncepcji. W najprostszym przypadku dryfu nagłego, sygnatury koncepcji pozwalają na jednoznaczną identyfikację aktualnej interpretacji problemu. Z drugiej strony, przy dryfach inkrementalnych i gradualnych, faza przejściowa koncepcji rejestrowana jest jako płynna, stopniowa zmiana sygnatur.

Proponowana w pracy metoda *Covariance-signature Concept Selector* (CSCS) wzoruje się w swojej procedurze uczącej na takich metodach *state-of-the-art*, jak *Adaptive Random Forest* (ARF), *Leveraging Bagging* (LBC) czy *Kappa Updated Ensemble* (KUE) i również buduje zespół klasyfikatorów, ale – w przeciwieństwie do nich – nie integruje modeli, dokonując selekcji najbardziej odpowiedniego modelu dla aktualnie interpretowanego wsadu. Oznacza to, że inaczej niż w dotychczas stosowanych metodach, decyzja podejmowana jest zawsze przez pojedynczy model, zgodnie z paradygmatem statycznej selekcji. Co szczególnie ważne, oparcie selekcji na sygnaturach koncepcji umożliwia identyfikację koncepcji bez etykiet, a więc w obrębie procedury predykcyjnej, a tym samym, identyfikację najbardziej odpowiedniego modelu z dostępnej puli. Procedura CSCS odrzuca standardowy paradygmat budowy zespołu, który opiera się w pozostałych metodach na zapewnieniu modeli o możliwie najwyż-

szej jakości i różnorodności, zastępując go paradygmatem najwyższego podobieństwa pomiędzy problemem i jego predyktorem. Taka zmiana pozwala zarówno na identyfikację nowych koncepcji, jak i na selekcję modelu odpowiedniego dla aktualnego, nieoznaczonego wsadu.

Metoda CSCS została poddana szerokiej ewaluacji eksperymentalnej, uwzględniającej zarówno osiem metod *state-of-the-art* (z trzech generacji zespołowych algorytmów przetwarzania strumieni), jak i trzy kategorie scenariuszy testowych:

- Dwa z nich uwzględniają ewaluację na danych syntetycznych – pochodzących zarówno z generatorów MOA, jak i generatora *stream-learn*
- Trzeci opiera się na strumieniach pół-syntetycznych, wstrzykujących dryfy do koncepcji rzeczywistych, pozwalając na ustrumieniowienie problemów stacjonarnych [Kom22b].

Dodatkowo, jako modele bazowe dla eksperymentów przyjęto zarówno *Hoeffding Tree* (w implementacji CVFDT), jak i *Multilayer Perceptron*. Podejście takie pozwoliło na odpowiednią analizę zachowania wszystkich rozważanych metod w problemach o różnej charakterystyce atrybutów (ilościowe, jakościowe i hybrydowe) przy różnych bazowych modelach klasyfikacji.

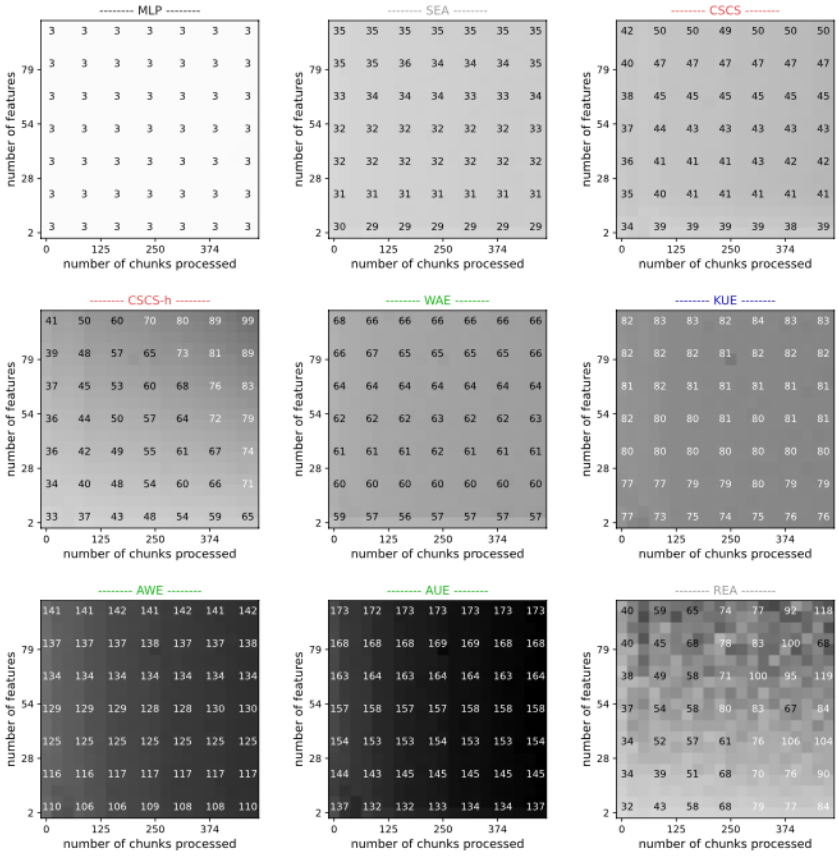
Globalna analiza, uwzględniająca również perceptron wielowarstwowy, pokazuje, że w wypadku wszystkich strumieni ilościowych, połączenie *Multilayer Perceptron* i CSCS wykazuje istotną przewagę statystyczną zarówno nad wszystkimi pozostałymi metodami rozpoznawania opartymi o sieci neuronowe, jak i względem modeli opartych o *Hoeffding Tree*.

Podobnie prezentują się wyniki osiągnięte dla strumieni pół-syntetycznych. W wypadku niskiej wymiarowości, w których sygnatura koncepcji jest niewielka i nie uzyskuje jeszcze pełnego potencjału różnicującego, pewną przewagę nad CSCS wykazują jeszcze modele ARF i LBC. Po przekroczeniu czterech wymiarów problemu, sygnatury CSCS pozwalają już na właściwą identyfikację i budują modele sieci neuronowych statystycznie istotnie lepsze od każdego z pozostałych testowanych rozwiązań.

Dodatkową właściwością CSCS jest minimalizacja narzutu obliczeniowego niezbędnego do konstrukcji modelu o dużej zdolności dyskryminacyjnej. Jak można zaobserwować na Rysunku 3.11, proponowana przeze mnie metoda wykazuje minimalnie większą złożoność jedynie względem najstarszego i najprostszego zespołu strumieniowego – *Streaming Ensemble Algorithm*. Ponadto, model ten, kiedy oparty jest o *Multilayer Perceptron*, wykazuje o rząd niższą złożoność obliczeniową niż bazowy model *Hoeffding Tree* w problemach wielowymiarowych, prezentując się jako niezwykle skuteczne narzędzie w przetwarzaniu wielowymiarowych strumieni danych o charakterystyce ilościowej.

[Kom22b] Joanna Komorniczak i Paweł Ksieniewicz. "Data stream generation through real concept's interpolation". W: *ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. 2022

Strategia ta opisana została w osobnym artykule, przyjętym do publikacji na konferencji ESANN'22, na którą przygotowaliśmy krótki film prezentujący zasadę działania generatora strumieni. `https://youtu.be/00KEJXAQrt8`



Rysunek 3.11: Czas (w milisekundach) wymagany do przetworzenia pojedynczego wsadu przez różne metody zespołowe wykorzystujące Multilayer Perceptron w zależności od liczby przetworzonych wsadów i wymiarowości problemu.

Zagadnienie identyfikacji zmian koncepcji, przyjmując już bardziej typowe podejście wykorzystujące detektory dryfów i przebudowę modeli bez identyfikacji rozkładu, rozważałem również w pracy [C1].

W pracy tej proponowany jest efektywny zespół detektorów dryfu *Statistical Drift Detection Ensemble* (SDDE), oparty na statystycznych miarach *drift magnitude* oraz *conditioned marginal covariate drift*, stanowiący przykład detektora agnostycznego, tj. niezależnego od odpowiedzi klasyfikatora. Należy zaznaczyć, że nie oznacza to budowy modelu nienadzorowanego, ponieważ wykorzystuje on etykiety obiektów do budowy reprezentacji wiedzy o rozkładzie klas, ale nie uzależnia swoich decyzji od zmian jakości klasyfikacji w funkcji czasu, jak robią to klasyczne detektory dryfu.

Miara *drift magnitude* definiowana jest przez dystans pomiędzy koncepcjami w punktach w czasie t i u, rozumiana jako dystans pomiędzy rozkładami atrybutów $P(X)$ w tychże

$$DM_{t,u} = D(P_t(X), P_u(X)), \tag{3.12}$$

gdzie dystans pomiędzy rozkładami estymowany jest przez metrykę Hellingera.

Druga wykorzystana miara, *conditioned marginal covariance drift* definiowana jest jako ważona suma odległości pomiędzy warunkowymi

[C1] Joanna Komorniczak, Paweł Zyblewski i Paweł Ksieniewicz. "Statistical Drift Detection Ensemble for batch processing of data streams". W: *Knowledge-Based Systems* 252 (2022), s. 109380. DOI: 10.1016/j.knosys.2022.109380

rozkładami prawdopodobieństwa dla możliwych kategorii $P(X|Y)$ pomiędzy punktami w czasie t i u. Wagi stanowią średnie prawdopodobieństwa wystąpienia obiektów danej klasy $P(Y)$ w obu punktach w czasie

$$\sigma_{t,u}^{X|Y} = \sum_{y \in Y} \left[\frac{P_t(y) + P_u(y)}{2} \frac{1}{2} \sum_{y \in Y} |P_t(\bar{x}|y) - P_u(\bar{x}|y)| \right]. \tag{3.13}$$

W metodzie SDDE miary nie są wyliczane na pełnej wymiarowości strumienia, a jedynie na jego podprzestrzeniach, czyniąc z niej rozwiązanie dedykowane strumieniom wielowymiarowym. Rozbicie problemu na podprzestrzenie nie tylko pozwala uniknąć problemów wynikających z klątwy wielowymiarowości, ale także zapewnia zespół detektorów w miejsce pojedynczego narzędzia pomiarowego. W jednej z wcześniejszych prac zespołu zauważyliśmy, że klasyczne metody integracji puli – typowe dla zadania klasyfikacji – nie przynoszą tak dobrych rezultatów w detekcji dryfu [Woz16], w związku z czym dla SDDE zaproponowana została integracja dwupoziomowa. W ramach detektorów bazowych, zbudowanych w obrębie tej samej podprzestrzeni), decyzja podejmowana jest na podstawie porównania aktualnych wartości metryk ze średnią harmoniczną wartości historycznych, opierając się na regule trzy-sigma. Integracja podprzestrzeni odbywa się już przez hiperparametr progu wzbudzenia, dobrany eksperymentalnie dla analizowanych strumieni. Przykład przetwarzania algorytmu SDDE zaprezentowany został na Rysunku 3.12.

[Woz16] Michał Woźniak i in. "A First Attempt to Construct Effective Concept Drift Detector Ensembles". W: *Advances in Intelligent Systems and Computing*. Springer International Publishing, paź. 2016, s. 27–34. DOI: 10.1007/978-3-319-47274-4_3

Za estymator rozkładów prawdopodobieństwa w każdym z niezbędnych elementów przyjęto jądrową estymację gęstości (*Kernel Density Estimation*).

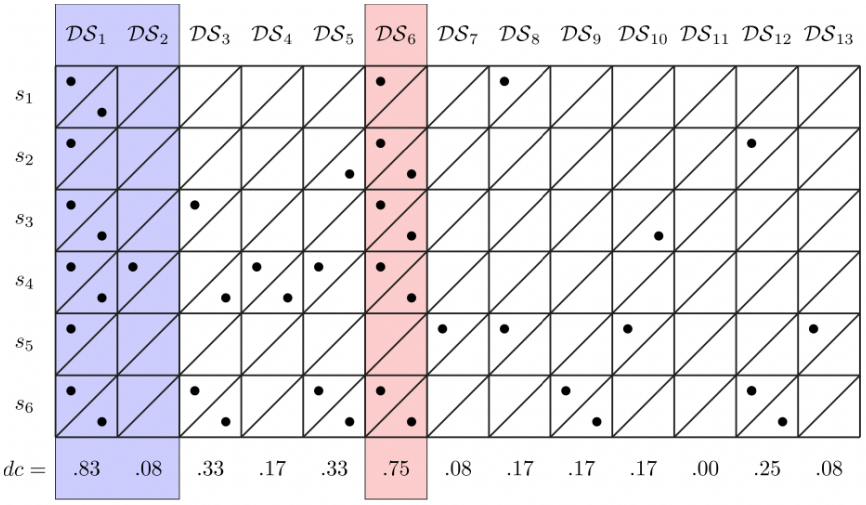
Istotnym uzupełnieniem pracy była propozycja korekty w standardowym podejściu do analizy efektywności detektorów dryfu. Jak zostało to już zauważone w literaturze, odradza się ewaluację detektorów wyłącznie w oparciu o jakość klasyfikacji, ponieważ podejście takie nie tylko utrudnia wyciąganie właściwych wniosków z badań, ale nawet promuje metody losowo przebudowujące modele co zadany interwał. Przedstawiona propozycja stanowi zbiór trzech metryk uzupełniających:

D_1 – miara najbliższego dryfu – średnia odległość każdej detekcji od najbliższego dryfu.

D_2 – miara najbliższej detekcji – średnia odległość każdego rzeczywistego dryfu do najbliższej detekcji.

R – współczynnik dryfu do detekcji – skalowana do zera w wartości oczekiwanej i wyznaczana z wartości bezwzględnej proporcja pomiędzy liczbą dryfów i detekcji.

Zrealizowana ewaluacja eksperymentalna pozwoliła na zweryfikowanie algorytmu SDDE jako efektywnego narzędzia detekcji dryfu, w szczególności w przypadkach trudnych. W ramach oceny zwalidowano jego jakość w zestawieniu z metodami *state-of-the-art* takimi jak HDDM (w odmianach HDDM_A i HDDM_W) czy ADWIN, oraz z klasycznymi detektorami DDM i EDDM. Jak można zauważyć na reprezentatywnym przypadku



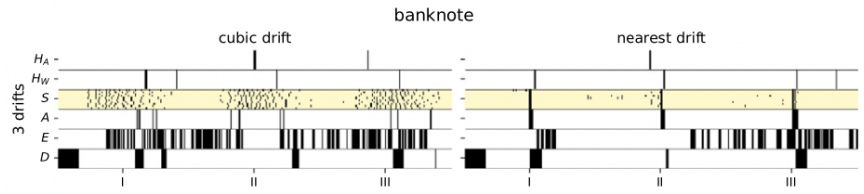
Rysunek 3.12: Przykład przetwarzania algorytmu *Statistical Drift Detection Ensemble*. Kolejne kolumny prezentują kolejne analizowane wsady, kolejne wiersze to następujące po sobie pary detektorów. Punkt oznacza wzbudzenie pojedynczego detektora, obszar niebieski – interwał ochronny zespołu detektorów, a obszar czerwony – wsad, w którym zespół identyfikuje dryf koncepcji. Opis procedury znajduje się w punktach poniżej ilustracji.

1. Początkowa faza przetwarzania, zaznaczona kolorem niebieskim, stanowi okres unieruchomienia wzbudzeń, w których niezależnie od odpowiedzi detektorów, zespół wskazuje na stabilność koncepcji.

2. Jednostkowe wzbudzenia detektorów bazowych – oznaczone czarnymi kropkami – nie prowadzą do przebudowania informacji o rozkładach, informując jedynie o aktualnym stanie odległości pomiędzy rozkładem pierwotnym i aktualnym.

3. Osiągnięcie progu wzbudzenia zespołu – zaznaczone na czerwono – prowadzi do wyzwolenia aktualizacji modelu, zapisania informacji o aktualnym rozkładzie w każdej podprzestrzeni i wykorzystywaniu jej jako punktu odniesienia w nadchodzących wsadach aż do momentu kolejnej detekcji dryfu.

(Rysunek 3.13), detektor ten nie tylko pozwala na jednoznaczną identyfikację dryfu w czytelnym scenariuszu dryfu nagłego, gdzie jedynie algorytmy HDDM$_w$ i ADWIN były zdolne do nawiązania z nim konkurencji, ale i w dryfach gradualnych, pozwalając na rozpoznanie zmian już w początkowej fazie dryfu, informując model na bieżąco o ich dynamice – podobnie jak metoda CSCS – będąc czułym nie tylko na stabilne koncepcje główne, ale też na każdą z koncepcji pośrednich.



Rysunek 3.13: Przykładowy wynik ewaluacji metody *Statistical Drift Detection Ensemble* (na żółto) w scenariuszu półsyntetycznym.

5.1 *Podsumowanie osiągnięcia naukowego i kierunki dalszych badań*

Prace opisane w ramach zaprezentowanego cyklu publikacji stanowią podsumowanie kolekcji metod pozwalających modelom klasyfikacji rozwiązywać szerokie spektrum problemów, w których często niemożliwe jest efektywne zastosowanie typowych rozwiązań znanych z literatury, a nawet algorytmów *state-of-the-art*.

Koncentracja trudności, przykładowo, w wielowymiarowych strumieniach danych wykazujących zarówno dryf koncepcji, jak i dynamiczne niezbalansowanie o charakterystyce uniemożliwiającej jego funkcyjną analizę[12], wymaga metod specyficznych, radzących sobie z wieloma wyzwaniami równocześnie. Zaproponowany przeze mnie zbiór algorytmów stara się wychodzić tym trudnościom naprzeciw, wykazując w większości przypadków statystycznie istotnie lepsze wyniki nad rozwiązaniami znanymi z literatury, zalecanymi często do typowych problemów, dostosowanymi do ogólnych przypadków klasyfikacji danych trudnych, ale przez to też niezdolnymi do efektywnego przetwarzania w przypadkach skrajnych.

[12] Strumienie niezbalansowane kategorii DDIS

Podczas projektowania eksperymentów takiego właśnie scenariusza [C7], większość implementacji wykorzystywanych w badaniach trudnych strumieni danych opierała się na frameworku MOA, stanowiącym ówczesny standard badawczy dla rozwiązań uczenia online, przeprowadzanych zgodnie z protokołem *Prequential Analysis*, opierając modele na odmianach *Hoeffding Tree*. Pakiet ten nie udostępniał jednak interfejsu programistycznego pozwalającego na badania z zakresu uczenia aktywnego, a więc konieczne było opracowanie dodatkowego oprogramowania pozwalającego na stosowną analizę.

[C7]

Paweł Ksieniewicz i in. "Data stream classification using active learned neural networks". W: *Neurocomputing* 353 (2019), s. 74–82. DOI: 10.1016/j.neucom.2018.05.130

Ze względu na rosnącą ówcześnie popularność pakietu *scikit-learn* i języka Python w środowisku badań nad sztuczną inteligencją, rozpocząłem prace nad nową biblioteką – nastawioną na przetwarzanie wsadowe i ewaluację uwzględniającą ilościową naturę danych strumieniowych. Jej pierwsza wersja rozwojowa użyta została jako podstawa do ówcześnie realizowanych badań, a stabilna wersja została opublikowana wraz z artykułem towarzyszącym w czasopiśmie Neurocomputing, w styczniu 2022 roku [Ksi22a].

[Ksi22a] P. Ksieniewicz i P. Zyblewski. "Stream-learn — open-source Python library for difficult data stream batch analysis". W: *Neurocomputing* 478 (2022), s. 11–21. DOI: https://doi.org/10.1016/j.neucom.2021.10.120

Takie podejście do badań sprawia, że zaproponowane rozwiązania nie stanowią jedynie rozważań teoretycznych, ograniczonych do odseparowanych środowisk eksperymentowania, ale w dominującej większości są dostępne dla społeczności akademickiej w formie zoptymalizowanych i udokumentowanych implementacji wchodzących w skład dostępnych w publicznych repozytoriach pakietów oprogramowania *stream-learn*, *weles* i *problexity*.

- *Stream-learn — open-source Python library for difficult data stream batch analysis*
  `https://github.com/w4k2/stream-learn`
  `https://stream-learn.readthedocs.io`

- *Weles — Collection of pattern recognition methods and experimental tools made by ML Group of Wrocław University of Science and Technology.*
  `https://github.com/w4k2/weles`
  `https://weles.readthedocs.io`

- *Problexity — an open-source python library containing the implementation of measures describing the complexity of the classification problem.*
  `https://github.com/w4k2/problexity`
  `https://problexity.readthedocs.io`

Pozwala to zarówno na replikację prezentowanych rezultatów badań, jak i dalszy ich rozwój, umożliwiający inkrementalne ulepszanie rozwiązań dedykowanych klasyfikacji danych trudnych. Wśród metod, które zaproponowałem w pracach wchodzących w skład cyklu należy wymienić:

- **Metodę zespołowej klasyfikacji obrazów nadwidmowych w oparciu o Extreme Learning Machines** – algorytm pozwalający na wykorzystanie, szczególnie pożądanego w praktycznych aplikacjach (w zagadnieniu rolnictwa precyzyjnego czy kontroli jakości), podejścia ręcznej inżynierii atrybutów projekcyjnych przy zastosowaniu szybko uczących się modeli neuronowych.

  dane wielowymiarowe

  Paweł Ksieniewicz, Bartosz Krawczyk i Michał Woźniak. "Ensemble of Extreme Learning Machines with trained classifier combination and statistical features for hyperspectral data". W: Neurocomputing *271 (2018), s. 28–37.* DOI: *10.1016/j.neucom.2016.04.076*

- **Genetyczną metodę doboru zespołu klasyfikatorów w środowisku niezbalansowanych danych wielowymiarowych** – metodę pozwalającą na optymalizację efektywności modeli w problemach niezbalansowanych o liczności klasy mniejszościowej uniemożliwiającej syntetyczne balansowanie problemu.

  dane wielowymiarowe
  dane niezbalansowane

  Paweł Ksieniewicz i Michał Woźniak. "Imbalanced Data Classification Based on Feature Selection Techniques". W: Intelligent Data Engineering and Automated Learning – IDEAL 2018. *Red. Hujun Yin i in. Cham: Springer International Publishing, 2018, s. 296–303.* DOI: *10.1007/978-3-030-03496-2_33*

- ***Exposer Classifier Ensemble*** – zespołową metodę klasyfikatora bazowego, odporną na klątwę wielowymiarowości i niezależną od stopnia niezbalansowania problemu przy jednoczesnym zachowaniu braku zależności statystycznej względem typowych modeli rozpoznawania.

  dane wielowymiarowe
  dane niezbalansowane

  Paweł Ksieniewicz i Michał Woźniak. "Dealing with the task of imbalanced, multidimensional data classification using ensembles of exposers". W: Proceedings of the First International Workshop on Learning with Imbalanced Domains: Theory and Applications. *Red. Paula Branco Luís Torgo i Nuno Moniz. T. 74. Proceedings of Machine Learning Research. PMLR, 22 Sep 2017, s. 164–175.* URL: *https://proceedings.mlr.press/v74/ksieniewicz17a.html*

- ***Subspace-driven SMOTE*** – architekturę zespołu klasyfikatorów przesuwającą fazę przetwarzania wstępnego do – zdatnego do krzyżowej oceny i dodatkowej kalibracji – zbioru podprzestrzeni problemu, redukujących negatywne skutki klątwy wielowymiarowości.

  dane wielowymiarowe
  dane niezbalansowane

Pawel Ksieniewicz. "Combining Random Subspace Approach with smote Oversampling for Imbalanced Data Classification". W: Hybrid Artificial Intelligent Systems. *Red. Hilde Pérez García i in. Cham: Springer International Publishing, 2019, s. 660–673.* ISBN: *978-3-030-29859-3.* DOI: *10.1007/978-3-030-29859-3_56*

- ***Undersampled Majority Class Ensemble*** – architekturę zespołu klasyfikatorów dedykowaną danym silnie niezbalansowanym, pozwalającą na pełne wykorzystanie dostępnych obiektów problemu, bez konieczności wprowadzania do przetwarzania obiektów syntetycznych.

  dane niezbalansowane

  Pawel Ksieniewicz. "Undersampled Majority Class Ensemble for highly imbalanced binary classification". W: Proceedings of the Second International Workshop on Learning with Imbalanced Domains: Theory and Applications. *Red. Luís Torgo i in. T. 94. Proceedings of Machine Learning Research. PMLR, paź. 2018, s. 82–94.* URL: *https://proceedings.mlr.press/v94/ksieniewicz18a.html*

- **Metodę aktywnego przetwarzania strumieni danych opartą o *Relative Support Function Difference*** – pozwalającą na racjonalizację kosztu pozyskiwania stronniczości eksperckiej w problemach o zmiennym prawdopodobieństwie *a posteriori*.

  strumienie danych

  Paweł Ksieniewicz i in. "Data stream classification using active learned neural networks". W: Neurocomputing *353 (2019), s. 74–82.* DOI: *10.1016/j.neucom.2018.05.130*

- ***Prior Imbalance Compensation*** – agnostyczną procedurę pozwalającą na zwiększenie mocy dyskryminacyjnej modeli klasyfikacji w strumieniach niezbalansowanych bez konieczności stosowania przetwarzania wstępnego, metod wbudowanych ani hybrydowych.

  dane niezbalansowane
  strumienie danych

  Paweł Ksieniewicz. "The prior probability in the batch classification of imbalanced data streams". W: Neurocomputing *452 (wrz. 2021), s. 309–316.* DOI: *10.1016/j.neucom.2019.11.126*

- ***Dynamic Statistical Concept Analysis*** – rozwinięcie metody PIC o efektywny estymator prawdopodobieństwa *a priori*, zdolny do osiągania wysokiej efektywności nawet w skrajnych przypadkach strumieni o niezbalansowaniu dyskretnie-dynamicznym.

  dane niezbalansowane
  strumienie danych

  Joanna Komorniczak, Pawel Zyblewski i Pawel Ksieniewicz. "Prior Probability Estimation in Dynamically Imbalanced Data Streams". W: 2021 International Joint Conference on Neural Networks (IJCNN). *IEEE, lip. 2021.* DOI: *10.1109/ijcnn52387.2021.9533795*

- ***Covariance-signature Concept Selector*** – zespołową metodę klasyfikacji trudnych strumieni danych o cechach ilościowych, dedykowaną przetwarzaniu z wykorzystaniem modeli sieci neuronowych.

  dane wielowymiarowe
  dane niezbalansowane
  strumienie danych

  Pawel Ksieniewicz. "Processing data stream with chunk-similarity model selection". W: Applied Intelligence *(lip. 2022).* DOI: *10.1007/s10489-022-03826-4*

- ***Statistical Drift Detection Ensemble*** – zespołowy detektor dryfu dedykowany rozpoznawaniu zmian w rozkładach *a posteriori* dla strumieni wielowymiarowych, stanowiący metodę agnostyczną – niezależną od wykorzystywanego modelu klasyfikacji.

  strumienie danych

  Joanna Komorniczak, Paweł Zyblewski i Paweł Ksieniewicz. "Statistical Drift Detection Ensemble for batch processing of data streams". W: Knowledge-Based Systems *252 (2022), s. 109380.* DOI: *10.1016/j.knosys.2022.109380*

Oprócz tematyki zawartej w prezentowanym cyklu moje zainteresowania naukowe dotyczą również innych tematów sztucznej inteligencji. Zagadnienie detekcji źródeł dezinformacji i klasyfikacji *fake news*, który poruszyłem tu w kontekście przetwarzania strumieniowego, rozwijam w kolejnych pracach analitycznych, w których znajdują się zarówno propozycje nowych algorytmów, jak i najaktualniejszy obecnie artykuł przeglądowy [Cho21] stanowiący zbiór punktów wyjścia dla prac prowadzonych w ramach kierowanego przeze mnie projektu SWAROG[13].

Zagadnienie przetwarzania strumieni danych analizuję również w kontekście zespołowych detektorów dryfu, strumieni niezbalansowanych skrajnie, modyfikacji standardowych metryk ważenia zespołów klasyfikatorów, dalszych prac nad uczeniem aktywnym czy oceny potencjału balansowania strumieni.

Dane niezbalansowane analizuję też nadal w ich stacjonarnej odmianie, zarówno dla danych sygnałowych jak i tabelarycznych, uwzględniając tu zarówno wątek integracji geometrycznej [Ksi21k] jak i potencjału algorytmów genetycznych w dywersyfikacji zespołów. Podobnie, rozwijam też wątek przetwarzania danych wielowymiarowych, zarówno w ujęciu tabelarycznym, jak i sygnałowym.

Ponadto, w swoich badaniach przykładam szczególną wagę do poprawności protokolarnej, o czym świadczyć może moje współautorstwo w pracy przeglądowej przedstawiającej dobre praktyki projektowania rzetelnych eksperymentów [Sta21]. Opisane przedsięwzięcie stanowi zarówno odpowiedź na aktualne i podlegające intensywnym badaniom problemy, jak i zbiór stabilnych rozwiązań, możliwych do zastosowania i stosowanych w rzeczywistych problemach.

[Cho21] Michał Choraś i in. "Advanced Machine Learning techniques for fake news (online disinformation) detection: A systematic mapping study". W: *Applied Soft Computing* 101 (2021), s. 107050. DOI: 10.1016/j.asoc.2020.107050

[13] Na tropie fake newsów, czyli projekt SWAROG https://wit.pwr.edu.pl/aktualnosci/na-tropie-fake-newsow-\-czyli-projekt-swarog-5.html

[Ksi21k] Paweł Ksieniewicz, Paweł Zyblewski i Robert Burduk. "Fusion of linear base classifiers in geometric space". W: *Knowledge-Based Systems* 227 (wrz. 2021). DOI: 10.1016/j.knosys.2021.107231

[Sta21] Katarzyna Stapor i in. "How to design the fair experimental classifier evaluation". W: *Applied Soft Computing* 104 (2021). DOI: https://doi.org/10.1016/j.asoc.2021.107219

Wykorzystana literatura pomocnicza

[Z1] P. Sulima-Samujłło. *Kolekcje polskich i dalekowschodnich (Chiny, Japonia, Korea) bibliotek cyfrowych. Analiza porównawcza*. 2017

[Z2] W. Van der Aalst, M. Bichler i A. Heinzl. *Robotic process automation*. 2018

[Z3] S. Zhang i in. "Deep learning based recommender system: A survey and new perspectives". W: *ACM Computing Surveys (CSUR)* (2019)

[Z4] S. M. Shavarani i in. "Application of hierarchical facility location problem for optimization of a drone delivery system: a case study of Amazon prime air in the city of San Francisco". W: *The International Journal of Advanced Manufacturing Technology* (2018)

[Z5] D. Neupane i J. Seok. "Bearing fault detection and diagnosis using case western reserve university dataset with deep learning approaches: A review". W: *IEEE Access* (2020)

[Z6] S. Miller i in. "Machine learning, ethics and law". W: *Australasian Journal of Information Systems* (2019)

[Z7] J Lighthill. *Artificial intelligence: a general survey. Science Research Council*. 1973

[Z8] J. McCarthy, E. Feigenbaum i J. Lederberg. *Artificial Intelligence Project*. Spraw. tech. Progress Report, 1973

[Z9] R. Tadeusiewicz. *Krótka historia informatyki*. Wydawnictwo RM, 2019

[Z10] E. Alpaydin. *Machine learning*. MIT Press, 2021

[Z11] Y. Sun, A. Wong i M. Kamel. "Classification of imbalanced data: A review". W: *International journal of pattern recognition and artificial intelligence* (2009)

[Z12] B. Krawczyk. "Learning from imbalanced data: open challenges and future directions". W: *Progress in Artificial Intelligence* (2016)

[Z13] V. Ganganwar. "An overview of classification algorithms for imbalanced datasets". W: *International Journal of Emerging Technology and Advanced Engineering* (2012)

[Z14] S. Kotsiantis. "Decision trees: a recent overview". W: *Artificial Intelligence Review* (2013)

[Z15] P. Alaba i in. "Towards a more efficient and cost-sensitive extreme learning machine: A state-of-the-art review of recent trend". W: *Neurocomputing* (2019)

[Z16] S. Sagiroglu i D. Sinanc. "Big data: A review". W: *2013 international conference on collaboration technologies and systems (CTS)*. IEEE. 2013

[Z17] T. Chan, G. Golub i R. LeVeque. "Updating formulae and a pairwise algorithm for computing sample variances". W: *COMPSTAT 1982 5th Symposium held at Toulouse 1982*. Springer. 1982

[Z18] V. da Costa, A. de Leon Ferreira, S. Junior i in. "Strict very fast decision tree: a memory conservative algorithm for data stream mining". W: *Pattern Recognition Letters* 116 (2018), s. 22–28

[Z19] Ian Goodfellow, Yoshua Bengio i Aaron Courville. *Deep learning*. MIT press, 2016

[Z20] Y. Wu i in. "Large scale incremental learning". W: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, s. 374–382

[Z21] B. Krawczyk i in. "Ensemble learning for data stream analysis: A survey". W: *Information Fusion* (2017)

[Z22] J. Gama i in. "A survey on concept drift adaptation". W: *ACM computing surveys (CSUR)* (2014)

[Z23] Guang-Bin Huang, Qin-Yu Zhu i Chee-Kheong Siew. "Extreme learning machine: theory and applications". W: *Neurocomputing* 70.1-3 (2006), s. 489–501

[Z24] Juan José Rodriguez, Ludmila I Kuncheva i Carlos J Alonso. "Rotation forest: A new classifier ensemble method". W: *IEEE transactions on pattern analysis and machine intelligence* (2006)

[Z25] Nitesh V Chawla i in. "SMOTE: synthetic minority over-sampling technique". W: *Journal of artificial intelligence research* 16 (2002), s. 321–357

[Z25] V. Franc, D. Prusa i V. Voracek. *Optimal strategies for reject option classifiers*. 2021. DOI: 10.48550/ARXIV.2101.12523

6 *Informacja o wykazywaniu się istotną aktywnością naukową albo artystyczną realizowaną w więcej niż jednej uczelni, instytucji naukowej lub instytucji kultury, w szczególności zagranicznej*

Po zatrudnieniu na stanowisku adiunkta, w październiku 2017 roku, zmieniłem główne zainteresowania badawcze, skupiając się na klasyfikacji danych niezbalansowanych i przetwarzaniu strumieni danych, ze szczególnym uwzględnieniem wsadowego paradygmatu przetwarzania.

Dalsze prace starałem się realizować w sposób, który pozwala na retencję wiedzy i stosowanych metod, przykładając szczególną wagę do replikowalności eksperymentów i otwartości oprogramowania opracowywanego w ramach bieżących prac *Katedry Systemów i Sieci Komputerowych*[14]. Silną współpracę w tym okresie prowadziłem zarówno z doktorantami zatrudnionymi w jednostce, z których dwóch uzyskało w tym roku stopień doktora, jak i z pracownikami innych zespołów badawczych *Politechniki Wrocławskiej*.

Szczególnie istotny jest dla mnie aspekt multidyscyplinarności badań, która pozwala na odnajdywanie szerszych zastosowań proponowanych przeze mnie metod. Wyraża się ona, między innymi, w ścisłej współpracy z *Zespołem Sieci Komputerowych*, gdzie wspólnie z badaczami z *Instytutu Łączności*[15] budujemy algorytmy wspomagające optymalizację kognitywnych sieci optycznych z wykorzystaniem modeli regresji [Ksi20s]. Wykorzystując zdobytą wcześniej wiedzę z zakresu przetwarzania cyfrowych sygnałów wielowymiarowych, angażuję się także w badania z zakresu bioinformatyki, gdzie wspólnie z pracownikami *Katedry Inżynierii Biomedycznej* oraz *School of Optometry and Vision Science*[16] i *Uniwersytetu Medycznego we Wrocławiu* analizujemy potencjał metod uczenia maszyn w zagadnieniu wczesnej detekcji jaskry [Sul21b].

Po zakończeniu doktoratu nawiązałem też silną współpracę z pracownikami *Zakładu Systemów Teleinformatycznych Wydziału Telekomunikacji, Informatyki i Elektrotechniki, Politechniki Bydgoskiej*. W ramach współpracy zrealizowaliśmy międzynarodowy projekt *SocialTruth*, finansowany ze

[14] Większość badań pracowników Katedry z ostatnich pięciu lat publikowana jest również jako oprogramowanie eksperymentalne dostępne na prowadzonym przeze mnie profilu organizacji na GitHub: https://github.com/w4k2

[15] *Instytut Łączności – Państwowy Instytut Badawczy* https://itl.waw.pl/

[Ksi20s] Paweł Ksieniewicz i in. "Pattern Recognition Model to Aid the Optimization of Dynamic Spectrally-Spatially Flexible Optical Networks". W: *Lecture Notes in Computer Science*. Springer International Publishing, 2020, s. 211–224

[16] *School of Optometry and Vision Science*, Brisbane, Australia

[Sul21b] Dominika Sułot i in. "Glaucoma classification based on scanning laser ophthalmoscopic images using a deep learning ensemble method". W: *PLOS ONE* 16.6 (czer. 2021), s. 1–12. DOI: 10.1371/journal.pone.0252339

środków programu *EU Horizon 2020*, prowadzony w konsorcjum z 11 państw *Unii Europejskiej*[17]. W latach 2019–2021 byłem zatrudniony w tym projekcie, publikując cztery prace naukowe, a dwie kolejne oczekują obecnie na recenzje.

Osiągnięcia poczynione w toku realizacji projektu *SocialTruth* pozwoliły nam również opracować wniosek projektowy w ramach programu INFOSTRATEG I[18], poświęcony zagadnieniu klasyfikacji fake news w języku polskim, który uzyskał maksymalną punktację NCBiR, zdobył finansowanie na poziomie ponad ośmiu milionów złotych [19] i w którym od grudnia 2021 roku pełnię rolę kierownika badawczo-rozwojowego. Projekt *SWAROG* realizowany jest w konsorcjum z *Politechniką Bydgoską* i przedsiębiorstwem *Matic SA*.

Wspólna praca w międzynarodowym konsorcjum – m. in. podczas spotkań projektowych w Paryżu i Rzymie – pozwoliła mi także na nawiązanie współpracy z innymi europejskimi ośrodkami w zakresie wspólnych publikacji. Wspólne badania z zakresu klasyfikacji *fake news* z zespołami *National Technical University of Athens*[20] i *Universidad de Burgos*[21] doprowadziły do opracowania pracy przeglądowej z zakresu metod detekcji źródeł dezinformacji [Cho21]. W ubiegłym roku pełniłem też rolę członka komisji doktorskiej przewodu Nuno Basutro z *Universidad de Burgos*, który podczas swoich studiów doktoranckich odbył staż na *Politechnice Wrocławskiej*.

Wymiana doświadczeń z zakresu projektowania eksperymentów rozpoznawania wzorców – ze szczególnym uwzględnieniem metod statystycznego testowania hipotez – podjęta wspólnie z zespołami *Politechniki Śląskiej* i *University of Granada*[22] pozwoliły z kolei na opracowanie pracy przeglądowej dotyczącej dobrych praktyk eksperymentowania z algorytmami klasyfikacji [Sta21]. Wprowadzamy tym zarówno zbiór wytycznych poprawnej ewaluacji, jak i prezentujemy praktyczne przykłady powszechnych, acz błędnych strategii oceny, które często prowadzą do uzyskiwania niejednoznacznych wniosków z badań.

W toku swojej pracy naukowej odbyłem także trzy krótko i średnioterminowe staże naukowe:

- Dwa staże naukowe zrealizowałem w ramach współpracy z *Universidad del Pais Vasco*[23], które zakończyły się nawiązaniem trwałej współpracy badawczej z zespołem *Group Faculty of Informatics* kierowanym przez profesora Manuela Granę [Ksi17p]. Współpraca ta wyraża się zarówno we wspólnych pracach badawczych, jak i w wymianie doświadczeń projektowych, której podsumowaniem może być wykład, który wygłosiłem na zaproszenie podczas konferencji *CybSPEED'22*. Współpraca nawiązana na wczesnym etapie pracy badawczej z *Universidad del Pais Vasco* była też bardzo ważnym czynnikiem w toku realizacji mojego doktoratu, ponieważ to podczas stażu Borja Ayerdi z zespołu prof. Manuela Grany realizowałem swoje pierwsze badania z zakresu przetwarzania obrazów nadwidmowych.

- Trzeci staż naukowy zrealizowałem w *Virginia Commonwealth University*[24], co pozwoliło mi na rozwinięcie wstępnych koncepcji badań

[17] *SocialTruth* http://socialtruth.eu

[18] https://www.gov.pl/web/ncbr/infostrateg-i-konkurs

[19] *System Wykrywania Dezinformacji Metodami Sztucznej Inteligencji* https://www.kssk.pwr.edu.pl/projects/swarog

[20] *National Technical University of Athens*, Ateny, Grecja

[21] *Universidad de Burgos*, Burgos, Hiszpania

[Cho21] Michał Choraś i in. "Advanced Machine Learning techniques for fake news (online disinformation) detection: A systematic mapping study". W: *Applied Soft Computing* 101 (2021), s. 107050. DOI: 10.1016/j.asoc.2020.107050

[22] *University of Granada*, Granada, Hiszpania

[Sta21] Katarzyna Stapor i in. "How to design the fair experimental classifier evaluation". W: *Applied Soft Computing* 104 (2021). DOI: https://doi.org/10.1016/j.asoc.2021.107219

[23] *Universidad del Pais Vasco* San Sebastian, Hiszpania

[Ksi17p] Paweł Ksieniewicz, Manuel Grana i Michał Woźniak. "Paired feature multilayer ensemble–concept and evaluation of a classifier". W: *Journal of Intelligent & Fuzzy Systems* 32.2 (2017), s. 1427–1436. DOI: 10.3233/JIFS-169139

[24] *Virginia Commonwealth University* Richmond, VA, USA

z zakresu klasyfikacji danych trudnych [C10]. W ramach rozwijania zakresu współpracy analizujemy aktualnie tematykę *wyjaśnialnej sztucznej inteligencji* (ang. *explainable AI*) dla zagadnień przetwarzania strumieni danych oraz homogeniczne metody dywersyfikacji puli klasyfikatorów oparte o rozkłady niejednostajne.

[C10]

Paweł Ksieniewicz, Bartosz Krawczyk i Michał Woźniak. "Ensemble of Extreme Learning Machines with trained classifier combination and statistical features for hyperspectral data". W: *Neurocomputing* 271 (2018), s. 28–37. DOI: 10.1016/j.neucom.2016.04.076

7 *Informacja o osiągnięciach dydaktycznych, organizacyjnych oraz popularyzujących naukę lub sztukę*

Podczas dziewięciu lat pracy naukowo-dydaktycznej prowadziłem dziewiętnaście kursów dla studentów kierunków *Informatyka*, *Informatyka Techniczna* i *Teleinformatyka*, w siedmiu z nich będąc głównym wykładowcą i autorem materiałów dydaktycznych:

1. Metody sztucznej inteligencji
2. Methods of Computational Intelligence and Decision Making,
3. Obrazowanie biomedyczne,
4. Przetwarzanie sygnałów wielowymiarowych,
5. Projektowanie systemów internetowych i mobilnych,
6. Projektowanie telemedycznych systemów internetowych i mobilnych,
7. Aplikacje mobilne.

Angażuję się również w opracowywanie materiałów dydaktycznych dla studentów, uczniów oraz nauczycieli, promocję nauki przez studenckie warsztaty naukowe i współpracę z kołami naukowymi oraz promocję uczelni przez organizację *hackathonów*:

- Przygotowałem materiały multimedialne do nauki sztucznej inteligencji oraz serię materiałów wideo dla projektu *Centrum Mistrzostwa Informatycznego*[25].
- Współorganizowałem kilka edycji studenckich warsztatów naukowych *International Students Workshop*[26] (2015-2019).
- Opiekuję się pracami badawczymi studentów w ramach działającego przy *Katedrze Systemów i Sieci Komputerowych*, *Koła Naukowego Systemów i Sieci Komputerowych* oraz założonego w zeszłym roku *Koła Uczenia Maszyn*.
- Byłem pomysłodawcą i jednym z głównych organizatorów *hackathonu JellyPizzaHack*[27], zorganizowanego we współpracy z *Credit Suisse* (16.12.2016).

[25] *Centrum Mistrzostwa Informatycznego* https://cmi.edu.pl

[26] *International Students Workshop* http://sisk.kssk.pwr.edu.pl/isw/

[27] *JellyPizzaHack* https://invest-in-wroclaw.pl/piwnica-dynamicznie-alokowana

Aktywnie uczestniczę także w opracowywaniu kart przedmiotów, będąc opiekunem czterech kursów:

- Projektowanie systemów informatyki medycznej,
- Przetwarzanie sygnałów wielowymiarowych,

- Metody przetwarzania języka naturalnego oraz wyszukiwanie,
- Uczenie Maszyn.

Ponadto, od pięciu lat pełnię rolę sekretarza komisji dyplomowej specjalności *Advanced Informatics and Control*, prowadzonej w języku angielskim. W tym czasie byłem także promotorem 43 prac magisterskich i 37 prac inżynierskich. Sześcioro z moich dyplomantów aktualnie realizuje swoje prace doktorskie na *Politechnice Wrocławskiej*, a z pięciorgiem z nich miałem przyjemność pracować przy ich pierwszych publikacjach konferencyjnych lub publikacjach w czasopismach. Poniżej prezentuję wybraną tematykę prowadzonych prac magisterskich na przykładzie obecnych doktorantów, odnosząc się także do wspólnych prac badawczych:

- Prior Probability Estimation in Dynamically Imbalanced Data Streams, 2022, [C3]
  mgr inż. Joanna Komorniczak

[C3] Joanna Komorniczak, Pawel Zyblewski i Pawel Ksieniewicz. "Prior Probability Estimation in Dynamically Imbalanced Data Streams". W: *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, lip. 2021. DOI: 10.1109/ijcnn52387.2021.9533795

- Feature extraction using n-gram methods for the purpose of ensemble classification of disinformation sources, 2021, [Bor22a]
  mgr inż. Weronika Borek-Marciniec – pełnię rolę promotora pomocniczego

[Bor22a] Weronika Borek-Marciniec i Paweł Ksieniewicz. "Inductive Parallel Learning for Multiple Classification Problems". W: *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022

- Wykorzystanie uczenia zespołowego w klasyfikacji binarnej niezbalansowanych strumieni danych, 2020, [Weg20]
  mgr inż. Weronika Węgier

[Weg20] Weronika Wegier i Pawel Ksieniewicz. "Application of Imbalanced Data Classification Quality Metrics as Weighting Methods of the Ensemble Data Stream Classification Algorithms". W: *Entropy* 22.8 (lip. 2020), s. 849. DOI: 10.3390/e22080849

- Analiza efektywności zastosowania sieci rekurencyjnych w zadaniu klasyfikacji, 2019, [Koz19]
  mgr inż. Jędrzej Kozal,

[Koz19] Jedrzej Kozal i Paweł Ksieniewicz. "Imbalance Reduction Techniques Applied to ECG Classification Problem". W: *Intelligent Data Engineering and Automated Learning – IDEAL 2019*. Springer International Publishing, 2019, s. 323–331

- Analiza efektywności odmian algorytmu SMOTE w balansowaniu strumieni danych, 2019, [Gul19]
  mgr inż. Bogdan Gulowaty,

[Gul19] Bogdan Gulowaty i Paweł Ksieniewicz. "SMOTE Algorithm Variations in Balancing Data Streams". W: *Intelligent Data Engineering and Automated Learning – IDEAL 2019*. Springer International Publishing, 2019, s. 305–312

- Zadanie uczenia nienadzorowanego w kontekście obrazowania nadwidmowego, 2018, [Sul21]
  mgr inż. Dominika Sułot

[Sul21] Sułot D, P. Zyblewski i P. Ksieniewicz. "Analysis of Variance Application in the Construction of Classifier Ensemble Based on Optimal Feature Subset for the Task of Supporting Glaucoma Diagnosis". W: *Computational Science – ICCS 2021*. Springer International Publishing. DOI: 10.1007/978-3-030-77967-2_10

Aktywnie udzielam się również w organizacji konferencji i warsztatów naukowych poświęconych zagadnieniom klasyfikacj danych trudnych, wśród których chciałbym wymienić:

- Organizację sesji specjalnej "*Classifier Learning from Difficult Data*" na konferencji *International Conference on Computational Science* (ICCS) 16–18 czerwca 2021, Kraków, Polska. Zasięg międzynarodowy. https://www.iccs-meeting.org/iccs2021/
- Organizację sesji specjalnej "*Classifier Learning from Difficult Data*" na konferencji *International Conference on Computational Science* (ICCS) 3–5 czerwca 2020, Amsterdam, Holandia. Zasięg międzynarodowy. https://www.iccs-meeting.org/iccs2020/
- Organizację sesji specjalnej "*Machine Learning Algorithms for Hard Problems*" na konferencji *20th International Conference on Intelligent Data Engineering and Automated Learning* (IDEAL) 14–16 listopada 2019, Manchester, Anglia. Zasięg międzynarodowy. http://www.confercare.manchester.ac.uk/events/ideal2019/sessions/
- Organizację sesji specjalnej "*Classifier Learning from Difficult Data*" na konferencji *International Conference on Computational Science* (ICCS) 12–14 czerwca 2019, Faro, Portugalia. Zasięg międzynarodowy. https://www.iccs-meeting.org/iccs2019/
- Organizację konferencji *Polskie Porozumienie na rzecz Rozwoju Sztucznej Inteligencji*, 16–18 października 2019, Wrocław, Polska. Zasięg krajowy. http://pp-rai.pwr.edu.pl/
- Organizację konferencji *The 9 International Conference on Computer Recognition Systems* CORES, Wrocław, Polska. Zasięg międzynarodowy http://cores.pwr.wroc.pl

Wielokrotnie udzielałem się także w działaniach mających na celu zwiększenie świadomości społecznej z zakresu ryzyka związanego z dezinformacją i potencjału metod sztucznej inteligencji w walce z nią. Należy tu wymienić:

- uczestnictwo w roli eksperta w debacie "*Szczepionka na kłamstwo.*" organizowanej przez *Fundację na rzecz Nauki Polskiej* w ramach cyklu „*Ufajmy nauce*" (21.04.2022r.),

  https://www.fnp.org.pl/debata-ekspertow-szczepionka-na-klamstwo

  https://www.youtube.com/watch?v=O75LsK0M4D4

- wygłoszenie prelekcji pod tytułem

  Wykorzystanie sztucznej inteligencji w walce z dezinformacją

  w ramach seminarium "*Sztuczna inteligencja w rozwoju miast i obszarów metropolitarnych*organizowanego przez *Wrocławskie Centrum Akademickie* pod patronatem *World Urban Forum* (9.03.2022r.),

  https://metropolie.pl/artykul/seminarium-sztuczna-inteligencja-dla-rozwoju-miast-i-obszarow-metropolitalnych

  https://vimeo.com/685793217

- wykład w ramach seminarium "*Machine Learning to Combat Fake News and Media Manipulation*" organizowanego przez Elsevier w ramach cyklu webinarów (20.04.2022r.).

  `https://www.workcast.com/register?cpak=7948916184707381`

- wywiady radiowe dotyczące detekcji źródeł dezinformacji:

  Radio RAM
  Politechnika Wrocławska pracuje nad systemem wykrywającym fake newsy

  `https://www.radiowroclaw.pl/articles/view/110758/Politechnika-Wroclawska-pracuje-nad-systemem-wykrywajacym-fake-newsy`

W ostatnich latach przeprowadziłem również trzy wykłady na zaproszenie, dotyczące tematyki klasyfikacji danych trudnych oraz detekcji źródeł dezinformacji:

- Keynote podczas sesji specjalnej CLDD w ramach konferencji International Conference on Computational Science.

  Chosen Challenges of Imbalanced Data Stream Classification
  16 czerwca 2021
  `https://www.iccs-meeting.org/iccs2021/`

- Wykład na zaproszenie:

  Research practices in data stream analysis and imbalanced data classification.
  22 czerwca 2020, Amity School of Engineering and Technology, Noida, Indie,

- Wykład w ramach Elsevier Webinar Machine Learning to combat Fake News and Media Manipulation.

  Using machine learning as the weapon against the disinformation
  20 kwietnia 2021
  `https://www.workcast.com/register?cpak=7948916184707381`

8 Dane bibliometryczne

8.1 Prace autorstwa Pawła Ksieniewicza spoza cyklu

[Ksi22a] P. Ksieniewicz i P. Zyblewski. "Stream-learn — open-source Python library for difficult data stream batch analysis". W: *Neurocomputing* 478 (2022), s. 11–21. DOI: `https://doi.org/10.1016/j.neucom.2021.10.120`

[Ksi22y] Michał Choraś i in. "SocialTruth - content verification for the digital society". W: *Proceedings of the Basque Conference on Cyber-Physical Systems and Artificial Intelligence*. 2022. DOI: `10.5281/zenodo.6562355`

[Ksi22z] Pawel Ksieniewicz i in. “SWAROG – fake news classification
for the local context”. W: *Proceedings of the Basque Conference on Cyber-Physical Systems and Artificial Intelligence*. 2022. DOI: 10.5281/zenodo.6562355

[Bor22a] Weronika Borek-Marciniec i Paweł Ksieniewicz. “Inductive Parallel Learning for Multiple Classification Problems”. W: *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022

[Gos22] Róża Goścień i Paweł Ksieniewicz. “Efficient dynamic routing in Spectrally-Spatially Flexible Optical Networks based on traffic categorization and supervised learning methods”. W: *Optical Switching and Networking* 43 (lut. 2022), s. 100650. DOI: 10.1016/j.osn.2021.100650

[Kom22] Joanna Komorniczak i Paweł Ksieniewicz. “Imbalanced Data Stream Classification Assisted by Prior Probability Estimation”. W: *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022

[Kom22c] Joanna Komorniczak, Paweł Ksieniewicz i Michał Woźniak. “Data complexity and classification accuracy correlation in oversampling algorithms”. W: *Proceedings of the Sixth International Workshop on Learning with Imbalanced Domains: Theory and Applications*. Red. Luís Torgo i in. Proceedings of Machine Learning Research. PMLR, 2022

[Kom22b] Joanna Komorniczak i Paweł Ksieniewicz. “Data stream generation through real concept’s interpolation”. W: *ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. 2022

[Kom22p] Joanna Komorniczak i Pawel Ksieniewicz. *problexity – an open-source Python library for binary classification problem complexity assessment*. 2022. DOI: 10.48550/ARXIV.2207.06709

[Woj22] Szymon Wojciechowski i in. “Hybrid Regression Model for Link Dimensioning in Spectrally-Spatially Flexible Optical Networks”. W: *IEEE Access* 10 (2022), s. 53810–53821. DOI: 10.1109/ACCESS.2022.3175193

[Ksi21k] Paweł Ksieniewicz, Paweł Zyblewski i Robert Burduk. “Fusion of linear base classifiers in geometric space”. W: *Knowledge-Based Systems* 227 (wrz. 2021). DOI: 10.1016/j.knosys.2021.107231

[Cho21] Michał Choraś i in. “Advanced Machine Learning techniques for fake news (online disinformation) detection: A sys-

tematic mapping study". W: *Applied Soft Computing* 101 (2021), s. 107050. DOI: 10.1016/j.asoc.2020.107050

[Sta21] Katarzyna Stapor i in. "How to design the fair experimental classifier evaluation". W: *Applied Soft Computing* 104 (2021). DOI: https://doi.org/10.1016/j.asoc.2021.107219

[Sul21] Sułot D, P. Zyblewski i P. Ksieniewicz. "Analysis of Variance Application in the Construction of Classifier Ensemble Based on Optimal Feature Subset for the Task of Supporting Glaucoma Diagnosis". W: *Computational Science – ICCS 2021*. Springer International Publishing. DOI: 10.1007/978-3-030-77967-2_10

[Sul21b] Dominika Sułot i in. "Glaucoma classification based on scanning laser ophthalmoscopic images using a deep learning ensemble method". W: *PLOS ONE* 16.6 (czer. 2021), s. 1–12. DOI: 10.1371/journal.pone.0252339

[Ksi20s] Paweł Ksieniewicz i in. "Pattern Recognition Model to Aid the Optimization of Dynamic Spectrally-Spatially Flexible Optical Networks". W: *Lecture Notes in Computer Science*. Springer International Publishing, 2020, s. 211–224

[Ksi20b] Paweł Ksieniewicz i Robert Burduk. "Clustering and Weighted Scoring in Geometric Space Support Vector Machine Ensemble for Highly Imbalanced Data Classification". W: *Lecture Notes in Computer Science*. Springer International Publishing, 2020, s. 128–140

[Ksi20e] Pawel Ksieniewicz. "Standard Decision Boundary in a Support-Domain of Fuzzy Classifier Prediction for the Task of Imbalanced Data Classification". W: *Lecture Notes in Computer Science*. Springer International Publishing, 2020, s. 103–116

[Klin20] Miroslaw Klinkowski i in. "Machine Learning Assisted Optimization of Dynamic Crosstalk-Aware Spectrally-Spatially Flexible Optical Networks". W: *Journal of Lightwave Technology* 38.7 (kw. 2020), s. 1625–1635. DOI: 10.1109/JLT.2020.2967087

[Kul20] Sebastian Kula i in. "Sentiment Analysis for Fake News Detection by Means of Neural Networks". W: *Lecture Notes in Computer Science*. Springer International Publishing, 2020, s. 653–666

[Weg20] Weronika Wegier i Pawel Ksieniewicz. "Application of Imbalanced Data Classification Quality Metrics as Weighting Methods of the Ensemble Data Stream Classification Algorithms". W: *Entropy* 22.8 (lip. 2020), s. 849. DOI: 10.3390/e22080849

[Zyb20a] Paweł Zyblewski, Paweł Ksieniewicz i Michał Woźniak. "Combination of Active and Random Labeling Strategy in the Non-stationary Data Stream Classification". W: *Artificial Intelligence and Soft Computing*. Springer International Publishing, 2020, s. 576–585

[Ksi19f] Paweł Ksieniewicz i in. "Machine Learning Methods for Fake News Classification". W: *Intelligent Data Engineering and Automated Learning – IDEAL 2019*. Springer International Publishing, 2019, s. 332–339

[Gul19] Bogdan Gulowaty i Paweł Ksieniewicz. "SMOTE Algorithm Variations in Balancing Data Streams". W: *Intelligent Data Engineering and Automated Learning – IDEAL 2019*. Springer International Publishing, 2019, s. 305–312

[Kli19] Jakub Klikowski, Paweł Ksieniewicz i Michał Woźniak. "A Genetic-Based Ensemble Learning Applied to Imbalanced Data Classification". W: *Intelligent Data Engineering and Automated Learning – IDEAL 2019*. Springer International Publishing, 2019, s. 340–352

[Koz19] Jedrzej Kozal i Paweł Ksieniewicz. "Imbalance Reduction Techniques Applied to ECG Classification Problem". W: *Intelligent Data Engineering and Automated Learning – IDEAL 2019*. Springer International Publishing, 2019, s. 323–331

[Zyb19] P. Zyblewski, P. Ksieniewicz i M. Woźniak. "Classifier Selection for Highly Imbalanced Data Streams with Minority Driven Ensemble". W: *Artificial Intelligence and Soft Computing*. Springer International Publishing, 2019, s. 626–635

[Ksi18e] Paweł Ksieniewicz. "Entropodynamiczny filtr percentylowy". W: *Edukacja – Technika – Informatyka*. Wydawnictwo Uniwersytetu Rzeszowskiego, 2018

[Ksi18c] Paweł Ksieniewicz. "Combined Classifier Based on Quantized Subspace Class Distribution". W: *Intelligent Data Engineering and Automated Learning – IDEAL 2018*. Springer International Publishing, 2018, s. 761–772

[Lap18] Andrzej Lapinski i in. "An Empirical Insight Into Concept Drift Detectors Ensemble Strategies". W: *2018 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, lip. 2018

[Ksi17p] Paweł Ksieniewicz, Manuel Grana i Michał Woźniak. "Paired feature multilayer ensemble–concept and evaluation of a classifier". W: *Journal of Intelligent & Fuzzy Systems* 32.2 (2017), s. 1427–1436. DOI: 10.3233/JIFS-169139

8.2 *Prace autorstwa Pawła Ksieniewicza przed uzyskaniem stopnia naukowego doktora inżyniera*

[Ksi16a] Pawel Ksieniewicz i Michał Woźniak. "Artificial Photoreceptors for Ensemble Classification of Hyperspectral Images". W: *Advances in Intelligent Systems and Computing*. Springer International Publishing, 2016, s. 471–479. DOI: 10.1007/978-3-319-26227-7_44

[Ksi16b] Paweł Ksieniewicz, Bartosz Krawczyk i Michał Woźniak. "Ensemble of One-Dimensional Classifiers for Hyperspectral Image Analysis". W: *Data Mining and Big Data*. Springer International Publishing, 2016, s. 513–520. DOI: 10.1007/978-3-319-40973-3_52

[Ksi16m] Pawel Ksieniewicz i Michal Wozniak. "Imbalance medical data classification using Exposer Classifier Ensemble". W: *4th Workshop on Machine Learning in Life Sciences (MLLS), 23 September 2016, Riva del Garda, Italy, 23 September 2016 : proceedings*. 2016

[Woz16] Michał Woźniak i in. "A First Attempt to Construct Effective Concept Drift Detector Ensembles". W: *Advances in Intelligent Systems and Computing*. Springer International Publishing, paź. 2016, s. 27–34. DOI: 10.1007/978-3-319-47274-4_3

[Woz16b] Michał Woźniak i in. "Active Learning Classification of Drifted Streaming Data". W: *Procedia Computer Science* 80 (2016), s. 1724–1733. DOI: 10.1016/j.procs.2016.05.514

[Woz16c] Michał Woźniak i in. "Active Learning Classifier for Streaming Data". W: *Lecture Notes in Computer Science*. Springer International Publishing, 2016, s. 186–197. DOI: 10.1007/978-3-319-32034-2_16

[Woz16d] Michał Woźniak i in. "Ensembles of Heterogeneous Concept Drift Detectors - Experimental Study". W: *Computer Information Systems and Industrial Management*. Springer International Publishing, 2016, s. 538–549. DOI: 10.1007/978-3-319-45378-1_48

[Ksi15a] Paweł Ksieniewicz, Manuel Graña i Michał Woźniak. "Blurred Labeling Segmentation Algorithm for Hyperspectral Images". W: *Computational Collective Intelligence*. Springer International Publishing, 2015, s. 578–587. DOI: 10.1007/978-3-319-24306-1_56

[Ksi14a] P. Ksieniewicz i in. "A novel hyperspectral segmentation algorithm–concept and evaluation". W: *Logic Journal of IGPL* 23.1 (grud. 2014), s. 105–120. DOI: 10.1093/jigpal/jzu045

[Jac14] Konrad Jackowski i in. "Ensemble Classifier Systems for Headache Diagnosis". W: *Advances in Intelligent Systems and Computing*. Springer International Publishing, 2014, s. 273–284. DOI: 10.1007/978-3-319-06596-0_25

[Kra14a] Bartosz Krawczyk, Paweł Ksieniewicz i Michał Woźniak. "Hyperspectral Image Analysis Based on Color Channels and Ensemble Classifier". W: *Lecture Notes in Computer Science*. Springer International Publishing, 2014, s. 274–284. DOI: 10.1007/978-3-319-07617-1_25

[Kra14b] Bartosz Krawczyk, Paweł Ksieniewicz i Michał Woźniak. "Hyperspectral Image Analysis Based on Quad Tree Decomposition". W: *Advances in Intelligent Systems and Computing*. Springer International Publishing, 2014, s. 105–113. DOI: 10.1007/978-3-319-07995-0_11

.....
(podpis wnioskodawcy)

ZALACZNIK 4

Wykaz osiągnięć naukowych albo artystycznych, stanowiących znaczny wkład w rozwój określonej dyscypliny

1 INFORMACJA O OSIĄGNIĘCIACH NAUKOWYCH ALBO ARTYSTYCZNYCH O KTÓRYCH MOWA W ART. 219 UST. 1. PKT 2 USTAWY

1.1 Monografia naukowa

—

1.2 Cykl powiązanych tematycznie artykułów naukowych, zgodnie z art. 219 ust. 1. pkt 2b Ustawy; pt:

Projektowanie algorytmów rozpoznawania wzorców dla zadania klasyfikacji trudnych danych

Wszystkie publikacje pochodzą z okresu po uzyskaniu stopnia doktora. Informacje dot. liczby punktów MEiN, współczynnika IF oraz liczby cytowań oddają **stan na dzień 27. sierpnia 2022 r.** zgodnie z bazami publikacji naukowych:

WoS *Web of Science* `https://www.webofscience.com/wos/author/record/1886494`

Sco *Scopus* `https://www.scopus.com/authid/detail.uri?authorId=56206176100`

GSc *Google Scholar* `https://scholar.google.com/citations?user=YSM30D8AAAAJ`

[C1] Joanna Komorniczak, Paweł Zyblewski i Paweł Ksieniewicz. "Statistical Drift Detection Ensemble for batch processing of data streams". W: *Knowledge-Based Systems* 252 (2022), s. 109380. DOI: 10.1016/j.knosys.2022.109380

CRediT: **Conceptualization** **Software** **Validation** **Investigation** **Writing - Original Draft** **Writing - Review & Editing** **Visualization** **Supervision**

	WoS	Sco	GSc
l. cytowań	—	—	—

Szacowany udział	70%
Impact Factor	8.139
l. punktów MEiN	200

[C2] Pawel Ksieniewicz. "Processing data stream with chunk-similarity model selection". W: *Applied Intelligence* (lip. 2022). DOI: 10.1007/s10489-022-03826-4

CRediT: **Conceptualization** **Methodology** **Software** **Validation** **Formal Analysis** **Investigation** **Resources** **Data Curation** **Writing - Original Draft** **Writing - Review & Editing** **Visualization**

	WoS	Sco	GSc
l. cytowań	—	—	—

Szacowany udział	100%
Impact Factor	5.086
l. punktów MEiN	70

[C3] Joanna Komorniczak, Pawel Zyblewski i Pawel Ksieniewicz. "Prior Probability Estimation in Dynamically Imbalanced Data Streams". W: *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, lip. 2021. DOI: 10 . 1109 / ijcnn52387 . 2021.9533795

CRediT: **Conceptualization** **Validation** **Formal Analysis** **Investigation** **Resources** **Writing - Original Draft** **Writing - Review & Editing** **Visualization** **Supervision**

	WoS	Sco	GSc
l. cytowań	2	1	4

Szacowany udział	70%
Core	B
l. punktów MEiN	140

[C4] Paweł Ksieniewicz. "The prior probability in the batch classification of imbalanced data streams". W: *Neurocomputing* 452 (wrz. 2021), s. 309–316. DOI: 10.1016/j.neucom.2019.11.126

CRediT: **Conceptualization** **Methodology** **Software** **Validation** **Formal Analysis** **Investigation** **Resources** **Data Curation** **Writing - Original Draft** **Writing - Review & Editing** **Visualization**

	WoS	Sco	GSc
l. cytowań	2	2	4

Szacowany udział	100%
Impact Factor	5.719
l. punktów MEiN	140

[C5] Paweł Ksieniewicz, Paweł Zyblewski, Michał Choraś, Rafał Kozik, Agata Giełczyk, Michał Woźniak, *"Fake News Detection from Data Streams"*. W: *2020 International Joint Conference on Neural Networks (IJCNN)*. 2020, s. 1-8. DOI: 10.1109/IJCNN48605.2020.9207498

CRediT: **Conceptualization** **Methodology** **Software** **Validation** **Investigation** **Writing - Original Draft** **Writing - Review & Editing** **Visualization**

	WoS	Sco	GSc
l. cytowań	6	7	13

Szacowany udział	50%
Core	A
l. punktów MEiN	140

[C6] Pawel Ksieniewicz. "Combining Random Subspace Approach with smote Oversampling for Imbalanced Data Classification". W: *Hybrid Artificial Intelligent Systems*. Red. Hilde Pérez García i in. Cham: Springer International Publishing, 2019, s. 660–673. ISBN: 978-3-030-29859-3. DOI: 10.1007/978-3-030-29859-3_56

CRediT: **Conceptualization** **Methodology** **Software** **Validation** **Formal Analysis** **Investigation** **Resources** **Data Curation** **Writing - Original Draft** **Writing - Review & Editing** **Visualization**

	WoS	Sco	GSc
l. cytowań	4	5	5

Szacowany udział 100%
Core C
l. punktów MEiN 20

[C7] Paweł Ksieniewicz, Michał Woźniak, Bogusław Cyganek, Andrzej Kasprzak i Krzysztof Walkowiak. *"Data stream classification using active learned neural networks"*. W: *Neurocomputing 353 (2019)*, s. 74-82. DOI: 10.1016/j.neucom.2018.05.130

CRediT: **Methodology** **Software** **Validation** **Investigation** **Writing - Original Draft** **Writing - Review & Editing** **Visualization**

	WoS	Sco	GSc
l. cytowań	11	18	28

Szacowany udział 50%
Impact Factor 4.438
l. punktów MEiN 140

[C8] Pawel Ksieniewicz. "Undersampled Majority Class Ensemble for highly imbalanced binary classification". W: *Proceedings of the Second International Workshop on Learning with Imbalanced Domains: Theory and Applications*. Red. Luís Torgo i in. T. 94. Proceedings of Machine Learning Research. PMLR, paź. 2018, s. 82–94. URL: https://proceedings.mlr.press/v94/ksieniewicz18a.html

CRediT: **Conceptualization** **Methodology** **Software** **Validation** **Formal Analysis** **Investigation** **Resources** **Data Curation** **Writing - Original Draft** **Writing - Review & Editing** **Visualization**

	WoS	Sco	GSc
l. cytowań	—	—	10

Szacowany udział 100%
Core A
l. punktów MEiN 140

[C9] Paweł Ksieniewicz i Michał Woźniak. "Imbalanced Data Classification Based on Feature Selection Techniques". W: *Intelligent Data Engineering and Automated Learning – IDEAL 2018*. Red. Hujun Yin i in. Cham: Springer International Publishing, 2018, s. 296–303. DOI: 10.1007/978-3-030-03496-2_33

CRediT: **Methodology** **Software** **Validation** **Investigation** **Writing - Original Draft** **Writing - Review & Editing** **Visualization**

	WoS	Sco	GSc
l. cytowań	6	11	13

Szacowany udział 80%
Core B
l. punktów MEiN 15

[C10] Paweł Ksieniewicz, Bartosz Krawczyk i Michał Woźniak. "Ensemble of Extreme Learning Machines with trained classifier combination and statistical features for hyperspectral data". W: *Neurocomputing* 271 (2018), s. 28–37. DOI: 10.1016/j.neucom.2016.04.076

CRediT: **Conceptualization** **Methodology** **Software** **Validation** **Writing - Original Draft** **Writing - Review & Editing** **Visualization**

	WoS	Sco	GSc
l. cytowań	15	15	18

Szacowany udział 60%
Impact Factor 4.072
l. punktów MEiN 30

[C11] Paweł Ksieniewicz i Michał Woźniak. "Dealing with the task of imbalanced, multidimensional data classification using ensembles of exposers". W: *Proceedings of the First International Workshop on Learning with Imbalanced Domains: Theory and Applications*. Red. Paula Branco Luís Torgo i Nuno Moniz. T. 74. Proceedings of Machine Learning Research. PMLR, 22 Sep 2017, s. 164–175. URL: https://proceedings.mlr.press/v74/ksieniewicz17a.html

CRediT: **Conceptualization** **Methodology** **Software** **Validation** **Formal Analysis** **Investigation** **Resources** **Data Curation** **Writing - Original Draft** **Writing - Review & Editing** **Visualization**

	WoS	Sco	GSc
l. cytowań	—	—	9

Szacowany udział 80%
Core A
l. punktów MEiN 140

2 *INFORMACJA O AKTYWNOŚCI NAUKOWEJ ALBO ARTYSTYCZNEJ*

2.1 Wykaz opublikowanych monografii naukowych (z zaznaczeniem pozycji niewymienionych w pkt 1.1).

- **Paweł Ksieniewicz [Red.]**, Mariusz Uchroński [Red.]
  Selected model based architectures and algorithms for learning, signal processing and optimization
  Warszawa: Akademicka Oficyna Wydawnicza EXIT, cop. 2021. 144 s. (Problemy Współczesnej Informatyki)

2.2 Wykaz opublikowanych rozdziałów w monografiach naukowych.

Jestem autorem dwóch rozdziałów w książkach:

- D. Sułot, P. Zyblewski, **P. Ksieniewicz**
  A novel approach to learning and designing neural networks-based ensemble
  Selected model based architectures and algorithms for learning, signal processing and optimization / red. Paweł Ksieniewicz, Mariusz Uchroński. Warszawa : Akademicka Oficyna Wydawnicza EXIT, cop. 2021. s. 103-112.

- K. Jackowski, D. Jankowski, **P. Ksieniewicz**, D. Simić, S. Simić, M. Woźniak.
  Ensemble classifier systems for headache diagnosis.
  W: Information technologies in biomedicine. Vol. 4 / Ewa Pietka, Jacek Kawa, Wojciech Wieclawek (eds.). Cham [i in.] : Springer, cop. 2014. s. 273-284. (Advances in Intelligent Systems and Computing, ISSN 2194-5357; vol. 284)

2.3 Informacja o członkowskie w redakcjach naukowych monografii

Jestem współredaktorem oraz współautorem rozdziału w książce:

- **Paweł Ksieniewicz [Red.]**, Mariusz Uchroński [Red.]
  Selected model based architectures and algorithms for learning, signal processing and optimization
  Warszawa: Akademicka Oficyna Wydawnicza EXIT, cop. 2021. 144 s. (Problemy Współczesnej Informatyki)

2.4 Wykaz opublikowanych artykułów w czasopismach naukowych (z zaznaczeniem pozycji niewymienionych w pkt I.2)

Informacje dot. liczby punktów MEiN oraz współczynnika IF podane na podstawie wskaźników z dnia 27 sierpnia 2022.

2.4a Artykuły opublikowane po uzyskaniu stopnia doktora, zgłoszone w punkcie I.2:

1.) **[IF: 8.139, PKT: 200]** [C1]
Joanna Komorniczak, Paweł Zyblewski i Paweł Ksieniewicz. "Statistical Drift Detection Ensemble for batch processing of data streams". W: *Knowledge-Based Systems* 252 (2022), s. 109380. DOI: 10.1016/j.knosys.2022.109380

2.) **[IF:5.019, PKT: 70]** [C2]
Pawel Ksieniewicz. "Processing data stream with chunk-similarity model selection". W: *Applied Intelligence* (lip. 2022). DOI: 10.1007/s10489-022-03826-4

3.) **[IF:5.719, PKT: 140]** [C4]
Paweł Ksieniewicz. "The prior probability in the batch classification of imbalanced data streams". W: *Neurocomputing* 452 (wrz. 2021), s. 309–316. DOI: 10.1016/j.neucom.2019.11.126

4.) **[IF: 4.438, PKT:140]** [C7]
Paweł Ksieniewicz i in. "Data stream classification using active learned neural networks". W: *Neurocomputing* 353 (2019), s. 74–82. DOI: 10.1016/j.neucom.2018.05.130

5.) **[IF: 4.072, PKT: 30]** [C10]
Paweł Ksieniewicz, Bartosz Krawczyk i Michał Woźniak. "Ensemble of Extreme Learning Machines with trained classifier combination and statistical features for hyperspectral data". W: *Neurocomputing* 271 (2018), s. 28–37. DOI: 10.1016/j.neucom.2016.04.076

2.4b Artykuły opublikowane po uzyskaniu stopnia doktora, niezgłoszone w punkcie I.2.:

1.) **[IF: 5.779, PKT: 140]** [Ksi22a]
P. Ksieniewicz i P. Zyblewski. "Stream-learn — open-source Python library for difficult data stream batch analysis". W: *Neurocomputing* 478 (2022), s. 11–21. DOI: https://doi.org/10.1016/j.neucom.2021.10.120

2.) **[IF: 8.263, PKT: 200]** [Cho21]
Michał Choraś i in. "Advanced Machine Learning techniques for fake news (online disinformation) detection: A systematic mapping study". W: *Applied Soft Computing* 101 (2021), s. 107050. DOI: 10.1016/j.asoc.2020.107050

3.) **[IF: 2.738, PKT: 100]** [Weg20]
Weronika Wegier i Pawel Ksieniewicz. "Application of Imbalanced Data Classification Quality Metrics as Weighting Methods of the Ensemble Data Stream Classification Algorithms". W: *Entropy* 22.8 (lip. 2020), s. 849. DOI: 10.3390/e22080849

4.) **[IF: 8.139, PKT: 200]** [Ksi21k]
Paweł Ksieniewicz, Paweł Zyblewski i Robert Burduk. "Fusion of linear base classifiers in geometric space". W: *Knowledge-Based Systems* 227 (wrz. 2021). DOI: 10.1016/j.knosys.2021.107231

5.) **[IF: 3.752, PKT: 100]** [Sul21b]
Dominika Sułot i in. "Glaucoma classification based on scanning laser ophthalmoscopic images using a deep learning ensemble method". W: *PLOS ONE* 16.6 (czer. 2021), s. 1–12. DOI: 10.1371/journal.pone.0252339

6.) **[IF: 8.263, PKT: 200]** [Sta21]
Katarzyna Stapor i in. "How to design the fair experimental classifier evaluation". W: *Applied Soft Computing* 104 (2021). DOI: https://doi.org/10.1016/j.asoc.2021.107219

7.) **[IF: 3.476, PKT: 100]** [Woj22]
Szymon Wojciechowski i in. "Hybrid Regression Model for Link Dimensioning in Spectrally-Spatially Flexible Optical Networks". W: *IEEE Access* 10 (2022), s. 53810–53821. DOI: 10.1109/ACCESS.2022.3175193

8.) **[IF: 2.786, PKT: 40]** [Gos22]
Róża Goścień i Paweł Ksieniewicz. "Efficient dynamic routing in Spectrally-Spatially Flexible Optical Networks based on traffic categorization and supervised learning methods". W: *Optical Switching and Networking* 43 (lut. 2022), s. 100650. DOI: 10.1016/j.osn.2021.100650

9.) **[IF: 4.142, PKT: 140]** [Klin20]
Miroslaw Klinkowski i in. "Machine Learning Assisted Optimization of Dynamic Crosstalk-Aware Spectrally-Spatially Flexible Optical Networks". W: *Journal of Lightwave Technology* 38.7 (kw. 2020), s. 1625–1635. DOI: 10.1109/JLT.2020.2967087

2.4c Artykuły opublikowane przed uzyskaniem stopnia doktora :

1.) [Woz16b]
Michał Woźniak i in. "Active Learning Classification of Drifted Streaming Data". W: *Procedia Computer Science* 80 (2016), s. 1724–1733. DOI: 10.1016/j.procs.2016.05.514

2.) [Ksi17p]
Paweł Ksieniewicz, Manuel Grana i Michał Woźniak. "Paired feature multilayer ensemble–concept and evaluation of a classifier". W: *Journal of Intelligent & Fuzzy Systems* 32.2 (2017), s. 1427–1436. DOI: 10.3233/JIFS-169139

3.) [Ksi14a]
P. Ksieniewicz i in. "A novel hyperspectral segmentation algorithm–concept and evaluation". W: *Logic Journal of IGPL* 23.1 (grud. 2014), s. 105–120. DOI: 10.1093/jigpal/jzu045

2.5 Wykaz osiągnięć projektowych, konstrukcyjnych, technologicznych (z zaznaczeniem pozycji niewymienionych w pkt I.3)

—

2.6 Wykaz publicznych realizacji dzieł artystycznych (z zaznaczeniem pozycji niewymienionych w pkt I.3)

—

2.7 Informacja o wystąpieniach na krajowych lub międzynarodowych konferencjach naukowych lub artystycznych, z wyszczególnieniem przedstawionych wykładów na zaproszenie i wykładów plenarnych.

2.7a Aktywny udział (prezentacja pracy) podczas międzynarodowych konferencji naukowych po uzyskaniu stopnia doktora:

1.)

Konferencja:	2022 International Joint Conference on Neural Networks (IJCNN)
Miejsce:	Padwa, Włochy
Referat:	**Joanna Komorniczak i Paweł Ksieniewicz. "Imbalanced Data Stream Classification Assisted by Prior Probability Estimation". W: *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022**

2.)

Konferencja:	2021 International Joint Conference on Neural Networks (IJCNN)
Miejsce:	Shenzen, Chiny
Referat:	**Joanna Komorniczak, Pawel Zyblewski i Pawel Ksieniewicz. "Prior Probability Estimation in Dynamically Imbalanced Data Streams". W: *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, lip. 2021. doi: 10.1109/ijcnn52387.2021.9533795**

3.)

Konferencja:	ICCS 2020 : 20th International Conference of Computational Science
Miejsce:	Amsterdam, Holandia, 3-5 czerwca 2020
Referat:	**Paweł Ksieniewicz i Robert Burduk. "Clustering and Weighted Scoring in Geometric Space Support Vector Machine Ensemble for Highly Imbalanced Data Classification". W: *Lecture Notes in Computer Science*. Springer International Publishing, 2020, s. 128–140**

4.)

Konferencja:	2020 International Joint Conference on Neural Networks (IJCNN)
Miejsce:	Glasgow, Szkocja, 19-24 czerwca 2020
Referat:	**Paweł Ksieniewicz i in. "Fake News Detection from Data Streams". W: *2020 International Joint Conference on Neural Networks (IJCNN)*. 2020, s. 1–8. doi: 10.1109/IJCNN48605.2020.9207498**

5.)

Konferencja:	Hybrid Artificial Intelligent Systems : 14th International Conference, HAIS 2019
Miejsce:	León, Hiszpania, 4–6 Września 2019
Referat:	**Pawel Ksieniewicz. "Combining Random Subspace Approach with smote Oversampling for Imbalanced Data Classification". W: *Hybrid Artificial Intelligent Systems*. Red. Hilde Pérez García i in. Cham: Springer International Publishing, 2019, s. 660–673. isbn: 978-3-030-29859-3. doi: 10.1007/978-3-030-29859-3_56**

6.)

Konferencja:	Intelligent Data Engineering and Automated Learning - IDEAL 2019 : 20th International Conference
Miejsce:	Manchester, Anglia, 14-16 listopada 2019
Referat:	**Jedrzej Kozal i Paweł Ksieniewicz. "Imbalance Reduction Techniques Applied to ECG Classification Problem". W: *Intelligent Data Engineering and Automated Learning – IDEAL 2019*. Springer International Publishing, 2019, s. 323–331**

7.)

Konferencja:	Intelligent Data Engineering and Automated Learning - IDEAL 2019 : 20th International Conference
Miejsce:	Manchester, Anglia, 14-16 listopada 2019
Referat:	**Paweł Ksieniewicz i in. "Machine Learning Methods for Fake News Classification". W: *Intelligent Data Engineering and Automated Learning – IDEAL 2019*. Springer International Publishing, 2019, s. 332–339**

8.)

Konferencja:	Intelligent Data Engineering and Automated Learning - IDEAL 2018 : 19th International Conference
Miejsce:	Madryt, Hiszpania, 21-23 listopada 2018
Referat:	**Paweł Ksieniewicz i Michał Woźniak. "Imbalanced Data Classification Based on Feature Selection Techniques". W: *Intelligent Data Engineering and Automated Learning – IDEAL 2018*. Red. Hujun Yin i in. Cham: Springer International Publishing, 2018, s. 296–303. doi: 10.1007/978-3-030-03496-2_33**

9.)

Konferencja:	Intelligent Data Engineering and Automated Learning - IDEAL 2018 : 19th International Conference
Miejsce:	Madryt, Hiszpania, 21-23 listopada 2018
Referat:	**Paweł Ksieniewicz. "Combined Classifier Based on Quantized Subspace Class Distribution". W: *Intelligent Data Engineering and Automated Learning – IDEAL 2018*. Springer International Publishing, 2018, s. 761–772**

10.)

Konferencja:	Second International Workshop on Learning with Imbalanced Domains: Theory and Applications,
Miejsce:	Dublin, Irlandia, 10 września 2018
Referat:	**Pawel Ksieniewicz. "Undersampled Majority Class Ensemble for highly imbalanced binary classification". W: *Proceedings of the Second International Workshop on Learning with Imbalanced Domains: Theory and Applications*. Red. Luís Torgo i in. T. 94. Proceedings of Machine Learning Research. PMLR, paź. 2018, s. 82–94. url: https://proceedings.mlr.press/v94/ksieniewicz18a.html**

11.)

Konferencja:	First International Workshop on Learning with Imbalanced Domains: Theory and Applications,
Miejsce:	Skopje, Macedonia, 22 września 2017
Referat:	**Paweł Ksieniewicz i Michał Woźniak. "Dealing with the task of imbalanced, multidimensional data classification using ensembles of exposers". W: *Proceedings of the First International Workshop on Learning with Imbalanced Domains: Theory and Applications*. Red. Paula Branco Luís Torgo i Nuno Moniz. T. 74. Proceedings of Machine Learning Research. PMLR, 22 Sep 2017, s. 164–175. url: https://proceedings.mlr.press/v74/ksieniewicz17a.html**

2.7b Aktywny udział (prezentacja pracy) podczas międzynarodowych konferencji naukowych przed uzyskaniem stopnia doktora:

1.)

Konferencja:	Image Processing and Communications Challenges 8 : 8th International Conference, IP&C 2016,
Miejsce:	Bydgoszcz, Polska, wrzesień 2016
Referat:	**Michał Woźniak i in. "A First Attempt to Construct Effective Concept Drift Detector Ensembles". W: *Advances in Intelligent Systems and Computing*. Springer International Publishing, paź. 2016, s. 27–34. doi: 10.1007/978-3-319-47274-4_3**

2.)

Konferencja:	Proceedings of the 9th International Conference on Computer Recognition Systems, CORES 2015
Miejsce:	Wrocław, Polska, 2015
Referat:	**Pawel Ksieniewicz i Michał Woźniak. "Artificial Photoreceptors for Ensemble Classification of Hyperspectral Images". W: *Advances in Intelligent Systems and Computing*. Springer International Publishing, 2016, s. 471–479. doi: 10.1007/978-3-319-26227-7_44**

3.)

Konferencja:	4th Workshop on Machine Learning in Life Sciences (MLLS), 23 September 2016,
Miejsce:	Riva del Garda, Włochy, 23 września 2016
Referat:	**Pawel Ksieniewicz i Michal Wozniak. "Imbalance medical data classification using Exposer Classifier Ensemble". W: *4th Workshop on Machine Learning in Life Sciences (MLLS), 23 September 2016, Riva del Garda, Italy, 23 September 2016 : proceedings*. 2016**.

4.)

Konferencja:	Computational Collective Intelligence : 7th International Conference, ICCCI 2015
Miejsce:	Madryt, Hiszpania, 21–23 września 2015
Referat:	**Paweł Ksieniewicz, Manuel Graña i Michał Woźniak. "Blurred Labeling Segmentation Algorithm for Hyperspectral Images". W: *Computational Collective Intelligence*. Springer International Publishing, 2015, s. 578–587. doi: 10.1007/978-3-319-24306-1_56**

5.)

Konferencja:	International Joint Conference SOCO'14-CISIS'14-ICEUTE'14,
Miejsce:	Bilbao, Hiszpania, 25-27 czerwca 2014
Referat:	**Bartosz Krawczyk, Paweł Ksieniewicz i Michał Woźniak. "Hyperspectral Image Analysis Based on Quad Tree Decomposition". W: *Advances in Intelligent Systems and Computing*. Springer International Publishing, 2014, s. 105–113. doi: 10.1007/978-3-319-07995-0_11**

6.)

Konferencja:	Hybrid artificial intelligence systems : 9th international conference, HAIS 2014
Miejsce:	Salamanca, Hiszpania, 11-13 czerwca 2014
Referat:	**Bartosz Krawczyk, Paweł Ksieniewicz i Michał Woźniak. "Hyperspectral Image Analysis Based on Color Channels and Ensemble Classifier". W: *Lecture Notes in Computer Science*. Springer International Publishing, 2014, s. 274–284. doi: 10.1007/978-3-319-07617-1_25**

7.)

Konferencja:	4th International Conference, Information Technologies in Biomedicine
Miejsce:	Kamień Śląski, Polska, 2-4 czerwca 2014
Referat:	**Konrad Jackowski i in. "Ensemble Classifier Systems for Headache Diagnosis". W: *Advances in Intelligent Systems and Computing*. Springer International Publishing, 2014, s. 273–284. doi: 10 . 1007 / 978 - 3 - 319-06596-0_25**

2.7c Wykłady (prezentacje) wygłoszone dla zagranicznych zespołów badawczych:

1.) Wykład na zaproszenie
Research practices in data stream analysis and imbalanced data classification
22 czerwca 2020, Amity School of Engineering and Technology, Noida, Indie

2.) Wykład w ramach Elsevier Webinar Machine Learning to combat Fake News and Media Manipulation,
Using machine learning as the weapon against the disinformation
20 kwietnia 2021
`https://www.workcast.com/register?cpak=7948916184707381`

3.) Keynote podczas sesji specjalnej CLDD w ramach konferencji International Conference on Computational Science,
Chosen Challenges of Imbalanced Data Stream Classification
16 czerwca 2021
`https://www.iccs-meeting.org/iccs2021/`

2.8 Informacja o udziale w komitetach organizacyjnych i naukowych konferencji krajowych lub międzynarodowych, z podaniem pełnionej funkcji.

8a.) Członek komitetu technicznego podczas międzynarodowych konferencji naukowych po uzyskaniu stopnia doktora:

- IEEE International Conference on Omni-layer Intelligent Systems 2022, IEEE COINS 2022
- The 12 International Conference on Computer Recognition Systems, CORES 2021,
- International Conference on Computational Science 2021, ICCS 2021
- 13th International Conference on Computational Intelligence in Security for Information Systems, CISIS 2020,
- International Conference on Computational Science 2020, ICCS 2020
- 11th International Conference on Image Processing and Communications, IP&C 2019
- The 11 International Conference on Computer Recognition Systems, CORES 2019,
- 20th International Conference on Intelligent Data Engineering and Automated Learning, IDEAL 2019

8b.) Członek komitetu technicznego podczas międzynarodowych konferencji naukowych przed uzyskaniem stopnia doktora

- International Conference on Data Mining and Big Data, DMDB 2016
- The 9 International Conference on Computer Recognition Systems, CORES 2015,
- Hybrid Ensemble Machine Learning for Complex and Dynamic Data, HEMLCDD 2014,

8c.) Członek komitetu organizacyjnego specjalnej sesji naukowej po uzyskaniu stopnia doktora

- Organizacja sesji specjalnej "Classifier Learning from Difficult Data" na konferencji International Conference on Computational Science, 3–5 czerwca 2020, Amsterdam, Holandia. Zasięg międzynarodowy.
- Organizacja sesji specjalnej "Machine Learning Algorithms for Hard Problems" na konferencji 20th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL), 14–16 listopada 2019, Manchester, Anglia. Zasięg międzynarodowy.

- Organizacja sesji specjalnej "Classifier Learning from Difficult Data" na konferencji International Conference on Computational Science, 12–14 czerwca 2019, Faro, Portugalia. Zasięg międzynarodowy.
- Organizacja konferencji Polskie Porozumienie na rzecz Rozwoju Sztucznej Inteligencji, 16– 18 października 2019, Wrocław, Polska. Zasięg krajowy.

8d.) Członek komitetu organizacyjnego specjalnej sesji naukowej przed uzyskaniem stopnia doktora

- Organizacja konferencji The 9 International Conference on Computer Recognition Systems, CORES 2015, Wrocław, Polska. Zasięg międzynarodowy

2.9 Informacja o uczestnictwie w pracach zespołów badawczych realizujących projekty finansowane w drodze konkursów krajowych lub zagranicznych, z podziałem na projekty zrealizowane i będące w toku realizacji, oraz z uwzględnieniem informacji o pełnionej funkcji w ramach prac zespołów.

9a.) Projekty w toku, rozpoczęte po uzyskaniu stopnia doktora:

1.)

Tytuł	**System Wykrywania Dezinformacji Metodami Sztucznej Inteligencji**
Źródło finansowania	NCBiR
Budżet	8 657 006 zł
Okres realizacji	2021-12-02 – 2024-04-01
Partnerzy	Matic S.A., Politechnika Bydgoska
Rola w projekcie	**Kierownik B+R**.

2.)

Tytuł	**Incat FaaS AI - Opracowanie platformy bezpieczeństwa operacyjnego podmiotów finansowych w oparciu o zaawansowane mechanizmy uczenia maszynowego**
Źródło finansowania	NCBiR
Budżet	8 267 875 zł
Okres realizacji	2020-04-01 – 2023-03-31
Partnerzy	INCAT Spółka z ograniczoną odpowiedzialnością
Rola w projekcie	Ekspert AI

3.)

Tytuł	**Math Solution innowacyjna platforma wspomagająca uczniów i korepetytorów w procesie nauczania indywidualnego lub wspólnego w oparciu o zaawansowane algorytmy przetwarzania obrazu i uczenia maszynowego w zakresie matematyki i innych przedmiotów ścisłych**
Źródło finansowania	NCBiR
Budżet	9 162 812,50 zł
Okres realizacji	2021-11-01 – 2023-10-01
Partnerzy	Sirius Education Sp. z o.o.
Rola w projekcie	Ekspert AI

5.)

Tytuł	**Optymalizacja kognitywnych sieci optycznych**
Źródło finansowania	NCN
Budżet	557 200 zł
Okres realizacji	2018-09-03 – 2022-09-02
Rola w projekcie	Wykonawca

6.)

Tytuł	**Optymalizacja szkieletowych sieci optycznych z wykorzystaniem narzędzi modelowania predykcji ruchu sieciowego**
Źródło finansowania	NCN
Budżet	218 400 zł
Okres realizacji	2019-10-01 – 2022-09-30
Rola w projekcie	Wykonawca

9b.) Projekty zakończone, realizowane po uzyskaniu stopnia doktora:

1.)

Tytuł	**Algorytmy klasyfikacji niezbalansowanych strumieni danych**
Źródło finansowania	NCN
Budżet	613 920 zł
Okres realizacji	2018-09-04 – 2021-09-30
Rola w projekcie	Wykonawca

2.)

Tytuł	**Metody klasyfikacji wieloklasowej danych niezbalansowanych**
Źródło finansowania	NCN
Budżet	440 044 zł
Okres realizacji	2016-07-22— 2020-01-21
Rola w projekcie	Wykonawca

3.)

Tytuł	**Integration of base classifiers in geometrical space**
Źródło finansowania	NCN
Budżet	497 980 zł
Okres realizacji	2018-01-25 – 2021-01-24
Rola w projekcie	Wykonawca

4.)

Tytuł	**European Union's Horizon 2020 / SocialTruth**
Źródło finansowania	EU H2020
Budżet	13 665 800 zł
Okres realizacji	2019-05-16 – 2021-11-30
Rola w projekcie	Researcher/Expert

9c.) Projekty zakończone, realizowane przed uzyskaniem stopnia doktora:

1.)

Tytuł	**Złożone metody klasyfikacji danych strumieniowych wykorzystujące paradygmaty uczenia nienadzorowanego i aktywnego**
Źródło finansowania	NCN
Budżet	669 292 zł
Okres realizacji	2014-03-12 – 2017-03-11
Rola w projekcie	Wykonawca

2.10 Członkostwo w międzynarodowych lub krajowych organizacjach i towarzystwach naukowych wraz z informacją o pełnionych funkcjach

—

2.11 Informacja o odbytych stażach w instytucjach naukowych lub artystycznych, w tym zagranicznych, z podaniem miejsca, terminu, czasu trwania stażu i jego charakteru.

1.)

Jednostka	**Universidad del Pais Vasco** San Sebastian, Hiszpania
Termin stażu	15-08-2019 – 30-08-2019
Tematyka	*Redakcja wniosków projektowych i przetwarzanie sygnałów cyfrowych*
Charakter stażu	Celem stażu była wymiana wiedzy w zakresie metodyki prowadzenia i redakcji wniosków projektowych oraz wspólne badania z zakresu przetwarzania cyfrowych sygnałów wielowymiarowych. Podjęte prace badawcze, przerwane ze względu na epidemię COVID-19, zostały ostatnio wznowione, ale nie przyniosły jeszcze efektów w postaci publikacji artykułów naukowych.

2.)

Jednostka	**Virginia Commonwealth University** Department of Computer Science School of Engineerig Richmond, VA, USA
Termin stażu	16-10-2019 – 26-10-2019
Tematyka	*Przetwarzanie trudnych strumieni danych*
Charakter stażu	Celem stażu naukowego było wypracowanie nowych koncepcji z zakresu przetwarzania trudnych strumieni danych. Głównym efektem stażu stało się opracowanie strategii dywersyfikacji komitetów podprzestrzennych przez losowanie ich cech dominujących z rozkładów niejednostajnych, wyznaczonych przez analizę strumienia. Efektem dodatkowym są prowadzone wciąż badania nad rozszerzeniem tej strategii do zagadnienia wyjaśnialnych modeli rozpoznawania.

11a.) Staże zrealizowane przed uzyskaniem stopnia doktora:

1.)

Jednostka badawcza	**Universidad del Pais Vasco** San Sebastian, Hiszpania
Termin stażu	06-02-2016 – 21-02-2016
Tematyka	Klasyfikacja obrazów nadwidmowych
Charakter stażu	W ramach stażu uczestniczyłem w bieżących pracach zespołu kierowanego przez prof. Manuela Granę, zdobywając cenne doświadczenia z zakresu metodyki pracy naukowej. Wyjazd stanowił kontynuację współpracy podjętej już w roku 2014. Ponadto, uczestniczyłem w seminariach poświęconych przetwarzaniu danych nadwidmowych – stanowiących kluczowy obszar pracy doktorskiej oraz sformalizowałem metodę kluczową dla realizacji pracy doktorskiej.

2.12 Członkostwo w komitetach redakcyjnych i radach naukowych czasopism wraz z informacją o pełnionych funkcjach

—

2.13 Informacja o recenzowanych pracach naukowych lub artystycznych, w szczególności publikowanych w czasopismach międzynarodowych.

Wykonywałem recenzje prac dla następujących czasopism naukowych:

Czasopismo	Impact Factor
Machine Learning	2.940
Pattern Analysis and Applications	1.307
Applied Soft Computing	8.263
Pattern Recognition	7.74
Journal of Computational Science	3.976
IEEE's Geoscience and Remote Sensing Letters	3.966
IEEE Access	3.367
Symmetry	2.713
Entropy	2.524

2.14 Informacja o uczestnictwie w programach europejskich lub innych programach międzynarodowych

Tytuł	**European Union's Horizon 2020 / Social-Truth**
Źródło finansowania	EU H2020
Budżet	13 665 800 zł
Okres realizacji	2019-05-16 – 2021-11-30
Rola w projekcie	Researcher/Expert

Tytuł	**ENGINE – European research centre of Network intelliGence for IN-novation Enhacement**
Źródło finansowania	EC: Coordination and support actions (Supporting Action) Work programme topics addressed: Capacities Work Programme: Research Potential
Budżet	4 731 164 EUR
Okres realizacji	2013-06-01 – 2016-12-31
Rola w projekcie	Wykonawca

2.15 Informacja o udziale w zespołach badawczych, realizujących projekty inne niż określone w pkt. II.9.

—

2.16 Informacja o uczestnictwie w zespołach oceniających wnioski o finansowanie badań, wnioski o przyznanie nagród naukowych, wnioski w innych konkursach mających charakter naukowy lub dydaktyczny

—

3 INFORMACJA O WSPÓŁPRACY Z OTOCZENIEM SPOŁECZNYM I GOSPODARCZYM

3.1 Wykaz dorobku technologicznego

—

3.2 Informacja o współpracy z sektorem gospodarczym

- W roku 2016 współorganizowałem wraz z pracownikami *Credit Suisse Group AG* maraton programistyczny JellyPizzaHack dla studentów Politechniki Wrocławskiej.
- uczestnictwo w roli eksperta w debacie "*Szczepionka na kłamstwo*." organizowanej przez Fundację na rzecz Nauki Polskiej w ramach cyklu „*Ufajmy nauce*" (21.04.2022r.),
- prelekcja "*Wykorzystanie sztucznej inteligencji w walce z dezinformacją*" w ramach seminarium "*Sztuczna inteligencja w rozwoju miast i obszarów metropolitarnych*organizowanego przez Wrocławskie Centrum Akademickie pod patronatem World Urban Forum (9.03.2022r.),
- wykład w ramach seminarium "*Machine Learning to Combat Fake News and Media Manipulation*" organizowanego przez Elsevier w ramach cyklu webinarów (20.04.2022r.).
- wywiady radiowe dotyczące detekcji źródeł dezinformacji.

3.3 Uzyskane prawa własności przemysłowej, w tym uzyskane patenty krajowe lub międzynarodowe

—

3.4 Informacja o wdrożonych technologiach

—

3.5 Informacja o wykonanych ekspertyzach lub innych opracowaniach wykonanych na zamówienie instytucji publicznych lub przedsiębiorców

Opracowałem wniosek projektowy o finansowanie prac badawczo-rozwojowych do funduszu *EVIGAlfa*[1] dla startupu *Hinter.ai*[2]. W roku 2022, zgodnie z opracowaną przeze mnie agendą, zrealizowałem prace badawcze w ramach tego projektu zakończone wdrożeniem modułu automatycznej analizy danych. Efektem działań jest system przygotowujący dla menedżerów przedsiębiorstw listę zalecanych działań, których wdrożenie pomoże usprawnić funkcjonowanie organizacji, który w czerwcu 2022 otrzymał główną nagrodę w konkursie *Polskiej Agencji Rozwoju Przedsiębiorczości*[3] dla przedsiębiorców funkcjonujących na rynku nie dłużej niż 3 lata.

[1] https://evigalfa.pl

[2] https://hinter.ai

[3] https://pap-mediaroom.pl/biznes-i-finanse/najlepsze-programy-rozwoju

3.6 Informacja o udziale w zespołach eksperckich lub konkursowych

—

3.7 Informacja o projektach artystycznych realizowanych ze środowiskami pozaartystycznymi.

—

4 *INFORMACJE NAUKOMETRYCZNE*

Informacje dot. liczby punktów MEiN, współczynnika IF oraz liczby cytowań podane na podstawie wskaźników z dnia 27 sierpnia 2022.

1. Informacja o punktacji Impact Factor (w dziedzinach i dyscyplinach w których parametr ten jest powszechnie używany jako wskaźnik naukometryczny).

	Liczba prac z IF	Suma IF
Ogółem	16	75,743
Po uzyskaniu stopnia doktora	15	75,309
Przed uzyskaniem stopnia doktora	1	0,434

2. Informacja o liczbie cytowań publikacji wnioskodawcy, z oddzielnym uwzględnieniem autocytowań.

	Liczba wszystkich cytowań	Liczba cytowań bez autocytowań
Google Scholar	377	Brak danych
Web of Science	157	134
Scopus	276	240

3. Informacja o posiadanym indeksie Hirscha.

	h-indeks
Google Scholar	13
Web of Science	8
Scopus	10

4. Informacja o liczbie punktów MEiN.

	Liczba prac z listy MEiN	Suma punktów MEiN
Ogółem	49	3 995
Po uzyskaniu stopnia doktora	39	3 709
Przed uzyskaniem stopnia doktora	10	286

.....
(podpis wnioskodawcy)

ZALACZNIK 5

Deklaracje współautorów dotyczące wkładu pracy

Dla pracy Joanna Komorniczak, Paweł Zyblewski i Paweł Ksieniewicz. "Statistical Drift Detection Ensemble for batch processing of data streams". W: *Knowledge-Based Systems* 252 (2022), s. 109380. DOI: 10.1016/j.knosys.2022.109380:

- Joanna Komorniczak,
- Paweł Zyblewski.

Dla pracy Pawel Ksieniewicz. "Processing data stream with chunk-similarity model selection". W: *Applied Intelligence* (lip. 2022). DOI: 10.1007/s10489-022-03826-4:

- Joanna Komorniczak,
- Paweł Zyblewski.

Dla pracy Paweł Ksieniewicz i in. "Fake News Detection from Data Streams". W: *2020 International Joint Conference on Neural Networks (IJCNN)*. 2020, s. 1–8. DOI: 10.1109/IJCNN48605.2020.9207498:

- Paweł Zyblewski,
- Michał Choraś,
- Rafał Kozik,
- Agata Giełczyk,
- Michał Woźniak.

Dla pracy Paweł Ksieniewicz i in. "Data stream classification using active learned neural networks". W: *Neurocomputing* 353 (2019), s. 74–82. DOI: 10.1016/j.neucom.2018.05.130:

- Michał Woźniak,
- Bogusław Cyganek,
- Andrzej Kasprzak,
- Krzysztof Walkowiak.

Dla pracy Paweł Ksieniewicz i Michał Woźniak. "Imbalanced Data Classification Based on Feature Selection Techniques". W: *Intelligent Data Engineering and Automated Learning – IDEAL 2018*. Red. Hujun Yin i in. Cham: Springer International Publishing, 2018, s. 296–303. DOI: 10.1007/978-3-030-03496-2_33:

- Michał Woźniak.

Dla pracy Paweł Ksieniewicz, Bartosz Krawczyk i Michał Woźniak. "Ensemble of Extreme Learning Machines with trained classifier combination and statistical features for hyperspectral data". W: *Neurocomputing* 271 (2018), s. 28–37. DOI: 10.1016/j.neucom.2016.04.076:

- Bartosz Krawczyk,
- Michał Woźniak.

Dla pracy Paweł Ksieniewicz i Michał Woźniak. "Dealing with the task of imbalanced, multidimensional data classification using ensembles of exposers". W: *Proceedings of the First International Workshop on Learning with Imbalanced Domains: Theory and Applications*. Red. Paula Branco Luís Torgo i Nuno Moniz. T. 74. Proceedings of Machine Learning Research. PMLR, 22 Sep 2017, s. 164–175. URL: https://proceedings.mlr.press/v74/ksieniewicz17a.html:

- Michał Woźniak.

Pozostałe cztery publikacje wchodzące w skład cyklu stanowią dzieła jednoautorskie.

Wrocław, 1.08.2022

To whom it may concern:

This letter is to confirm that the contribution of authors of the paper

Joanna Komorniczak, Paweł Zyblewski i Paweł Ksieniewicz. "Statistical Drift Detection Ensemble for batch processing of data streams". W: Knowledge-Based Systems 252 (2022), s. 109380. doi: https://doi.org/10.1016/j.knosys.2022. 109380

are as follows:

Joanna Komorniczak
Overall contribution: 20%
Conceptualization, Software, Writing – Original Draft, Writing – Review & Editing.
Corresponding author.

Paweł Zyblewski
Overall contribution: 10%
Conceptualization, Methodology, Formal Analysis, Writing – Original Draft.

Paweł Ksieniewicz
Overall contribution: 70%
Conceptualization, Software, Validation, Formal Analysis, Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization, Supervision.

Sincerely,

Joanna Komorniczak, MSc.

Joanna Komorniczak

Wrocław, 1.08.2022

To whom it may concern:

This letter is to confirm that the contribution of authors of the paper

Joanna Komorniczak, Paweł Zyblewski i Paweł Ksieniewicz. "Statistical Drift Detection Ensemble for batch processing of data streams". W: Knowledge-Based Systems 252 (2022), s. 109380. doi: https://doi.org/10.1016/j.knosys.2022. 109380

are as follows:

Joanna Komorniczak
Overall contribution: 20%
Conceptualization, Software, Writing – Original Draft, Writing – Review & Editing.
Corresponding author.

Paweł Zyblewski
Overall contribution: 10%
Conceptualization, Methodology, Formal Analysis, Writing – Original Draft.

Paweł Ksieniewicz
Overall contribution: 70%
Conceptualization, Software, Validation, Formal Analysis, Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization, Supervision.

Sincerely,

Paweł Zyblewski
Wroclaw University of Science and Technology
Department of Computer Systems and Networks

P. Zyblewski

Wrocław, 1.08.2022

To whom it may concern:

This letter is to confirm that the contribution of authors of the paper

Joanna Komorniczak, Pawel Zyblewski i Pawel Ksieniewicz. "Prior Probability Estimation in Dynamically Imbalanced Data Streams". W: 2021 International Joint Conference on Neural Networks (IJCNN). IEEE, lip. 2021. doi: 10.1109/ijcnn52387.2021.9533795

are as follows:

Joanna Komorniczak
Overall contribution: 20%
Conceptualization, Software, Writing – Original Draft, Writing – Review & Editing.
Corresponding author.

Paweł Zyblewski
Overall contribution: 10%
Conceptualization, Methodology, Formal Analysis, Writing – Original Draft.

Paweł Ksieniewicz
Overall contribution: 70%
Conceptualization, Software, Validation, Formal Analysis, Investigation, Resources, Writing - Original Draft, Writing - Review & Editing, Visualization, Supervision.

Sincerely,

Joanna Komorniczak, MSc.

Joanna Komorniczak

Wrocław, 1.08.2022

To whom it may concern:

This letter is to confirm that the contribution of authors of the paper

Joanna Komorniczak, Pawel Zyblewski i Pawel Ksieniewicz. "Prior Probability Estimation in Dynamically Imbalanced Data Streams". W: 2021 International Joint Conference on Neural Networks (IJCNN). IEEE, lip. 2021. doi: 10.1109/ijcnn52387.2021.9533795

are as follows:

Joanna Komorniczak
Overall contribution: 20%
Conceptualization, Software, Writing – Original Draft, Writing – Review & Editing.
Corresponding author.

Paweł Zyblewski
Overall contribution: 10%
Conceptualization, Methodology, Formal Analysis, Writing – Original Draft.

Paweł Ksieniewicz
Overall contribution: 70%
Conceptualization, Software, Validation, Formal Analysis, Investigation, Resources, Writing - Original Draft, Writing - Review & Editing, Visualization, Supervision.

Sincerely,

Paweł Zyblewski
Wroclaw University of Science and Technology
Department of Computer Systems and Networks

P. Zyblewski

Wrocław, 1.08.2022

To whom it may concern:

This letter is to confirm that the contribution of authors of the paper

Paweł Ksieniewicz; Paweł Zyblewski; Michał Choraś; Rafał Kozik; Agata Giełczyk; Michał Woźniak "Fake News Detection from Data Streams". W: 2020 International Joint Conference on Neural Networks (IJCNN). 2020, s. 1–8. doi: 10.1109/IJCNN48605.2020.9207498
are as follows:

Paweł Ksieniewicz
Overall contribution: 50%
Conceptualization, Methodology, Software, Validation, Investigation, Writing – Original Draft, Writing - Review & Editing, Visualization

Paweł Zyblewski
Overall contribution: 5%
Formal Analysis, Writing – Original Draft, Writing - Review & Editing

Michał Choraś
Overall contribution: 30%
Conceptualization, Methodology, Formal Analysis, Writing – Original Draft, Writing - Review & Editing.

Rafał Kozik
Overall contribution: 5%
Formal Analysis, Writing – Original Draft.

Agata Giełczyk
Overall contribution: 5%
Formal Analysis, Writing – Original Draft.

Michał Woźniak
Overall contribution: 5%
Resources, Formal Analysis, Writing – Original Draft, Writing - Review & Editing

Sincerely,

Paweł Zyblewski
Wroclaw University of Science and Technology
Department of Computer Systems and Networks

P. Zyblewski

Wrocław, 1.08.2022

To whom it may concern:

This letter is to confirm that the contribution of authors of the paper

Paweł Ksieniewicz; Paweł Zyblewski; Michał Choraś; Rafał Kozik; Agata Giełczyk; Michał Woźniak "Fake News Detection from Data Streams". W: 2020 International Joint Conference on Neural Networks (IJCNN). 2020, s. 1–8. doi: 10.1109/IJCNN48605.2020.9207498
are as follows:

Paweł Ksieniewicz
Overall contribution: 50%
Conceptualization, Methodology, Software, Validation, Investigation, Writing – Original Draft, Writing - Review & Editing, Visualization

Paweł Zyblewski
Overall contribution: 5%
Formal Analysis, Writing – Original Draft, Writing - Review & Editing

Michał Choraś
Overall contribution: 30%
Conceptualization, Methodology, Formal Analysis, Writing – Original Draft, Writing - Review & Editing.

Rafał Kozik
Overall contribution: 5%
Formal Analysis, Writing – Original Draft.

Agata Giełczyk
Overall contribution: 5%
Formal Analysis, Writing – Original Draft.

Michał Woźniak
Overall contribution: 5%
Resources, Formal Analysis, Writing – Original Draft, Writing - Review & Editing

Sincerely,

Michał Choraś
Bydgoszcz University of Science and Technology

Wrocław, 1.08.2022

To whom it may concern:

This letter is to confirm that the contribution of authors of the paper

Paweł Ksieniewicz; Paweł Zyblewski; Michał Choraś; Rafał Kozik; Agata Giełczyk; Michał Woźniak "Fake News Detection from Data Streams". W: 2020 International Joint Conference on Neural Networks (IJCNN). 2020, s. 1–8. doi: 10.1109/IJCNN48605.2020.9207498
are as follows:

Paweł Ksieniewicz
Overall contribution: 50%
Conceptualization, Methodology, Software, Validation, Investigation, Writing – Original Draft, Writing - Review & Editing, Visualization

Paweł Zyblewski
Overall contribution: 5%
Formal Analysis, Writing – Original Draft, Writing - Review & Editing

Michał Choraś
Overall contribution: 30%
Conceptualization, Methodology, Formal Analysis, Writing – Original Draft, Writing - Review & Editing.

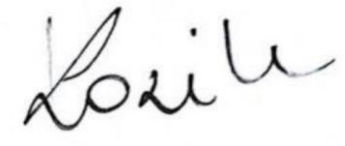
Rafał Kozik
Overall contribution: 5%
Formal Analysis, Writing – Original Draft.

Agata Giełczyk
Overall contribution: 5%
Formal Analysis, Writing – Original Draft.

Michał Woźniak
Overall contribution: 5%
Resources, Formal Analysis, Writing – Original Draft, Writing - Review & Editing

Sincerely,

Rafał Kozik
Bydgoszcz University of Science and Technology



Wrocław, 1.08.2022

To whom it may concern:

This letter is to confirm that the contribution of authors of the paper

Paweł Ksieniewicz; Paweł Zyblewski; Michał Choraś; Rafał Kozik; Agata Giełczyk; Michał Woźniak
"Fake News Detection from Data Streams". W: 2020 International Joint Conference on Neural Networks (IJCNN). 2020, s. 1–8. doi: 10.1109/IJCNN48605.2020.9207498
are as follows:

Paweł Ksieniewicz
Overall contribution: 50%
Conceptualization, Methodology, Software, Validation, Investigation, Writing – Original Draft, Writing - Review & Editing, Visualization

Paweł Zyblewski
Overall contribution: 5%
Formal Analysis, Writing – Original Draft, Writing - Review & Editing

Michał Choraś
Overall contribution: 30%
Conceptualization, Methodology, Formal Analysis, Writing – Original Draft, Writing - Review & Editing.

Rafał Kozik
Overall contribution: 5%
Formal Analysis, Writing – Original Draft.

Agata Giełczyk
Overall contribution: 5%
Formal Analysis, Writing – Original Draft.

Michał Woźniak
Overall contribution: 5%
Resources, Formal Analysis, Writing – Original Draft, Writing - Review & Editing

Sincerely,

Agata Giełczyk
Bydgoszcz University of Science and Technology

Wrocław, 1.08.2022

To whom it may concern:

This letter is to confirm that the contribution of authors of the paper

Paweł Ksieniewicz; Paweł Zyblewski; Michał Choraś; Rafał Kozik; Agata Giełczyk; Michał Woźniak "Fake News Detection from Data Streams". W: 2020 International Joint Conference on Neural Networks (IJCNN). 2020, s. 1–8. doi: 10.1109/IJCNN48605.2020.9207498
are as follows:

Paweł Ksieniewicz
Overall contribution: 50%
Conceptualization, Methodology, Software, Validation, Investigation, Writing – Original Draft, Writing - Review & Editing, Visualization

Paweł Zyblewski
Overall contribution: 5%
Formal Analysis, Writing – Original Draft, Writing - Review & Editing

Michał Choraś
Overall contribution: 30%
Conceptualization, Methodology, Formal Analysis, Writing – Original Draft, Writing - Review & Editing.

Rafał Kozik
Overall contribution: 5%
Formal Analysis, Writing – Original Draft.

Agata Giełczyk
Overall contribution: 5%
Formal Analysis, Writing – Original Draft.

Michał Woźniak
Overall contribution: 5%
Resources, Formal Analysis, Writing – Original Draft, Writing - Review & Editing

Sincerely,

Michał Woźniak
Wroclaw University of Science and Technology
Department of Computer Systems and Networks

Wrocław, 1.08.2022

To whom it may concern:

This letter if to confirm that the contributions of authors of the paper

Paweł Ksieniewicz, Michał Woźniak, Bogusław Cyganek, Andrzej Kasprzak, Krzysztof Walkowiaka, Data stream classification using active learned neural networks, Volume 353, 11 August 2019, Pages 74-82, https://doi.org/10.1016/j.neucom.2018.05.130

are as follows:

Paweł Ksieniewicz

Overall contribution: 70%
Proposed the algorithm
Contributed to design, implementation, experiments, data analysis and interpretation, writing.
Corresponding author.

Michał Woźniak

Overall contribution: 15%
Proposed joint research and selected the journal. Correction of the algorithm proposal
Contributed to data collection and data analysis.

Bogusław Cyganek

Overall contribution: 5%
Contributed to data collection and data analysis.

Andrzej Kasprzak

Overall contribution: 5%
Contributed to data collection and data analysis.

Krzysztof Walkowiak

Overall contribution: 5%
Contributed to data collection and data analysis.

Sincerely,

Michał Woźniak
Wroclaw University of Science and Technology
Department of Computer Systems and Networks

Kraków, 28.05.2019

To whom it may concern:

This letter if to confirm that the contributions of authors of the paper

Paweł Ksieniewicz, Michał Woźniak, Bogusław Cyganek, Andrzej Kasprzak, Krzysztof Walkowiaka, Data stream classification using active learned neural networks, Volume 353, 11 August 2019, Pages 74-82, https://doi.org/10.1016/j.neucom.2018.05.130

are as follows:

Paweł Ksieniewicz
Overall contribution: 70%
Proposed the algorithm
Contributed to design, implementation, experiments, data analysis and interpretation, writing.
Corresponding author.

Michał Woźniak
Overall contribution: 15%
Proposed joint research and selected the journal.
Correction of the algorithm proposal
Contributed to data collection and data analysis.

Bogusław Cyganek
Overall contribution: 5%
Contributed to data collection and data analysis.

Andrzej Kasprzak
Overall contribution: 5%
Contributed to data collection and data analysis.

Krzysztof Walkowiak
Overall contribution: 5%
Contributed to data collection and data analysis.

Sincerely,

Prof. dr hab. inż. Bogusław Cyganek
AGH University of Science and Technology
Faculty of Computer Science, Electronics and Telecommunication
Department of Electronics
Al. Mickiewicza 30/C2/413
30-059 Kraków
Poland

Wrocław, 1.08.2022

To whom it may concern:

This letter if to confirm that the contributions of authors of the paper

Paweł Ksieniewicz, Michał Woźniak, Bogusław Cyganek, Andrzej Kasprzak, Krzysztof Walkowiaka, Data stream classification using active learned neural networks, Volume 353, 11 August 2019, Pages 74-82, https://doi.org/10.1016/j.neucom.2018.05.130

are as follows:

Paweł Ksieniewicz

Overall contribution: 70%
Proposed the algorithm
Contributed to design, implementation, experiments, data analysis and interpretation, writing.
Corresponding author.

Michał Woźniak

Overall contribution: 15%
Proposed joint research and selected the journal. Correction of the algorithm proposal
Contributed to data collection and data analysis.

Bogusław Cyganek

Overall contribution: 5%
Contributed to data collection and data analysis.

Andrzej Kasprzak

Overall contribution: 5%
Contributed to data collection and data analysis.

Krzysztof Walkowiak

Overall contribution: 5%
Contributed to data collection and data analysis.

Sincerely,

Andrzej Kasprzak
Wroclaw University of Science and Technology
Department of Computer Systems and Networks

Wrocław, June 10th, 2019

To whom it may concern:

This letter if to confirm that the contributions of authors of the paper

Paweł Ksieniewicz, Michał Woźniak, Bogusław Cyganek, Andrzej Kasprzak, Krzysztof Walkowiaka, Data stream classification using active learned neural networks, Volume 353, 11 August 2019, Pages 74-82, https://doi.org/10.1016/j.neucom.2018.05.130

are as follows:

Paweł Ksieniewicz
Overall contribution: 70%
Proposed the algorithm
Contributed to design, implementation, experiments, data analysis and interpretation, writing.
Corresponding author.

Michał Woźniak
Overall contribution: 15%
Proposed joint research and selected the journal.
Correction of the algorithm proposal
Contributed to data collection and data analysis.

Bogusław Cyganek
Overall contribution: 5%
Contributed to data collection and data analysis.

Andrzej Kasprzak
Overall contribution: 5%
Contributed to data collection and data analysis.

Krzysztof Walkowiak
Overall contribution: 5%
Contributed to data collection and data analysis.

Sincerely,

Krzysztof Walkowiak
Department of Systems and Computer Networks
Wrocław University of Sciencec and Techology

Wrocław, 1.08.2022

To whom it may concern:

This letter if to confirm that the contributions of authors of the paper

Paweł Ksieniewicz i Michał Woźniak. “Imbalanced Data Classification Based on Feature Selection Techniques”. W: Intelligent Data Engineering and Automated Learning – IDEAL 2018. Red. Hujun Yin i in. Cham: Springer International Publishing, 2018, s. 296–303. doi: 10.1007/978-3-030-03496-2_33

are as follows:

Paweł Ksieniewicz

Overall contribution: 80%
Methodology, Software, Validation, Investigation, Writing – Original Draft, Writing – Review & Editing, Visualization

Michał Woźniak

Overall contribution: 20%
Conceptualization, Investigation, Writing – Original Draft, Supervision

Sincerely,

Michał Woźniak
Wroclaw University of Science and Technology
Department of Computer Systems and Networks

Wrocław, 1.08.2022

To whom it may concern:

This letter is to confirm that the contribution of authors of the paper

Ksieniewicz, Paweł, Bartosz Krawczyk, and Michał Woźniak. "Ensemble of Extreme Learning Machines with trained classifier combination and statistical features for hyperspectral data." Neurocomputing 271 (2018): 28-37.

are as follows:

Paweł Ksieniewicz
Overall contribution: 60%
Proposed the algorithm
Contributed to design, implementation, experiments, data analysis and interpretation, writing.
Corresponding author.

Bartosz Krawczyk
Overall contribution: 20%
Correction of the algorithm proposal.
Contributed to data collection and data analysis.

Michał Woźniak
Overall contribution: 20%
Proposed joint research and selected the journal.
Contributed to data collection and data analysis.

Sincerely,

Bartosz Krawczyk, Ph. D.
Virginia Commonwealth University
Department of Computer Science
Engineering East Hall, Room E4238
Richmond, VA, United States

Wrocław, 1.08.2022

To whom it may concern:

This letter is to confirm that the contribution of authors of the paper

Ksieniewicz, Paweł, Bartosz Krawczyk, and Michał Woźniak. "Ensemble of Extreme Learning Machines with trained classifier combination and statistical features for hyperspectral data." Neurocomputing 271 (2018): 28-37.

are as follows:

Paweł Ksieniewicz
Overall contribution: 60%
Proposed the algorithm
Contributed to design, implementation, experiments, data analysis and interpretation, writing.
Corresponding author.

Bartosz Krawczyk
Overall contribution: 20%
Correction of the algorithm proposal.
Contributed to data collection and data analysis.

Michał Woźniak
Overall contribution: 20%
Proposed joint research and selected the journal.
Contributed to data collection and data analysis.

Sincerely,

Michał Woźniak
Wroclaw University of Science and Technology
Department of Computer Systems and Networks

Wrocław, 1.08.2022

To whom it may concern:

This letter if to confirm that the contributions of authors of the paper

Paweł Ksieniewicz and Michał Woźniak. “Dealing with the task of imbalanced, multidimensional data classification using ensembles of exposers”. In: Proceedings of the First International Workshop on Learning with Imbalanced Domains: Theory and Applications. Red. Paula Branco Luís Torgo and Nuno Moniz. T. 74. Proceedings of Machine Learning Research. PMLR, 22 Sep 2017, s. 164–175. url: https://proceedings.mlr.press/v74/ ksieniewicz17a.html

are as follows:

Paweł Ksieniewicz

Overall contribution: 80%
Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Resources, Data Curation, Writing – Original Draft, Writing – Review & Editing, Visualization

Michał Woźniak

Overall contribution: 20%
Conceptualization, Methodology, Formal Analysis, Investigation, Writing – Original Draft, Supervision

Sincerely,

Michał Woźniak
Wroclaw University of Science and Technology
Department of Computer Systems and Networks

ZALACZNIK 6

Publikacje wchodzące w skład osiągnięcia naukowego

[C1] Joanna Komorniczak, Paweł Zyblewski i Paweł Ksieniewicz. "Statistical Drift Detection Ensemble for batch processing of data streams". W: *Knowledge-Based Systems* 252 (2022), s. 109380. DOI: 10.1016/j.knosys.2022.109380

[C2] Pawel Ksieniewicz. "Processing data stream with chunk-similarity model selection". W: *Applied Intelligence* (lip. 2022). DOI: 10.1007/s10489-022-03826-4

[C3] Joanna Komorniczak, Pawel Zyblewski i Pawel Ksieniewicz. "Prior Probability Estimation in Dynamically Imbalanced Data Streams". W: *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, lip. 2021. DOI: 10.1109/ijcnn52387.2021.9533795

[C4] Paweł Ksieniewicz. "The prior probability in the batch classification of imbalanced data streams". W: *Neurocomputing* 452 (wrz. 2021), s. 309–316. DOI: 10.1016/j.neucom.2019.11.126

[C5] Paweł Ksieniewicz i in. "Fake News Detection from Data Streams". W: *2020 International Joint Conference on Neural Networks (IJCNN)*. 2020, s. 1–8. DOI: 10.1109/IJCNN48605.2020.9207498

[C6] Pawel Ksieniewicz. "Combining Random Subspace Approach with smote Oversampling for Imbalanced Data Classification". W: *Hybrid Artificial Intelligent Systems*. Red. Hilde Pérez García i in. Cham: Springer International Publishing, 2019, s. 660–673. ISBN: 978-3-030-29859-3. DOI: 10.1007/978-3-030-29859-3_56

[C7] Paweł Ksieniewicz i in. "Data stream classification using active learned neural networks". W: *Neurocomputing* 353 (2019), s. 74–82. DOI: 10.1016/j.neucom.2018.05.130

[C8] Pawel Ksieniewicz. "Undersampled Majority Class Ensemble for highly imbalanced binary classification". W: *Proceedings of the Second International Workshop on Learning with Imbalanced Domains: Theory and Applications*. Red. Luís Torgo i in. T. 94. Proceedings of Machine Learning Research. PMLR, paź. 2018, s. 82–94. URL: https://proceedings.mlr.press/v94/ksieniewicz18a.html

[C9] Paweł Ksieniewicz i Michał Woźniak. "Imbalanced Data Classification Based on Feature Selection Techniques". W: *Intelligent Data Engineering and Automated Learning – IDEAL 2018*. Red. Hujun Yin i in. Cham: Springer International Publishing, 2018, s. 296–303. DOI: 10.1007/978-3-030-03496-2_33

[C10] Paweł Ksieniewicz, Bartosz Krawczyk i Michał Woźniak. "Ensemble of Extreme Learning Machines with trained classifier combination and statistical features for hyperspectral data". W: *Neurocomputing* 271 (2018), s. 28–37. DOI: 10.1016/j.neucom.2016.04.076

[C11] Paweł Ksieniewicz i Michał Woźniak. "Dealing with the task of imbalanced, multidimensional data classification using ensembles of exposers". W: *Proceedings of the First International Workshop on Learning with Imbalanced Domains: Theory and Applications*. Red. Paula Branco Luís Torgo i Nuno Moniz. T. 74. Proceedings of Machine Learning Research. PMLR, 22 Sep 2017, s. 164–175. URL: https://proceedings.mlr.press/v74/ksieniewicz17a.html

[C1]
Joanna Komorniczak, Paweł Zyblewski i Paweł Ksieniewicz. “Statistical Drift Detection Ensemble for batch processing of data streams”. W: *Knowledge-Based Systems* 252 (2022), s. 109380. DOI: 10.1016/j.knosys.2022.109380

Knowledge-Based Systems 252 (2022) 109380

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Statistical Drift Detection Ensemble for batch processing of data streams

Joanna Komorniczak *, Paweł Zyblewski, Paweł Ksieniewicz

Department of Systems and Computer Networks, Wroclaw University of Science and Technology, wyb. Wyspianskiego 27, Wrocław, 50370, Poland

ARTICLE INFO

Article history:
Received 24 March 2022
Received in revised form 2 June 2022
Accepted 3 July 2022
Available online 14 July 2022

MSC:
00-01
99-00

Keywords:
Data streams
Concept drift
Drift detection
Statistical drift detection
Classification

ABSTRACT

Among the difficulties being considered in data stream processing, a particularly interesting one is the phenomenon of concept drift. Methods of concept drift detection are frequently used to eliminate the negative impact on the quality of classification in the environment of evolving concepts. This article proposes *Statistical Drift Detection Ensemble* (SDDE), a novel method of concept drift detection. The method uses *drift magnitude* and *conditioned marginal covariate drift* measures, analyzed by an ensemble of detectors, whose members focus on random subspaces of the stream's features. The proposed detector was compared with *state-of-the-art* methods on both synthetic data streams and the semi-synthetic streams generated based on the real-world concepts. A series of computer experiments and a statistical analysis of the results, both for the classification accuracy and *Drift Detection errors* were carried out and confirmed the effectiveness of the proposed method.

1. Introduction

In recent years, with the dynamic development of mobile technologies and network services, the process of data transmission, analysis, and collection has been taking place almost continuously — from communication via computer networks to monitoring of weather conditions [1]. In conjunction with these events and the forecasted growth of generated data [2], the demand for methods dedicated to the processing of streaming data is growing.

The hallmark of a data stream is a potentially infinite flow of data. There are also expected difficulties in the case of stream data analysis – e.g., missing values [3] or lack of uniformity [4]. Methods dedicated to analyzing streaming data should be prepared to deal with any faults and inconveniences throughout the data inflow. They should also assume single-time access to the data [5] – one cannot allow the accumulation of a theoretically infinite amount of data in the physically limited memory of the implemented method [6].

Changes are a natural consequence of analyzing infinitely incoming data. As a result of daily physiological and seasonal cycles and changes in trends in social media, we will observe drifts both in concept and in the rate of data imbalance [7,8]. Concept drifts describe the change of posterior probability in the data stream over time. In the context of the dynamics of these changes, we can divide concept drifts into *sudden*, *incremental*, and *gradual* [9]. In the event of *sudden* drifts, the concept is changing rapidly in the span of just a few consecutive observations. In the case of *gradual* and *incremental* concept drift, the changes are more fluid, stretched over time. Correspondingly in *gradual* drifts – the change is smooth – but there are moments of coexistence of two separate concepts without a uniform transition between them. Drifts can as well be described by *recurring context* – a concept that has arisen in the past may reappear after some time period.

Another known taxonomy is the separation into *real* and *virtual* drifts. *Real* drifts, also mentioned in the literature as *class drifts* [10], are related to the shift of the decision boundary during processing. In the case of *virtual* drifts, on the other hand, despite the changes taking place in the data distribution, the decision boundary is not shifted between the classes in the recognition problem [4]. However, over the years, different definitions of *virtual* drift arose. According to the original work of Widmer et al. [11], this category of drifts was viewed as the effect of the computer model's bias changing over time [2].

The main contributions of the following work are:

- Proposing an effective concept drift detection ensemble method, analyzing *drift magnitude* and *conditioned marginal covariate drift* measures, without a need for base classifier evaluation,
- Implementation of the proposed method as well as *state-of-the-art* drift detectors for batch processing of data streams

* Corresponding author.
E-mail address: joanna.komorniczak@vp.pl (J. Komorniczak).

https://doi.org/10.1016/j.knosys.2022.109380
0950-7051/

in *Python* programming language available in the GitHub repository,[1]
- Proposing three base *Drift Detection error* measures for assessing the quality of drift detection in data streams,
- Evaluation of the proposed method performance and comparison with reference drift detectors.

2. Related works

2.1. Drift detectors

Drift detectors are one of the possible approaches to minimizing the negative effect of concept drifts on the effectiveness of *recognition systems*. Detectors are usually integrated with classifiers [4], not only to have a possibility to utilize the measure of a classification error rate but also to recover its quality after the drift event. The usual procedure used in the case of detection is to replace the base classifier with a new one [12]. Effective and well-known drift detectors using classification error measurements are – among others – *Drift Detection Method* (DDM) [13] and *Early Drift Detection Method* (EDDM) [14].

DDM detects drifts based on the classification error of the base classifier and its standard deviation. It uses the assumption that the quality of the classification should increase with the time of stream processing. If otherwise, the method detects a change of concept. EDDM uses a similar method but measures the distances between upcoming classification errors and assumes that these distances should decrease with time. As for DDM – if otherwise – a concept change is signaled.

Adaptive Windowing (ADWIN) [15] uses the instances of a resizable sliding window. Two window instances – representing old and new data – are stored in memory. The specific difference between the mean values of the window data determines the moment of concept change detection. The window approach was also used in *Paired Learners* (PL) [16] method of concept drift detection. The PL algorithm uses two recognition models — the *static* one, trained with all incoming patterns, and the *reactive* one, trained only with patterns from the recent past. If the classification quality of the *static* model falls below the classification quality of the *reactive* classifier, a concept change is marked.

A *Drift Detection Method* based on the Hoeffding's inequality (HDDM) [17] monitors the performance during the learning process, using probability inequalities. The method can use two approaches — moving averages (HDDM_A), which is more suitable for sudden changes in concept, and weighted moving averages (HDDM_W), dedicated to gradual concept drifts detection.

An interesting approach, being also the starting point for the proposal presented in this article, is *Statistical Drift Detection Method* (SDDM) [18] – a drift detector dedicated to the environment with the lack of immediate access to the classification quality. In real-world applications, labels may often arrive after an extended period or not be available. Detecting drift with delay will not provide the benefits of an immediate model rebuild [6], therefore will affect classification accuracy. SDDM provides, in addition to the detection itself, information about the source of the drift and its nature. Therefore it belongs to interpretable machine learning methods. The mentioned algorithms enable the model's behaviors and predictions to be understood by humans. By using interpretable methods, we can ensure that the systems are not prejudiced against the particular types of input [19].

A series of measures have been proposed to determine the drift characteristics [10], e.g. *drift magnitude* – a measure that describes the distance between two concepts. The original measure of distance was *Hellinger Distance*. The publication by Webb et al. [20] proposes measures of *total drift magnitude*, which uses *total variation distance* [21], and two other *marginal drift magnitude* measures: *conditioned marginal covariate drift* and *posterior drift*.

The research by Micevska et al. [18] shows that the measures mentioned above become uninformative in high dimensional spaces; hence the authors decided to evaluate streams based on individual features' *concept drift magnitude*. However, reducing the analysis level to single dimensions also requires the subsequent integration of its results to obtain aggregate detection information for the entire stream.

Ensembles dedicated to minimizing the negative effect of concept drift on pattern recognition tasks have been proposed [22,23], some of which use auxiliary drift detection methods. There exist as well ensembles dedicated to the concept drift detection. The methods proposed in [24,25] integrate detections of different base drift detectors to boost their performance. Overall, the topic of ensemble-assisted drift detection has not been well explored [5].

The drift detection methods' effectiveness measures are often based on the classification quality during stream processing. The work by Bifet [26] sensibly advises not to evaluate drift detection methods using only overall classification quality. Evaluation based on the classification quality is derived from the most common and straightforward type of action performed in the case of the concept change, aimed at maintaining the quality of the classification — updating the classifier or replacing the current one with a new instance [27]. The experiment carried out in the publication mentioned above [26] shows that the reference method that does not detect changes but artificially simulates the detection can achieve better results than all of the examined detectors.

Three basic measures for the assessment of drift detectors are proposed: *Mean Time between False Alarms*, *Mean Time to Detection*, *Missed Detection Rate*, and two aggregated measures: *Average Run Length* and *Mean Time Ratio*. Before drift occurs, all detections are treated as a false alarm. After the actual drift, the first detection is considered a true alarm. These measures require ground-truth of a stream in the context of the concept changes. Obtaining such information is a significant advantage of synthetic drift injection. In the real-world streams, we cannot be sure about the moment of the concept drift and their type, which makes carrying out experimental evaluation on this type of data complex and highly inconclusive [27].

This publication proposes evaluations using a series of measures other than the ones proposed by Bifet [26]. In the case of gradual and incremental drifts, the unequivocal moment of the concept change is indistinct. In the streams used in this publication, ground truth is determined in the middle of the concept change period. All detections recognized earlier than the ground truth would be marked as a false alarm, which means that the correct detection of incremental and gradual drift, appearing in its initial phase, would be interpreted as erroneous. Such an evaluation approach significantly lowers the results for methods efficient in detecting non-sudden drift. The moment of detection in incremental and gradual drifts should depend on the detection method's sensitivity.

Additionally, according to the mentioned publication, the ideal detector's average time between false alarms should be high. In the case of several detections signaling one non-sudden drift of long duration, the value of this measure will be small; however, we consider such behavior of the detector as desirable.

[1] https://github.com/w4k2/statistical-drift-detection.

2.2. Ensemble methods for drifting data stream classification

Ensemble methods are another approach that allows for minimizing the impact of concept drift on the performance of the machine learning systems. These algorithms process the data stream in batches/data chunks or an online manner — one instance at a time [28]. In this case, the goal is to design the ensemble and the combination rule to adapt to changes due to the concept drift emergence.

The majority of batch-based methods are the development of an idea presented initially by the *Streaming Ensemble Algorithm* (SEA) [29]. This method maintains a pool of classifiers with a predetermined size, consisting of models trained on consecutive data chunks. If the limit of ensemble size is exceeded, the classifier with the lowest classification accuracy is removed. Another example is the *Accuracy Weighted Ensemble* (AWE) [30], based on mean square error and proposed by Wang et al. AWE was later extended by Brzezinski and Stefanowski, who proposed the *Accuracy Updated Ensemble* (AUE), introducing the ability to update the base models [31]. Wozniak et al. introduced the *Weighted Aging Ensemble* (WAE), which modifies AWE by allowing various classifier selection and weight calculation approaches [32]. Elwell and Polikar proposed *Learn++.NSE* [33], which modifies *Learn++* [34] by setting weight for training samples to deal with concept drift occurrence, while Gomes et al. introduced the *Adaptive Random Forest* (ARF) algorithm, which employs resampling and adaptive operators to cope with various types of concept drift [35].

Among online ensemble methods for data stream classification, we can distinguish *Online Bagging* (OB) proposed by Oza, which updates base models with the appearance of a new sample according to the Poisson distribution [36]. Later, Bifet et al. modified OB by allowing specifying the lambda value and introducing the output detection codes in the form of the *Leveraging Bagging* [37]. Gomes et al. proposed the *Streaming Random Patches* (SRP) algorithm [38], which combines *Online Bagging* with *Random Subspace* [39]. The sparse online learning algorithm was employed in the *Sparse Online Classification* (SOC) [40] framework by Wang et al.

Lastly, we can distinguish ensemble methods that combine batch-based and online approaches. An example of such an algorithm is the *Kappa Updated Ensemble* (KUE) proposed by Cano and Krawczyk [41]. KUE uses Kappa statistics for dynamic classifier weighting and selection while simultaneously allowing voting abstaining of chosen base models.

3. Methods

In order to enable the experimental evaluation of the method proposed in this paper, it was necessary to develop three essential processing blocks. The first is the *stream-learn* meta-estimator described in Section 3.1, which defines the strategy of using the drift detector in the data stream classification, allowing for effective measurement of the classification model quality in the course of processing. The second – described in Section 3.2 – is a set of evaluation metrics that effectively validate detection quality in environments with non-sudden drifts. The third – most important for this work and described in Section 3.3 – the *Statistical Drift Detection Ensemble*, which is a new method of data stream drift detection, using statistical measures of data stream properties calculated on problem subspaces, integrated into ensemble as a set of binary detectors.

3.1. Meta-estimator

Some concept change detection methods determine their decision based on the classifier's output [12]. Detectors are often integrated with classifiers to enable an up-to-date analysis of classification errors, which allows regeneration of the classification quality after a concept change. If the detector is two-state, i.e., it signalizes warnings of a concept change prior to detection, a parallel classifier can be created and trained with incoming data. If detection is confirmed, the old model is replaced with the new one [4].

Not all detectors analyzed in this paper provide warnings. Hence only confident detections are taken into account. When a concept change is marked, the classifier is restored to its initial state, and the training is restarted from patterns derived from the new concept. The described procedure is performed by implemented meta-estimator. The model passes problem instances to both the detector and the classifier during stream processing and acquires information about recognized detections.

The use of the described meta-estimator allows the evaluation of the detection quality based on the classification accuracy results and makes it possible to compare the detection methods. When the concept changes, the classification accuracy usually degenerates. Therefore if the detector cannot recognize the concept drift, the overall quality of the classification would decrease throughout stream processing. Delay of concept change detection would also harm the classification accuracy score.

3.2. Drift detection errors

Research on the reliable assessment of the recognition model's quality, in particular in the field of classification, often indicates the problem of using aggregated metrics, such as *F-score* or *Gmean*, which tend to hide some part of proper classification efficiency [42]. The apparent profit resulting from the assessment of a single criterion, in a comparative review of the effectiveness of the methods, does not allow obtaining complete information on the processing [43].

In addition, it is crucial where the non-sudden drift reference point is located on the course of a stream. By default, as indicated in the Related works section, it is defined as the equilibrium point of the dynamics of changes [26], the effects of which might be observed – by a suitable method of detection – earlier, at the first signs of drift. Theoretically, this problem could be solved by shifting the ground truth to the very beginning of the drift period. However, this would raise the question of whether it would not be necessary to treat all the detections occurring during the drift as correct. Such an approach would not measure the differences between the detectors' effectiveness. However, it would only give information about its detection and would require further modification and further complications of the metric. Additionally, in the case of non-sudden drifts, it is difficult to identify the correct beginning of the drift. We must remember that concept drifts occur in a space that is both continuous (course of the stream in which we functionally determine the probability of an object coming from a given concept) and discrete (sampling of this function by the objects that appear). Therefore, it is imperative to propose a reasonable compromise of proper drift detection for non-sudden cases.

Trying to avoid the problems mentioned above in the field with a much less structured research protocol, which is the concept drift detection, it was decided not to use or propose any aggregate metric. Batch processing of the data stream allows us to collect indexes of the chunks in which the detection occurred. In the case of synthetic streams, there is also a possibility to calculate the central point of concept drift. Therefore,

in the experimental analysis presented in this paper, three easy to interpret, simple metrics defining the basic properties of drift detectors will be used:

- D_1 – *closest drift* – the distance of each detection from the nearest drift, normalized as the arithmetic mean from the obtained values.
- D_2 – *closest detection* – the distance of each drift from the nearest detection, normalized as the arithmetic mean from the obtained values.
- R – *drift to detection ratio* – the proportions between number of drifts and number of detection. The measure is scaled to obtain the optimum value at zero. Absolute value of the scaled ratio is the final error.

Additionally, apart from the basic drift detection metric, the classification accuracy score of the final decision system will also be calculated.

3.3. Statistical Drift Detection Ensemble

This section introduces an ensemble concept drift detection method called the *Statistical Drift Detection Ensemble* (SDDE). This algorithm expands the work by Micevska et al. [18], in which the authors proposed the *Statistical Drift Detection Method* (SDDM), using statistical measures of concept drift defined by Webb et al. [10,20].

According to the taxonomies proposed by Lu et al. [27], and Hu et al. [44], two main groups of solutions can be distinguished among the supervised drift detection methods. The first one, relatively the most popular, bases the detection on the assessment of errors made by the classifier and thus empirically verifies the correctness of the currently used recognition model. The second, to which the method proposed in this paper should be assigned, departs from the performance evaluation and focuses on the analysis of statistical dependencies between the distant portions of data. Such portions should be large enough to be able to infer their distribution and, at the same time, small enough to make this inference valid and current.

The SDDE algorithm detection – as in the case of SDDM – is based on the statistical measures proposed by Webb et al. namely *Drift Magnitude* (DM) [10] and *Conditioned Marginal Covariate Drift* (CMCD) [20]. The first of these measures is defined by the distance between the concepts at time points t and u, defined as the distance between the features distributions $P(X)$ at these time points

$$DM_{t,u} = D(P_t(X), P_u(X)). \tag{1}$$

Any measure of the distance between the distributions can be used here, however, for the purposes of the SDDE algorithm, *Hellinger's distance* [45] was chosen which, for two discrete probability distributions $P = (p_1, \ldots, p_k)$ and $Q = (q_1, \ldots, q_k)$ is defined as follows

$$H(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{k} (\sqrt{p_i} - \sqrt{q_i})^2}. \tag{2}$$

It is worth mentioning here that SDDE uses *Drift Magnitude* as defined in Web et al. from 2016 [10]. In the study from 2017 [20], the *Total Drift Magnitude* (TDM) metric was proposed, which was based on the *Total Variation Distance* [21].

Conditioned Marginal Covariate Drift is defined as the weighted sum of the distances between the conditional probability distributions for the possible values of covariate attributes for each problem class $P(X|Y)$ between the time points t and u. The weights are the average probability of occurrence of a given class $P(Y)$ at both points in time

$$\sigma_{t,u}^{X|Y} = \sum_{y \in Y} \left[\frac{P_t(y) + P_u(y)}{2} \frac{1}{2} \sum_{\bar{x} \in X} |P_t(\bar{x}|y) - P_u(\bar{x}|y)| \right]. \tag{3}$$

The first version of the SDDE algorithm also used the *Posterior Drift* (PD) measure [20], proposed by Webb et al. This metric is defined similarly to CMCD but uses covariate distributions $P(X)$ and posterior distributions $P(Y|X)$

$$\sigma_{t,u}^{Y|X} = \sum_{\bar{x} \in X} \left[\frac{P_t(\bar{x}) + P_u(\bar{x})}{2} \frac{1}{2} \sum_{y \in Y} |P_t(y|\bar{x}) - P_u(y|\bar{x})| \right]. \tag{4}$$

However, after conducting preliminary experiments, *Posterior Drift* was dropped from the pool of metrics used due to the significant number of unjustified concept drift detections.

All the above-mentioned statistical measures are calculated based on the probability density distributions estimated using *Kernel Density Estimation* (KDE). The KDE implementation according to the *scikit-learn* library [46] with default hyperparameterization was used. *Kernel Density Estimation* is sensitive to the number of problem dimensions, and a large number of features can lead to performance degradation due to the curse of dimensionality. Therefore, SDDE estimates the densities of the probability distributions in the problem subspace set $\mathcal{S} = \{s_1, \ldots, s_{n_{max}}\}$, where n_{max} denotes the subspace number. Each problem subspace is created using sampling with replacement, and the *subspace_size* hyperparameter defines their size. At each given moment, we always keep two sets of KDE models. One trained on kth data chunk, denoted by $kernels_k$, and the second one built based on the chunk in which the concept drift was last detected (or on the first chunk of the stream), denoted by $kernels_b$. If a drift is detected, base kernels $kernels_b$ are replaced with current $kernels_k$. These sets correspond to the moments in time t and u, respectively.

Breaking down the problem into subspaces not only allows SDDE to avoid issues related to the large dimensionality but also allows obtaining an ensemble of detectors instead of a single one. In this case, the actual number of detectors corresponds to the number of subspaces multiplied by the number of statistical measures used, which for the proposed SDDE algorithm is equal to 2.

Due to the use of an ensemble approach to concept drift detection, the decision about drift occurrence must be considered on two levels: (i) each base detector and (ii) the entire ensemble. In the case of base detectors, the decision is made based on the difference comparison between the values of the DM_k and $CMCD_k$ statistical measures for the kth data chunk and the mean of the historical values harmonic mean – calculated over each of the base detectors for all previous data chunks – to the standard deviation of the harmonic mean, based on the three-sigma rule.

The principle of the SDDE algorithm is presented in detail by Algorithm 1 and additionally in Fig. 1.

The description of the functions used in the Algorithm 1 is as follows:

- PREPARE_SUBSPACES() – generates – using sampling with replacement – a set of subspaces with a cardinality defined by the n_{max} hyperparameter, where each subspace contains a number of problem features equal to the value of the *subspace_size* hyperparameter.
- FIT_KERNEL_DENSITY() — builds a set of KDE models on each of the subspace that allows estimating the density of probability distributions in the DS_k data chunk using the *Kernel Density Estimation* approach.

Algorithm 1 Pseudocode of the proposed SDDE algorithm.

Input:
$Stream = \{\mathcal{DS}_1, \mathcal{DS}_2, \ldots, \mathcal{DS}_k, \mathcal{DS}_{k+1}, \ldots\}$ – data stream,
σ – value used for a single detector threshold calculation,
sensitivity – value used for ensemble detection threshold calculation,
n_{max} – number of subspaces,
subspace_size – subspace size for each drift detector,
immobilizer – number of chunks skipped before the first detection.

Symbols:
$\mathcal{DS}_k$ – k-th data chunk.
$\mathcal{S} = \{s_1, \ldots, s_{n_{max}}\}$ – set of subspaces for each detector,
$kernels_k = \{\mathcal{KDE}^k_1, \ldots, \mathcal{KDE}^k_{n_{max}}\}$ – set of *KernelDensity* models for k-th data chunk,
$kernels_b = \{\mathcal{KDE}^b_1, \ldots, \mathcal{KDE}^b_{n_{max}}\}$ – set of base *KernelDensity* models,
PD_k – probability density estimated using $kernels_k$ on a k-th chunk,
PD^b_k – probability density estimated using $kernels_b$ on a k-th chunk.

Output:
drift – the list containing information about drift occurrence for each $\mathcal{DS}$.

$drift \leftarrow \varnothing$
$kernels_b \leftarrow \varnothing$
$\mathcal{S} =$ PREPARE_SUBSPACES($\mathcal{DS}_k$, *subspace_size*, n_{max})
$drf\_threshold = 2 * n_{max} * sensitivity$
for each $k, \mathcal{DS}_k = \{x^1_k, x^2_k, \ldots, x^N_k\}$ **in** *Stream* **do**
  $kernels_k \leftarrow \varnothing$
  for each s_i **in** $\mathcal{S} = \{s_1, \ldots, s_{n_{max}}\}$ **do**
    $kernels_k \leftarrow$ FIT_KERNEL_DENSITY($\mathcal{DS}_k, s_i$)
  end for
  if $k == 0$ **then**
    $kernels_b = kernels_k$
  end if
  $PD_k =$ ESTIMATE_DENSITY($\mathcal{DS}_k, kernels_k$)
  $PD^b_k =$ ESTIMATE_DENSITY($\mathcal{DS}_k, kernels^b_k$)
  $DM_k, CMCD_k =$ CALCULATE_METRICS(PD_k, PD^b_k)
  if $k >$ *immobilizer* **then**
    $detection\_count_k =$ GET_DETECTIONS($DM_k, CMCD_k, \sigma$)
    if $detection\_count_k >= drf\_threshold$ **then**
      $drift_k \leftarrow True$
      $kernels_b = kernels_k$
    else
      $drift_k \leftarrow False$
    end if
  else
    $drift_k \leftarrow False$
  end if
end for

- ESTIMATE_DENSITY() — estimates the density of the probability distributions in the DS_k data chunk for each s_i subspace, using a set of previously trained *Kernel Density Estimation* models found in *kernels*.
- CALCULATE_METRICS() — calculates the values of *Drift Magnitude* and *Conditioned Marginal Covariate Drift* based on the density of probability distributions estimated using Kernel Density models built on kth data chunk and those stored from the beginning of the stream or the last detection of the concept drift occurrence.
- GET_DETECTIONS() — determines, based on the historical values of the statistical metrics, what number of base detectors identified a concept drift in the kth data chunk. The final decision to detect the drift occurrence is based on the current DM and CMCD values and the harmonic mean of the historical values over the base detectors. The sigma parameter defines to which multiple of the standard deviation of the harmonic mean the difference between the current metrics and the averaged harmonic mean is compared to indicate a concept drift occurrence.

In summary, the SDDE method builds a fixed-size pool of detectors, basing the recognition on the distribution density function of consecutive processing chunks. When the detection threshold is exceeded, i.e., a significant change between reference ($kernels_b$) and current distribution ($kernels_k$) is recognized by the ensemble, the estimated reference distributions are exchanged in each of its members, which allows updating the knowledge about the current concept. The critical element here is the batch exchange of base detectors, which are not updated at the time of individual recognition of a concept change but only after reaching consensus within the ensemble.

A simplified example of the SDDE operation is presented in Fig. 2. It shows processing on a flow of thirteen chunks ($\mathcal{DS}_1$–$\mathcal{DS}_{13}$), where the ensemble is built on six random subspaces (s_1–s_6), assuming an *immobilizer* of 2 and a *drift threshold* of 50 percent. The null chunk – where the initial reference distributions are built – is omitted from the example. Each illustration cell represents the response of a pair of detectors (*DM* and *CMCD* metrics) built on the same subspace (s_n).

Ten detections can be observed in the first chunk of processing, which is 83 percent of all detectors. However, this does not lead to a drift detection as the model is then in an immobilized state (blue area). In chunks 3 and 5, the drift is signaled by four detectors. However, it is only 33 percent of the available pool, so the ensemble is not yet reporting the drift, and each detector remains in its original state. In the sixth chunk, the critical mass of 9 detections is reached, which exceeds the *drift threshold* and leads to the recognition of a change in the concept. Each of the detectors is then updated to the distributions obtained on the $\mathcal{DS}_6$ chunk subspaces, and the meta-estimator receives information about the need to rebuild the classification model.

The presented method is – by definition – based on batch processing due to the need to estimate the probability density distributions on the given window. However, it can be adapted for online learning by replacing disjoint data chunks with a sliding window, typical for prequential analysis. This approach will allow the data stream to be processed at the level of individual problem instances while maintaining the memory of a predefined or dynamically selected number of previous samples. Such a solution will enable the estimation of the probability density distributions for online learning. However, it will be associated with a significant increase in the time complexity due to performing the drift detection after the arrival of each problem instance.

4. Experimental set-up

The following section will describe goals of the planed experiments and experimental setup of the conducted research. All experiments and methods have been implemented in *Python* programming language using the *scikit-learn* [46], *stream-learn* [47], *scikit-multiflow* [48] and *numpy* [49] libraries. The base classifier of the *meta-estimator* 3.1, to simplify calculations for extensive computer experiments, was *Gaussian Naïve Bayes Classifier*.

In experiments, both classification accuracy and chunk indexes of detection were collected in order to enable the calculation of *drift detection errors* 3.2.

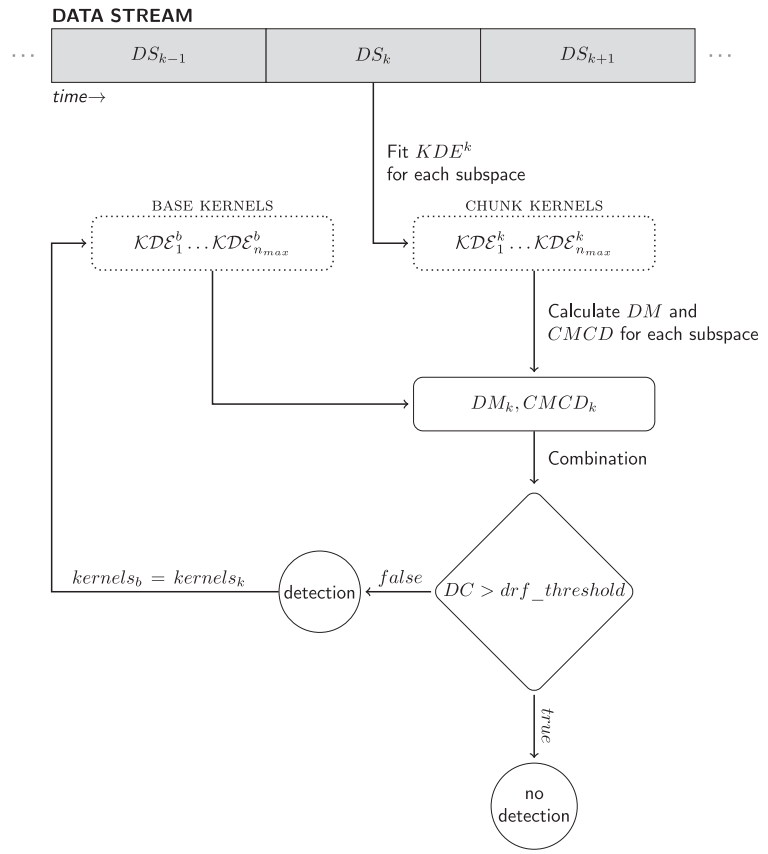


Fig. 1. The principle of SDDE processing.

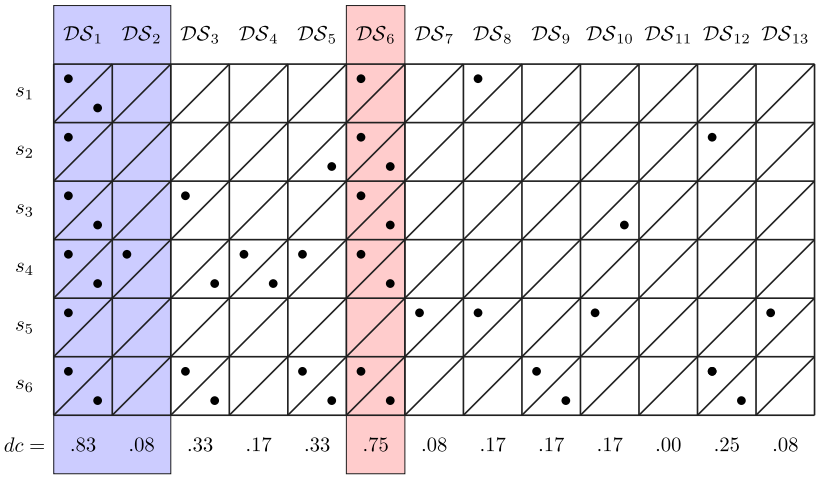


Fig. 2. The example of SDDE processing.

4.1. Experiment 1 — method optimization

The aim of the first experiment was to optimize the SDDE method's hyperparameters such as *sensitivity*, *subspace size* and the *number of detectors*. The experiment was carried out on 30 binary, balanced data streams with five non-recurring concept drifts. Three different concept drift types were taken into consideration during generation of data streams: *sudden*, *incremental* and *gradual*. The streams for hyperparameter optimization were divided into 100 data chunks, each containing 200 instances. Instances were described by 15 informative features and contained label noise at 1%. Each stream was replicated ten times with a different random state. For the purpose of this experiment synthetic data streams were generated by *StreamGenerator* from a *stream-learn* package [47].

In order to optimize the method, the following hyperparameter values were considered:

- *subspace size* containing 1, 2 or 3 random features from initial feature space;
  Research in [18] has shown that *drift magnitude* measures provide the most information for low dimensional spaces, hence low *subspace size* values were evaluated.
- *detector's sensitivity* containing 20 values form 0% to 100%, sampled from linear space;

- The *sensitivity* parameter specifies the fraction of detectors in the ensemble that must detect drift for it to be considered as an integrated decision. The threshold parameter set to zero is the equivalent of stable, deterministic drift detection on every data chunk, even when none of the detectors indicates it. Likewise, a 100 percent threshold requires all detectors to be acclaimed, making the most strict integration rule. Examining the full range of parameter values will allow for a detailed analysis of this parameter effect on an algorithm.
- *number of detectors* containing 20 values from 1 detector to 100, sampled from quadratic function.
- The increase of the *number of detectors* may potentially increase the quality of detection but as well entails an increase in memory and computational complexity. For the evaluated streams containing 15 features, increasing the number of detectors increases the chance of analyzing all features during processing. For the algorithm's usability, we are looking for the smallest parameter value that gives satisfactory results.

4.2. Experiment 2 — comparison with the reference methods

The second experiment aimed to compare the proposed SDDE method with reference drift detection algorithms. For this purpose, the following detectors have been implemented into the programming interface of *stream-learn* library:

METHOD		HYPERPARAMETERS	CITE
DDM	*Drift Detection Method*	warning level = 2.0; drift level = 3.0; skip= 30	[13]
EDDM	*Early Drift Detection Method*	warning level = 0.95; drift level = 0.9;	[14]
ADWIN	*Adaptive Windowing*	$\delta = .002$	[15]
$HDDM_A$	*Hoeffding Drift Detection Method* with *Bounding Moving Averages*	drift level = 0.001; warning level = 0.005	[17]
$HDDM_W$	*Hoeffding Drift Detection Method* with *Bounding Weighted Moving Averages*	warning level = 0.005; drift level = 0.001; $\lambda = .05$	[17]

A total of 320 binary, balanced data streams were generated to carry out a second experiment. The streams were characterized by 3, 5, or 7 concept drifts in 200 data chunks, each containing 250 instances. The instances were described by 10, 15, or 20 informative features with label noise of 1%. The drifts were both recurring and non-recurring of the following types: *sudden*, *incremental* and *gradual*. Each of the streams was replicated ten times with a different random state. Streams were as well generated by *stream-learn* [47] package.

The proposed method – SDDE was parameterized by *sensitivity*, *number of detectors* and *subspace size* selected during analysis of Experiment 1.

4.3. Experiment 3 — real-concept data streams analysis

The purpose of the third experiment was to evaluate the performance of SDDE on streams generated from real concepts using *random projection-based concept drift injector* proposed by Komorniczak et al. [50]. The method is converting real static datasets to data streams with concept drifts of *nearest* and *cubic* types, which correspond to *sudden* and *incremental* drifts. Generator code is publicly available on *GitHub* repository.[2]

Table 1
Original number of instances and features of real-world datasets used for generating data streams.

Dataset	Samples	Features
australian	690	14
banknote	1 372	4
diabetes	768	8
wisconsin	699	9

Utilizing a generator converting real static data to data streams with concept drifts will enable calculating moments of concept change, therefore comparing detections with actual drifts. The available real-world data streams do not contain ground-truth in the context of concept drifts as well may contain drifts of mixed type [27]. The reasons mentioned above contributed to the decision to use streams generated based on real data, ensuring the moment of occurrence and type of drift instead of the original real-world data streams.

Four semi-synthetic data streams were generated based on real-world concepts. Each data stream was described by 15 attributes originating from actual static dataset features. Streams were characterized by 3, 5, and 7 drifts throughout 400 chunks of 250 instances each. The original datasets are described in Table 1.

All detection methods from *Experiment 2* were used to detect drift in the evaluated streams. Two additional pseudo-detectors, designed as the baselines for the purpose of experimental evaluation, were:

ALWAYS – method detecting drift in each chunk;
NEVER – method that never detects drift.

The *sensitivity* parameter was recalibrated for the optimal algorithm's performance of the semi-synthetic streams based on the real-world concepts.

5. Experimental evaluation

This section presents the results and critical analysis of the conducted experimental evaluation, which consisted of experiments looking for (E1) the optimal hyperparameterization of the SDDE method and the assessment of its effectiveness in the comparison with *state-of-art* methods in the context of data streams build on (*E2*) synthetic and (*E3*) based on real concepts.

5.1. Experiment 1

The first experiment aimed to select the values of the SDDE method hyperparameters for further processing, maximizing the metrics proposed in Section 3.2 for separated, overview data streams.

Figs. 3 and 4 show the context of optimization constituting a visualization that allows the analysis of the influence of hyperparameters on the values of quality assessment metrics. For the selected *size of the subspace*, it presents visualizations of heat maps showing the dependence of the six assessment criteria in the two-dimensional function of (a) the number of models in the detector ensemble pool and (b) the sensitivity of their integration function.

Fig. 3 shows the three base assessment metrics defined in Section 3.2: $D1$ – closest drift, $D2$ – closest detection and R – the ratio between the number of detections and drifts. It should be noted that while R is the dominant factor here, clearly informing about

[2] https://github.com/w4k2/ip_stream_generator/blob/master/generator.py.

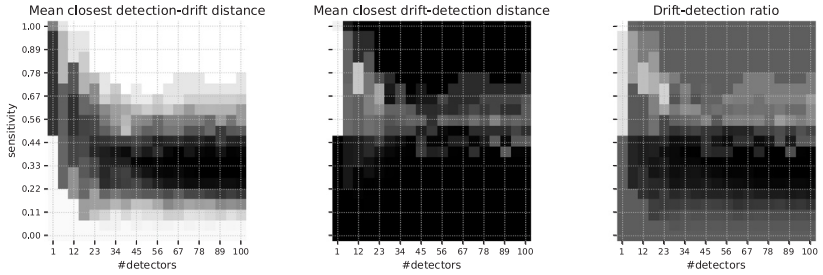


Fig. 3. Exemplary optimization results of proposed method. Number of detectors and sensitivity influence on model's drift detection errors.

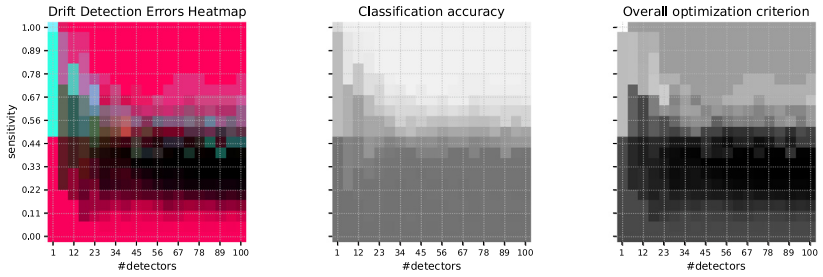


Fig. 4. Exemplary optimization results of proposed method. Aggregated drift detection errors (left), classification accuracy (center) as well as overall optimization criterion (right) consisting of combination of classification accuracy and drift detection errors.

possible indifference or hyperactivity of the detector ensemble, it is possible to achieve its optimal value by random detections with prior knowledge of the number of drifts in the stream. Therefore, the combined knowledge of many factors is necessary for the appropriate selection of suboptimal hyperparameters.

To enable the calculation of *Drift Detection error* measures in cases with the absence of drift detection for a given configuration, the detection was assumed in each data portion. In a detection task, an algorithm that is unable to detect drift carries as little information about the data as a method that detects a change of concept in each batch of data.

The combination of these factors is presented in Fig. 4. Its first cell is a color combination of RGB channels defined by the base metrics, where white represents the globally worst configuration and the most saturated colors magenta and cyan — the worst configurations according to the D_1 and D_2 criteria. Similarly, the region of the best and statistically dependent on the best configurations is represented by the black area in the illustration. The second heatmap shows the classification accuracy heat map, which is most often the primary criterion for evaluating recognition models. The last heatmap of the Figure shows the averaging of the integrated base metrics with the classification quality, constituting the basic tool for selecting hyperparameters used in this work.

In presenting the above-described visualization tool as an overview context of the drift type and size of the analyzed subspace, it is necessary to start with the assessment of the overall accuracy of models built with a specific detection strategy. The observation of Fig. 5 (left) gives a clear and intuitive observation showing that the ALWAYS rule – rebuilding the model at each batch, identical to high sensitivity SDDE detection – is optimal in terms of recognition quality. From the computational perspective, however, it is the worst and the slowest strategy, so it is necessary to select a method configuration in the bright region of the heat map. At the same time, it should not lead to redundant detection — increasing the time complexity of stream processing. Stabilization of measurements is noticeable here after reaching a sufficiently large number of detectors, the number of which strongly depends on the size of the subspace. The most stable in this context, as confirmed by the observations from [18], seems to be an independent analysis of the features, stabilizing with the number of detectors equal to the number of attributes, narrowing down the search for the optimal value of the sensitivity parameter to the range of 30%–60%.

Responses from the remaining metrics (right side of Fig. 5) confirm the observations about the stability of detection after reaching a certain number of detectors. The most straightforward and most unequivocal interpretation here is the ability to detect in the case of sudden drift, where the ground truth perfectly coincides with the point where the drift occurs, leading to a readable, broad black band representing the optimum of combined D_1, D_2 and R metrics.

In gradual and incremental drift – due to dynamics of changes – the black area is absent, replaced by an area with reduced saturation in the range of 25%–60% sensitivity. Interestingly, there is a much smaller correlation between the D_1 and D_2 metrics in the case of incremental drift, which leads to a clear magenta region which shows that we get good results in one of these metrics. However, we have to reject them due to the unsatisfactory results of the other. This suggests that the optimum in the analysis of metrics other than accuracy should be slightly less than 50% of sensitivity.

The observations made on the separate analysis of the accuracy and the proposed metrics seem to be confirmed in the case of their combination visible in Fig. 6. Between the black area of low-quality classification and the gray area of the average quality of detection, there is a narrow area of increased intensity of the heat map, in which we should look for the optimum sensitivity hyperparameter. However, strong disturbances in the read value are also clearly visible here, which may indicate that it is not possible to select one global sensitivity value, which is the optimal SDDE configuration. Such observation suggests that this parameter depends on the specific problem and should be selected individually for each data stream.

The selected basic driver in the optimization of hyperparameters for further processing was the minimization of the identified R parameter corresponding to the ratio between the number of drifts and detections while maintaining a low value of the auxiliary metrics $D1$ and $D2$, responsible for the mutual distances between the drifts and detections. A review of Fig. 7, showing

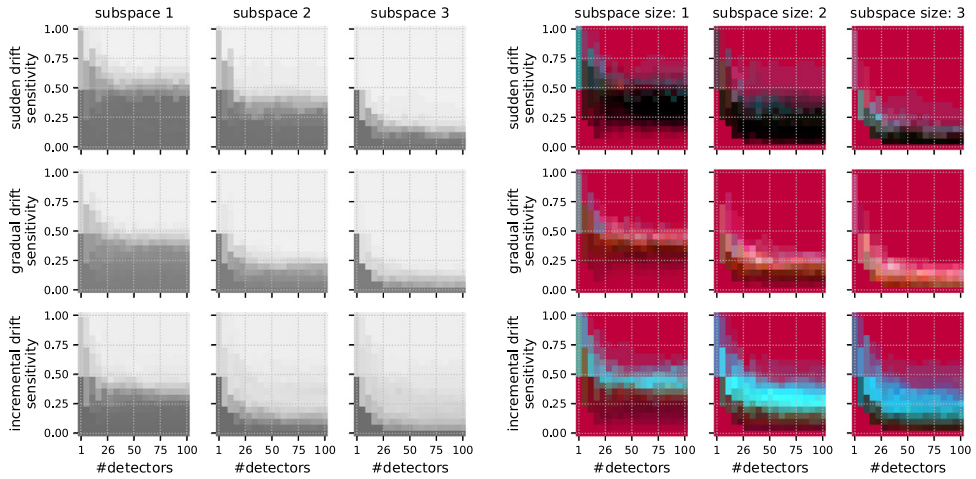


Fig. 5. Classification accuracy (left) and drift detection errors (right) for tested subspace sizes and drift types.

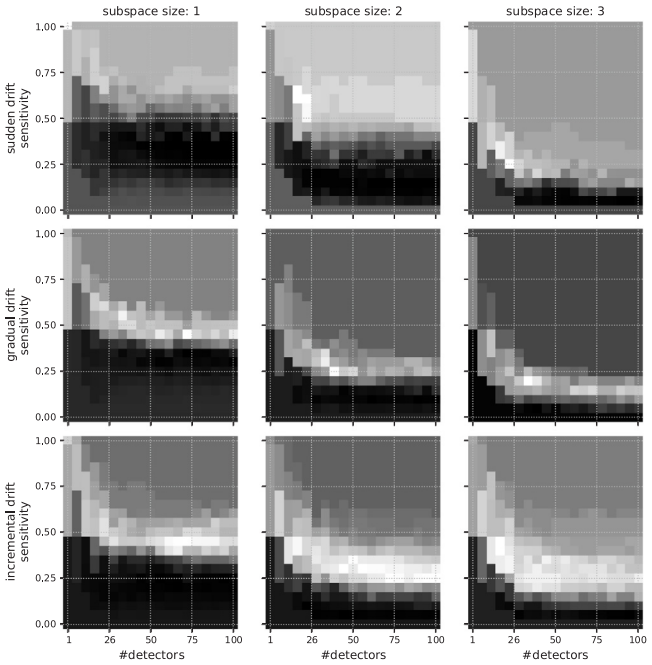


Fig. 6. Overall optimization of hyperparameters for tested subspace sizes and drift types.

the cumulative mean of the metrics as a function of detector sensitivity, made it possible to decide on a sensitivity value of 45% in the case of independent analysis of features. However, it should be noted that this choice was made on the example of synthetic streams and applies only to Experiment 2. In the case of Experiment 3, an additional calibration was made, which allowed for the selection of sensitivity of 35% as better suited to the characteristics of real concepts.

The rationale for such decisions may be found in Figs. 8 and 9, which show a time course of SDDE detections for different sensitivity values between 30%–60% in averaging ten replications. It is observable here that the value of the sensitivity parameter for synthetic streams giving almost perfect results for sudden drift (50%) is not appropriate for the other types of drift. A high value of this parameter leads to desensitization of the detector to drift and too low — to redundant detections. Therefore, the selected values of this hyperparameter represent a compromise that maximizes the ability to recognize concept changes over different classes of drift.

For streams based on real-world concepts, the conclusions remain similar. However, the optimal values of the parameter change. The *nearest* drifts, equivalent to *sudden* drifts, give satisfactory results with the tested parameter values below 45%. For *cubic* drifts, corresponding to *incremental* changes of the concept, the optimal value of the parameter is smaller, around 30 to 35%. Additionally, differences between specific data streams can be noticed. Depending on the static datasets based on which the streaming data was generated, the confidence of drift detection varies, i.e. the same sensitivity parameter value shows the different number of detections on different streams.

5.2. Experiment 2

The second experiment aimed to conduct a comprehensive overview analysis of drift detectors' ability to raise concept change alerts using examples of a variety of fully synthetic data streams. While – as is often mentioned in reviews of works in the field of data stream processing – the use of completely synthetic data is not a good strategy for the final evaluation of recognition methods and a sufficient contribution to their recommendation or rejection from practical applications, this approach allows for a clear revision of the characteristics behavior of recognition algorithms. Only in such an environment of problems is it possible to replicate streams with identical properties. The achieved results no longer depend on the difficulties of the concepts described by the data and present a possibly objectified comparison of the analyzed methods. Therefore, the results of the second experiment are presented in the paper by visualizing the location of alerts in the course of streams differing in (*i*) the number of attributes, (*ii*) drifts, and (*iii*) the type of drift of the concept.

Fig. 10 presents an exemplary visualization of experiment results. Each horizontal block shows drift detections given by consecutive methods over time, indicated by concept drifts marked on *X* axis. Successive rows within blocks corresponding to algorithms mean successive replications of streams with given characteristics, distinguishing accidental warnings from clearly visible trends. The detections of the SDDE algorithm, proposed for this paper, are marked by a yellow row. Each of the chunks in which the given detector indicated the occurrence of drift is marked with a black point in the illustration.

Fig. 11 (on the left side) shows the drift alarms for non-recurring concepts. The first visible observation in the case of this analysis is the apparent hyperactivity of the EDDM detector, leading to redundant alerts present on the entire course of

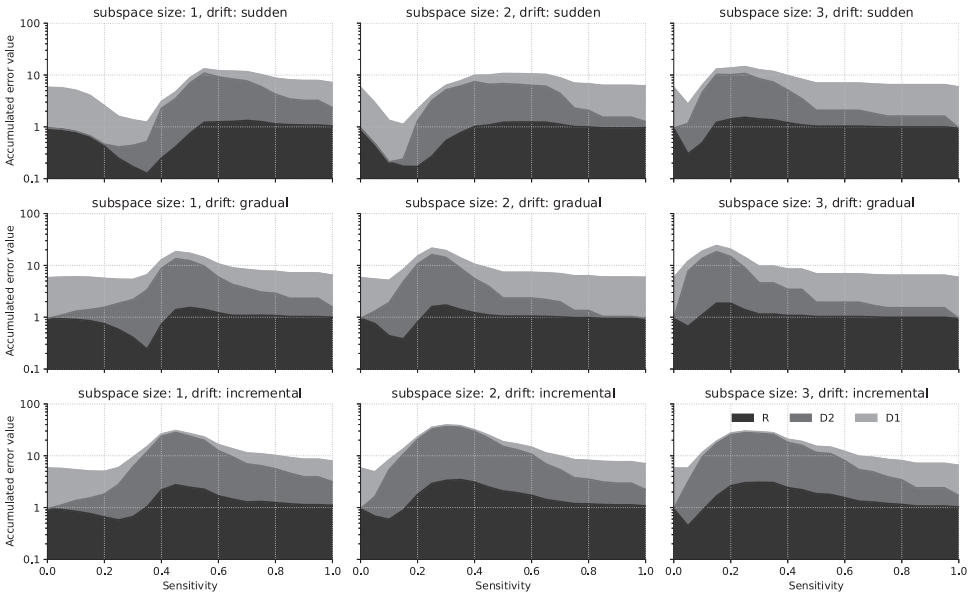


Fig. 7. Mean accumulated Drift Detection Errors (drift detection ratio, closest detection, closest drift) values for all tested number of detectors. In columns – subspace size, in rows – drift type.

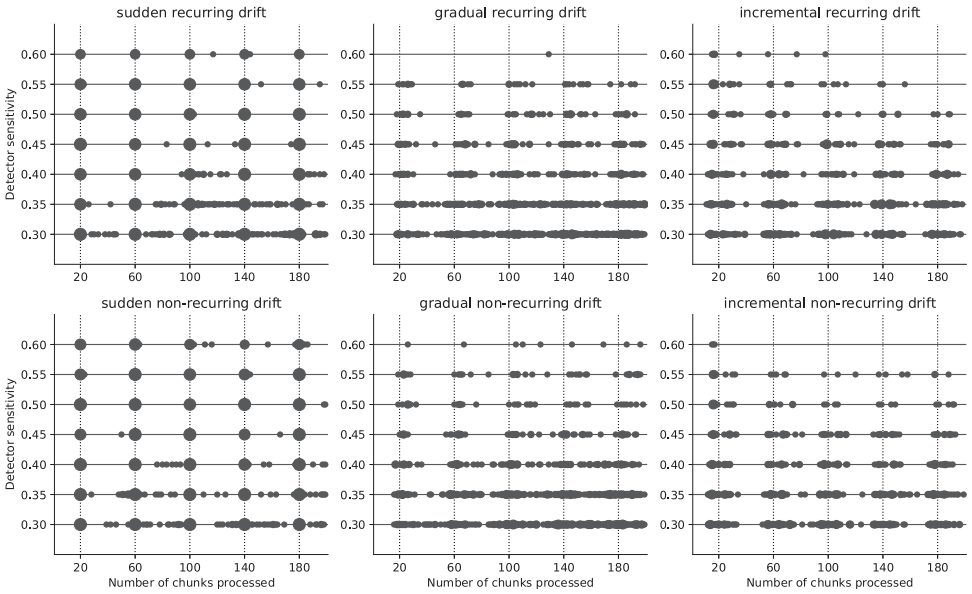


Fig. 8. Method sensitivity impact on detections number in 10 replications for synthetic streams.

the streams. There is indeed a decrease in their density at the moments of stable concepts (evident with sudden drift), but it is still a method that definitely too often indicates a change of concept. The DDM method behaves similarly, especially in the initial phase of streams. In many cases, it either quickly becomes desensitized to changes, ceasing to be useful in recognizing, or leads to continuous alerts that begin with the emergence of a new concept and finish little before the next one appears.

Interestingly, such an approach allows for a seemingly high-quality classification, but a similar one would be achieved in many cases by the abstract ALWAYS detector. Based on the conducted analysis, it seems that the wrong approach – often indicating the high usefulness of EDDM and DDM methods – is to base these conclusions on the high overall quality of recognition. It may result mainly from a frequent reconstruction of the recognition model, which in the presented strategy of updating models in chunks in which no detection takes place, allows them to gain an advantage in the overall quality of recognition, but at an apparent cost of the high time complexity.

The results for the ADWIN and SDDE methods present much better — in the case of sudden drifts, the detections often cover the line of occurrence of the drift almost perfectly. In the case of gradual drifts, they also behave quite similarly – however – spreading the detection points wider over the entire course of the ongoing change. It is sometimes alerting several times then, highlighting the transition phases of the drift and, at the same time, allowing for detection adequately earlier than the central drift point. Apparent differences between SDDE and ADWIN appear only in the case of incremental drifts, where the standard deviation of the detection distance from the drift is higher for SDDE while maintaining a uniform distribution around the central drift point, which can be interpreted as a higher ability of the proposition presented in this work to signal a drift early.

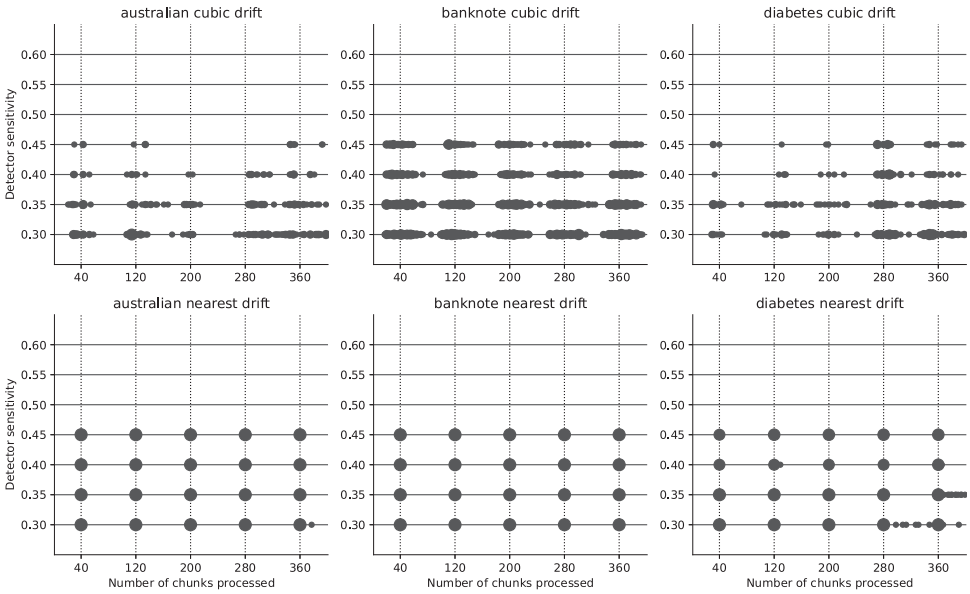


Fig. 9. Method's sensitivity impact on detections number in 10 replications for streams based on the real-world concepts.

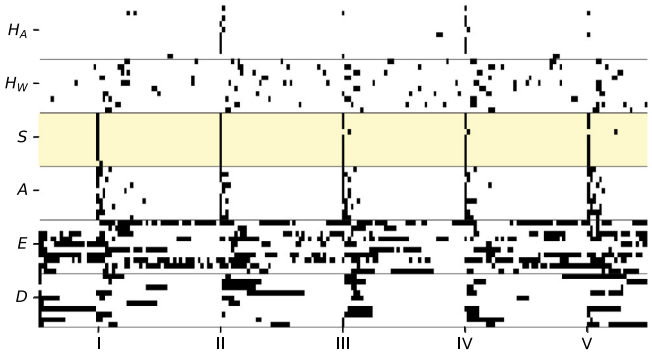


Fig. 10. Exemplary result of Experiment 2 — detection moments of evaluated methods for 10 replications of stream with 5 concept drifts.

The HDDM$_A$ method recognizes drifts only in a few of the replications, especially in the case of streams characterized by three drifts. Additionally, in case of recurring sudden concept changes, only some of the actual ones are signaled. Generally, the method presents low sensitivity to concept drifts. The HDDM$_W$ detector, however, is showing frequent detections, which rarely lay in the exact moment of drift but are rather signaled after a certain number of chunks describing a new concept. The results for recurrent drifts, included in Fig. 11 (on the right side), are similar. The observations seem constant, regardless of the problem's dimensionality and the number of drifts.

5.3. Experiment 3

The final experiment aimed to test the operation of the proposed method and to compare it with other drift detection methods on data streams based on real-world concepts.

The results of the experiment are presented in Figs. 12 and 13. The proposition of this paper is highlighted in a yellow color. A single black point on the plot indicates a concept drift detection in one of the repetitions of the experiment. Each of the replications of analyzed methods (DDM, EDDM, ADWIN, SDDE, HDDM$_A$, HDDM$_W$) is presented in the figure.

In every replication, the reference methods mark drifts at the same moments. Since their behavior depends only on the classification quality, the detectors are deterministic. While evaluating one stable stream, the classification quality of the deterministic *Gaussian Naive Bayes classifier* used for evaluation will be as well stable in each replication. Thus drifts will be identified at the same moment.

The proposed SDDE method does not use the classification quality during detection and has an element of randomness — the selection of a subset of the analyzed features. Experiment 1 showed that the algorithm performs best with a *subspace size* parameter set to one. Despite setting the *number of detectors* in ensemble equal to the number of features in the stream, it is not required that each attribute will be used for analysis. Duplicate features may appear as a result of randomization. For the reasons mentioned above, the SDDE method can mark detections in various moments on the same stream, depending on the selected features taken into account during the calculations. The aforementioned situation is also seen in Figs. 12 and 13.

The static datasets, based on which the streams are generated, are characterized by different difficulties in the context of the classification task. It can be noticed that the data difficulty differs in the context of concept drift detection. In the case of the *banknote* and *wisconsin* datasets, the detections of all methods occur more often than in the case of *australian* and *diabetes* datasets.

The results show that the proposed method has the potential to make more accurate detections than the reference methods. Since the drifts of type *nearest*, which is the equivalent of *sudden* type, are occurring at an unequivocal moment, the method detected accurately on almost every set tested and in every

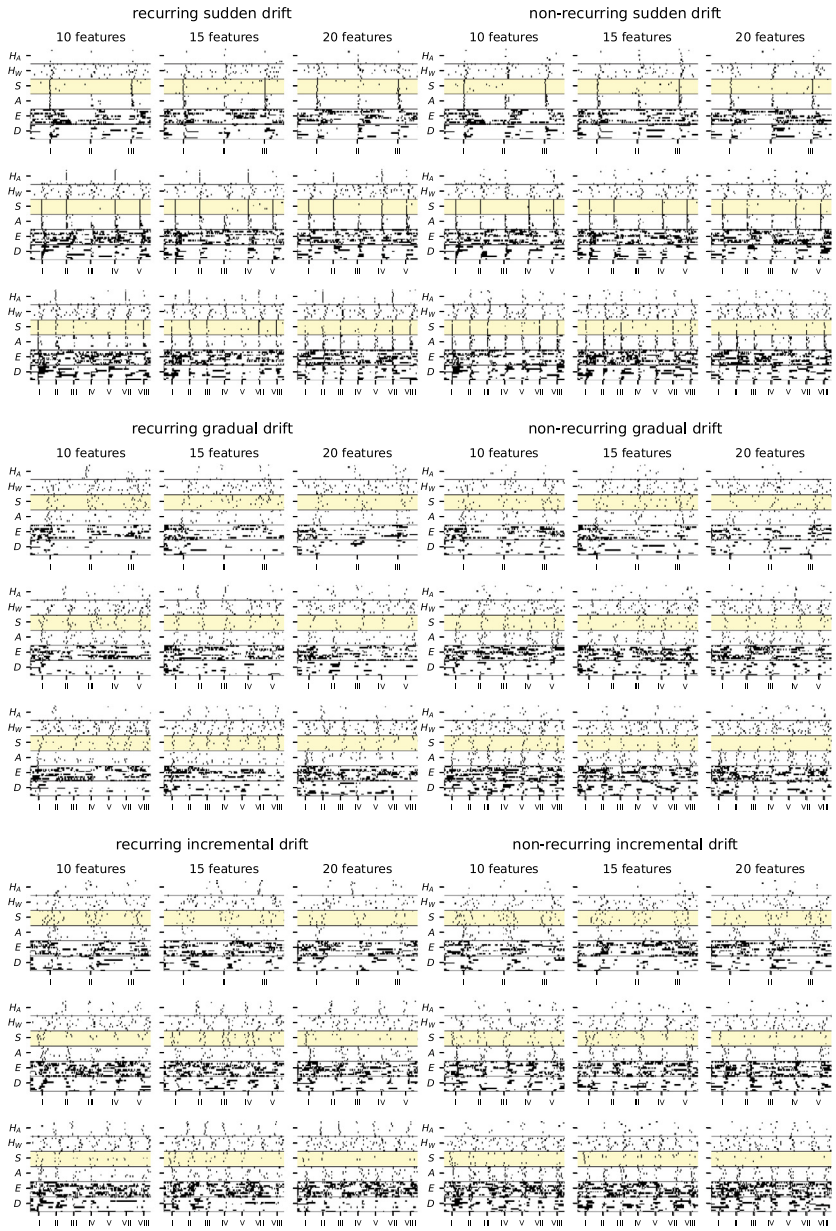


Fig. 11. Drifts detected by evaluated methods on streams with recurring (left) and non-recurring concept drift (right).

replication. Only sporadic false-positive errors appeared. On the other hand, for *cubic* drifts, corresponding to *incremental* type, we cannot determine the precise moment of drift. In the case of these drifts, reference methods often fail. Results for the proposed SDDE detector show a visible accumulation of detections in the vicinity of the drift.

Moreover, one drift can be detected several times, as the concept changes in *incremental* drifts are fluid and long-lasting. The method is also sensitive to the detection of transient concepts between the target ones and is capable of detecting an early phase of incremental drifts. This is, however, an advantageous effect. When the classification model is rebuilt to reduce the loss of recognition quality in standard detector applications, the quality will also be maintained throughout the drift duration. By rebuilding the model once, for instance, at the beginning of the *incremental* drift occurrence, the classifier would also be trained using the transition patterns lying between the two concepts, which could potentially affect classification accuracy.

The results of the comparison are also presented in Tables 2–5. The statistically best performing method was emphasized in the results. The tables are followed by *Critical Difference* diagrams with ranks obtained with Nemenyi post-hoc statistical test.

Classification results (Table 2) present all evaluated detectors and both extreme ALWAYS and NEVER methods. However, the pseudo-detector NEVER was omitted in the comparison of the drift detection quality metrics (Tables 3–5). The lack of detection makes it impossible to calculate the desired distances (D_1, D_2) and as well the proportion of drift number and detection number (R). We can consider that a method not detecting a change of concept is unprofitable and its drift detection errors will be infinite.

In terms of classification quality, the results are similar for all streams except for the streams originating from the *australian* dataset. The best results are achieved by the ADWIN, SDDE and ALWAYS methods. The NEVER method achieves the worst classification accuracy, which is due to the lack of adaptation of the base

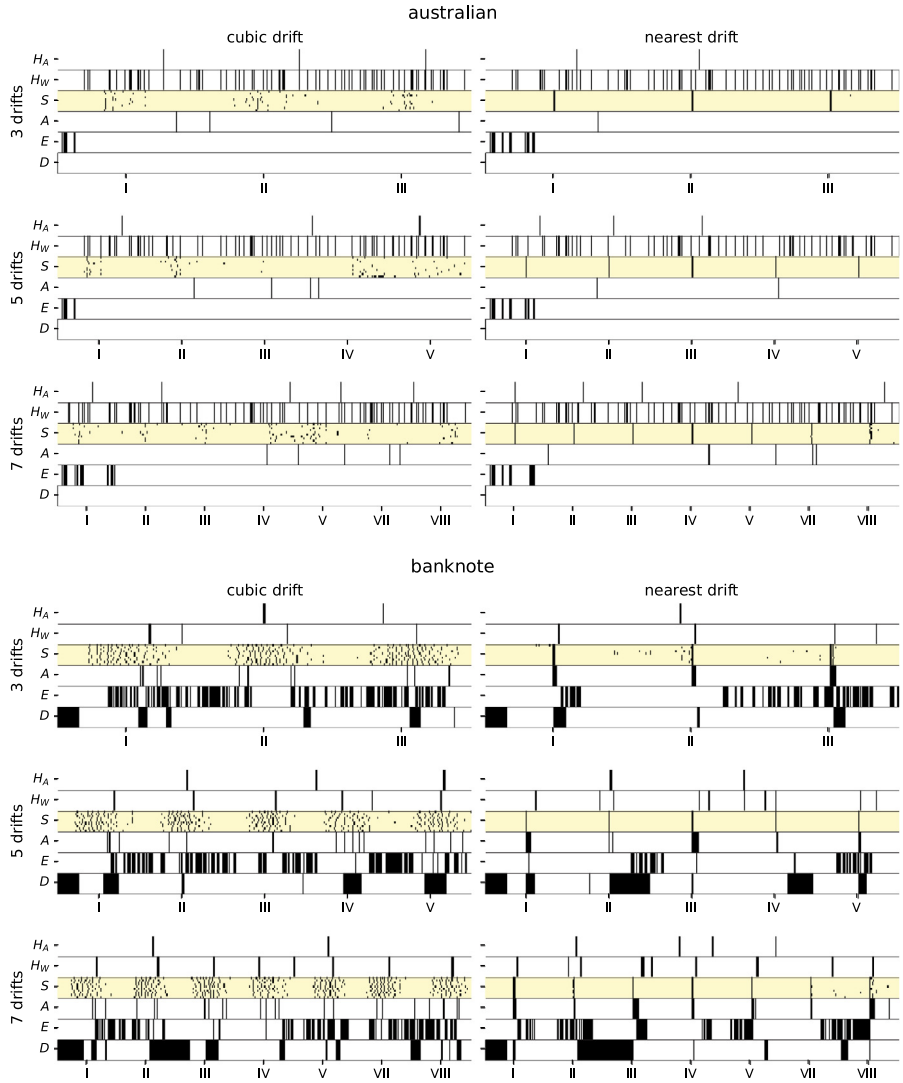


Fig. 12. Moments of detections of evaluated concept drift detection methods on real-concept data streams (*australian* and *banknote*).

classifier to changes in the concept throughout stream flow. The fact that a pseudo-detector often obtains the best results confirms the observations from the publication [26], in which a method that does not analyze data but only deterministically signals a drift every given number of patterns is able to achieve statistically better results than methods dedicated to the analysis of concept changes. Therefore confirms the above-mentioned work's conclusion that classification accuracy measures should not be used to measure drift detection effectiveness.

The *australian* set seems to be particularly difficult in terms of the classification task because the patterns derived from one concept are characterized by much noise. Only for this data stream, the best results are achieved by the detectors EDDM (for *cubic* drift) and DDM (for *nearest* drift). Both detectors, which can be noticed in the first rows of Fig. 12, detected the drifts only at the beginning of the stream analysis.

Table 3 shows the results of the D_1 (*closest drift*) measure, which indicates the average distance of each detection to the nearest drift. For most of the analyzed methods, the error values are higher for *cubic* type of drift changes. The time period of drift occurrence is wider and the exact moment of detection depends on the sensitivity of the method to concept changes. In the case of *nearest* drift types, the SDDE method for streams characterized with 5 drifts achieved ideal results — the error value is exactly zero. Statistically, the best results of this evaluation criterion in most of evaluated streams were achieved by the proposed SDDE method. The worst results were achieved by the DDM, EDDM and ALWAYS methods.

Table 4 shows the results of the D_2 (*closest detection*) measure which describes the average distance of each drift from the closest detection. The ALWAYS pseudo-detector will have zero error value in the context of this evaluation criterion. By signaling the drift in each data chunk, detection will certainly be signaled at the moment of the actual concept drift. Redundant detections are not taken into account in this evaluation measure. As with the measure D_1, the error values are often larger with *cubic* drift. This is for the same reason — the detections will be more spread over time compared to the *nearest* drift. The SDDE method achieved zero error values for the *nearest* type of drift and streams containing 5 concept changes. When disregarding the ALWAYS method from the comparison, SDDE achieves the statistically best results in the case of most of the streams with *nearest* drift. In the case of *cubic* drift, in the data stream originating from the *diabetes* dataset, the method does not perform well, and the best results are presented by HDDM$_W$ method.

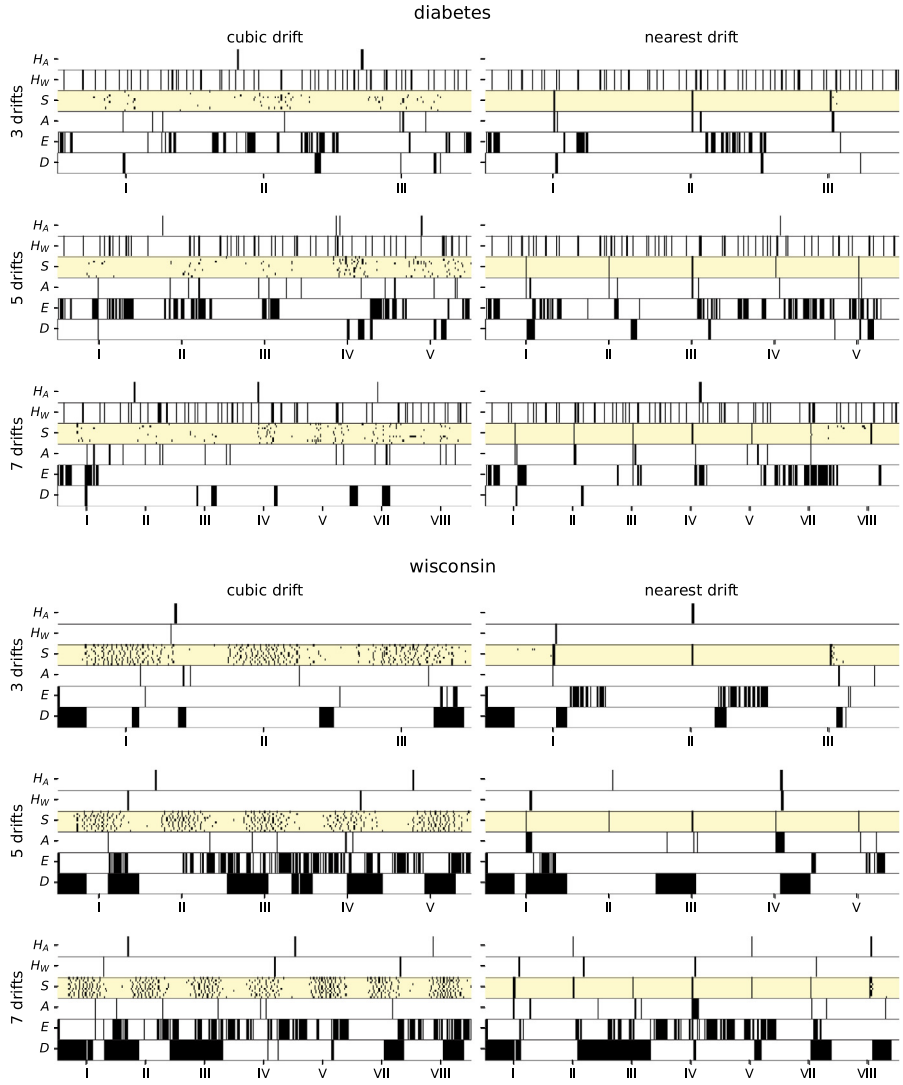


Fig. 13. Moments of detections of evaluated concept drift detection methods on real-concept data streams (*diabetes* and *wisconsin*).

Table 2
Classification accuracy of data stream processing.

		CUBIC								NEAREST							
		D	*E*	*A*	*S*	H_W	H_A	*a*	*n*	*D*	*E*	*A*	*S*	H_W	H_A	*a*	*n*
		1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
AUSTRALIAN	3	.613 3,5:8	**.639** *all*	.598 5:8	.611 3,5:8	.590 7	.596 5,7:8	.588 –	.594 5,7	.621 3:8	**.624** *all*	.620 4:8	.585 –	.590 4,7	.604 4:5,7	.589 4	.615 4:7
	5	.617 3,5:8	**.649** *all*	.608 5:8	.618 3,5:8	.589 7:8	.603 5,7:8	.589 8	.573 –	**.619** *all*	.610 4:8	.616 2,4:8	.587 –	.592 4,7	.598 4:5,7	.589 4	.607 4:7
	7	.622 3:8	**.631** *all*	.596 5,7:8	.612 3,5:8	.595 7:8	.599 3,5,7:8	.589 8	.587 –	**.620** *all*	.617 3:8	.603 4:7	.589 –	.590 4,7	.599 4:5,7	.589 –	.611 3:7
BANKNOTE	3	.859 6,8	.880 1,5:6,8	**.890** *all*	.880 1,5:6,8	.868 1,6,8	.653 8	.883 1,2,4:6,8	.490 –	.856 2,5:6,8	.830 6,8	**.875** *all*	.869 1:2,5:8	.845 2,6,8	.542 8	.859 1:2,5:6,8	.471 –
	5	.853 6,8	.861 1,5:6,8	**.875** *all*	.866 1:2,5:6,8	.859 1,6,8	.648 8	.868 1:2,5:6,8	.479 –	.825 2,5:6,8	.799 6,8	**.850** *all*	.839 1:2,5:6,8	.804 2,6,8	.601 8	.841 1:2,4:6,8	.475 –
	7	.810 6,8	.861 1,5:6,8	**.876** *all*	.871 1:2,5:6,8	.822 1,6,8	.592 8	.875 1:2,4:6,8	.457 –	.787 6,8	.819 1,5:6,8	.843 1:2,5:6,8	.848 1:3,5:6,8	.809 1,6,8	.615 8	**.854** *all*	.465 –

(*continued on next page*)

As the number of drifts in the stream increases, the maximum error values of the D_1 and D_2 measures decrease. The mentioned measures are based on the distance between the actual drift and the detection. In case drifts occur more frequently, the reduction of the distance between detection and drift is a consequence. Therefore, drift detection methods should be compared under the same conditions, i.e., the same number and location of drifts in the stream.

Table 2 (*continued*).

		CUBIC								NEAREST							
		D	*E*	*A*	*S*	H_W	H_A	*a*	*n*	*D*	*E*	*A*	*S*	H_W	H_A	*a*	*n*
		1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
DIABETES	3	.594 6,8	.597 1,6:8	**.606** *all*	.592 6,8	.599 1:2,6:8	.553 8	.596 1,6,8	.550 –	.571 6,8	.594 1,3,5:6,8	.589 1,5:6,8	**.602** *all*	.588 1,6,8	.531 –	.595 1:3,5:6,8	.531 –
	5	.578 6,8	**.603** *all*	.600 1,4:8	.582 6,8	.595 1,6,8	.556 8	.597 1,4:6,8	.547 –	.586 6,8	.592 1,3,5:6,8	.591 1,6,8	.597 1:3,5:6,8	.592 1,3,6,8	.543 8	**.597** *all*	.528 –
	7	.580 2,6,8	.579 6,8	.584 1:2,4,6,8	.576 6,8	.595 1:4,6,8	.547 8	**.596** *all*	.541 –	.541 6,8	.595 1,3:6,8	.593 1,4:6,8	.588 1,6,8	.592 1,4,6,8	.529 8	**.597** *all*	.521 –
WISCONSIN	3	.959 2:3,5:6,8	.956 5:6,8	.957 2,5:6,8	**.960** 1:3,5:6,8	.948 6,8	.935 8	**.960** 1:3,5:6,8	.856 –	.954 2,6,8	.935 6,8	.959 1:2,5:6,8	**.961** *all*	.957 1:2,6,8	.837 8	.960 1:3,5:6,8	.796 –
	5	.955 5:6,8	.957 1,3,5:6,8	.956 1,5:6	.960 1:3,5:6,8	.943 6,8	.906 8	**.961** *all*	.842 –	.951 2,5:6,8	.932 6,8	.958 1:2,5:8	**.958** *all*	.945 2,6,8	.863 8	.958 1:2,5:6,8	.786 –
	7	.954 3,5:6,8	.955 1,3,5:6,8	.952 5:6,8	**.959** 1:3,5:6,8	.941 6,8	.896 8	**.959** 1:3,5:6,8	.868 –	.955 2,5:6,8	.948 5:6,8	.956 1:2,5:6,8	**.957** 1:3,5:6,8	.932 6,8	.846 8	**.957** 1:3,5:6,8	.820 –

D – DDM; *E* – EDDM; *A* – ADWIN; *S* – SDDE; H_W – HDDM$_W$; H_A – HDDM$_A$; *a* – ALWAYS; *n* – NEVER

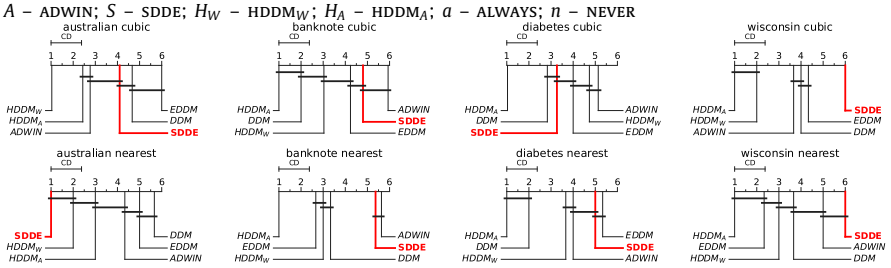
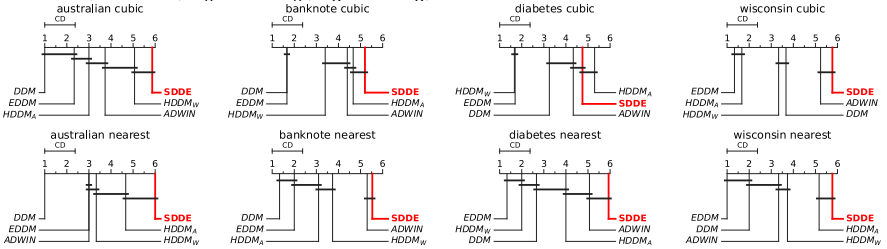


Table 3
Closest drift (D_1) error of drift detection (scaled by 10^{-2}).

		CUBIC							NEAREST						
		D	*E*	*A*	*S*	H_W	H_A	*a*	*D*	*E*	*A*	*S*	H_W	H_A	*a*
		1	2	3	4	5	6	7	1	2	3	4	5	6	7
AUSTRALIAN	3	.650 –	.554 1	.553 1:2	**.123** *all*	.288 1:3,6:7	.313 1:3,7	.333 1:3	.650 –	.393 1,3	.430 1	**.015** *all*	.297 1:3,7	.155 1:3,5,7	.333 1:3
	5	.390 –	.294 1	.208 1:2,6	**.126** *all*	.197 1:3,6:7	.227 1:2	.200 1:3,6	.390 –	.170 1,5,7	.075 1:2,5:7	**.000** *all*	.207 1	.087 1:2,5,7	.200 1,5
	7	.003 –	.002 1,6	.144 1:2,6	**.094** *all*	.130 1:3,6:7	.184 1	.142 1:3,6	.270 –	.147 1,3	.148 1	**.016** *all*	.133 1:3,7	.096 1:3,5,7	.142 1:3
BANKNOTE	3	.375 –	.314 1,7	.250 1:2,5,7	.178 1:3,5,7	.278 1:2,7	**.063** *all*	.333 1	.285 2,7	.310 7	**.026** 1:2,5:7	.074 1:2,7	.113 1:2,7	.105 1:2,5,7	.333 –
	5	.135 2,6:7	.247 –	**.111** 1:2,5:7	**.109** 1:2,5:7	.116 1:2,6:7	.162 2,7	.200 2	.199 7	.191 1,7	.021 1:2,5:7	**.000** *all*	.104 1:2,6:7	.160 1:2,7	.200 –
	7	.200 –	.140 1,7	.095 1:2,5,7	.078 1:3,5,7	.114 1:2,7	**.065** *all*	.143 1	.155 –	.116 1,6:7	.031 1:2,5:7	**.019** *all*	.087 1:2,6:7	.0136 1,7	.143 1
DIABETES	3	.355 2	.423 –	**.157** 1:2,5:7	**.208** 1:2,5:7	.351 1:2	.315 1:2,5,7	.333 1:2,5	.374 2	.390 –	.034 1:2,5:7	**.014** *all*	.340 1:2	.332 1:2,5	.333 1:2,5
	5	**.114** 2:3,5,7	.201 5	.169 2,5,7	.122 2:3,5,7	.204 –	**.112** 1:3,5,7	.200 2,5	.132 2,5,7	.204 –	.028 1:2,5:7	**.000** *all*	.184 2,7	.050 1:2,5,7	.200 2
	7	0.124 2,5,7	.129 5,7	.111 1:2,5,7	.090 1:3,5,7	.151 –	**.063** *all*	.143 5	.098 2,5,7	.130 5,7	**.038** 1:2,5:7	**.026** 1:2,5:7	.136 7	.085 1:2,5,7	.143 –
WISCONSIN	3	.477 2,6	.488 –	.386 1:2,5:6	**.201** *all*	.440 1:2,6	.485 2	.333 1:3,5:6	.335 2	.429 –	.173 1:2,7	.025 1:3,7	.025 1:3,7	**.015** 1:3,5,7	.333 1:2
	5	.200 2,5:6	.226 –	**.113** 1:2,5:7	**.116** 1:2,5:7	.205 2,6	.210 2	.200 1:2,5:6	.197 2,7	.222 –	.050 1:2,5,7	**.000** *all*	.055 1:2,7	.047 1:3,5,7	.200 2
	7	.137 2,5:7	.174 6	.116 1:2,5:7	**.080** *all*	.146 2,6	.188 –	.143 2,5:6	.122 2,7	.150 –	.076 1:2,7	**.011** 1:3,5,7	.064 1:3,7	**.010** 1:3,5,7	.143 2

D – DDM; *E* – EDDM; *A* – ADWIN; *S* – SDDE; H_W – HDDM$_W$; H_A – HDDM$_A$; *a* – ALWAYS

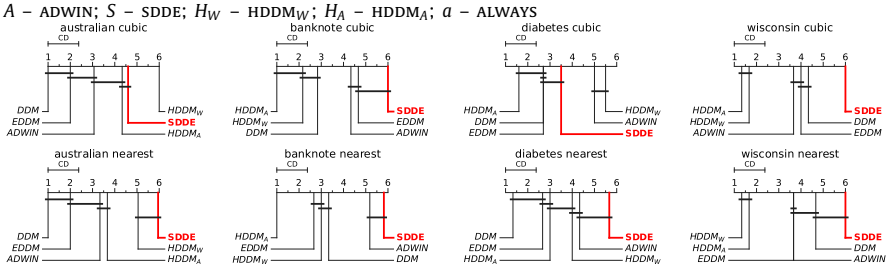


The values of the last analyzed criterion of the detection quality – *R* (*drift detection ratio*) – are presented in Table 5. The measure is dependent on the proportion between the number of drifts and the number of detections. The errors resulting from excessive alerts will be smaller than errors in the case of too few detections. It can be noticed, especially in the case of the

Table 4
Closest detection (D_2) error of drift detection (scaled by 10^{-3}).

		CUBIC							NEAREST						
		D	*E*	*A*	*S*	H_W	H_A	*a*	*D*	*E*	*A*	*S*	H_W	H_A	*a*
		1	2	3	4	5	6	7	1	2	3	4	5	6	7
AUSTRALIAN	3	.198 –	.181 1	.052 1:2	.015 1:3,6	.000 1:3,6	.031 1:3	**.000** 1:3,5:6	.198 –	.151 1	.119 1:2	.001 1:3,5:6	.003 1:3,6	.052 1:3	**.000** *all*
	5	.199 –	.182 1	.049 1:2	.027 1:3	.002 1:4,6	.034 1:3	**.000** *all*	.199 –	.154 1	.049 1:2,6	**.000** 1:3,5:6	.003 1:3,6	.050 1,2	**.000** 1:3,5:6
	7	.198 –	.147 1	.060 1:2	.020 1:3	.002 1:4,6	.023 1:3	**.000** *all*	.198 –	.154 1	.032 1:2	.001 1:3,5:6	.001 1:3,6	.023 1:3	**.000** *all*
BANKNOTE	3	.020 6	.004 1,3,5:6	.012 1,5:6	.001 1:3,5:6	.020 6	.050 –	**.000** *all*	.004 2,5:6	.013 6	.000 1,2,5:6	.000 1:2,5:6	.004 2,6	.092 –	**.000** *all*
	5	.009 5:6	.004 1,3,5:6	.006 1,5:6	.001 1:3,5:6	.010 6	.036 –	**.000** *all*	.003 2,5:6	.009 6	.000 1:2,5:6	**.000** 1:3,5:6	.004 2,6	.054 –	**.000** 1:3,5:6
	7	.008 5:6	.003 1,5:6	.003 1:2,5:6	.001 1:3,5:6	.009 6	.049 –	**.000** *all*	.007 6	.004 1,5:6	.000 1:2,5:6	.000 1:3,5:6	.007 1,6	.038 –	**.000** *all*
DIABETES	3	.017 6	.012 1,6	.008 1:2,4,6	.031 6	.002 1:4,6	.057 –	**.000** *all*	.033 –	.011 1	.001 1:2,5	.001 1:3,5	.003 1:2	**.000** 1:5	**.000** 1:5
	5	.032 6	.008 1,6	.005 1:2,4:6	.027 –	.006 1:2,4,6	.033 –	**.000** *all*	.019 6	.004 1,6	.000 1:2,5:6	**.000** 1:3,5:6	.003 1:2,6	.115 –	**.000** 1:3,5:6
	7	.020 2,6	.161 –	.007 1:2,4,6	.018 2,6	.005 1:4,6	.033 2	**.000** *all*	.116 –	.010 1,6	.010 1:2,6	.001 1:3,5:6	.003 1:3,6	.098 1	**.000** *all*
WISCONSIN	3	.031 2,5:6	.043 5:6	.025 1:2,5:6	.002 1:3,5:6	.118 –	.116 5	**.000** *all*	.011 2:3,5:6	.020 3,5:6	.047 5:6	.000 1:3,5:6	.132 –	.089 5	**.000** *all*
	5	.010 3,5:6	.004 1,3,5:6	.025 5:6	.001 1:3,5:6	.050 6	.052 –	**.000** *all*	.019 2,5:6	.043 5:6	.012 1:2,5:6	**.000** 1:3,5:6	.049 –	.048 5	**.000** 1:3,5:6
	7	.003 2:3,5:6	.005 3,5:6	.022 5:6	.000 1:3,5:6	.034 6	.035 –	**.000** *all*	.002 2:3,5:6	.009 3,5:6	.014 5:6	.000 1:3,5:6	.025 6	.033 –	**.000** *all*

D – DDM; *E* – EDDM; *A* – ADWIN; *S* – SDDE; H_W – HDDM$_W$; H_A – HDDM$_A$; *a* – ALWAYS



stream originating from the *australian* dataset, that the methods marking too little detections – such as DDM, ADWIN and HDDM$_A$ in the case of *nearest* drift – achieve greater errors than the ALWAYS method. For other data streams in the case of *cubic* drifts, ADWIN and HDDM$_A$ achieve the statistically best results. As can be seen in Fig. 13, the proposed SDDE method repeatedly signals a single change of concept during a *cubic* drift. Multiple detections separated in time emphasize the method's sensitivity to concept changes and may have beneficial effects. However, this measure of detector evaluation increases the method's error in the context of this measure of detector evaluation. In the case of *nearest* drifts, the issue does not occur, and the proposed SDDE detector has a statistical advantage over other methods.

In summary, the proposed SDDE detector achieves satisfactory classification accuracy and *drift detection error* values for the evaluated data streams. The detector is particularly effective in detecting sudden drifts and has high sensitivity in detection of incremental drifts.

6. Conclusions and future works

This publication proposes a *Statistical Drift Detection Ensemble* (SDDE) which is a novel method for concept drift detection in evolving data streams. The method was designed for general data stream processing and applications with no or delayed information about the classification quality. The proposed detector and other *state-of-the-art* detectors known from the literature were implemented in *Python* programming language and evaluated on synthetic data streams and as well generated streams based on real-world concepts.

Reference *pseudo-detectors*, either marking detectors in each data batch or not detecting any drifts, also was a part of the evaluation. To evaluate the operation of the detectors, three metrics for assessing the quality of drift detection in data streams with available drift occurrence ground-truth were proposed. The method parameters were reviewed, and the optimal hyperparameters were selected for further evaluation.

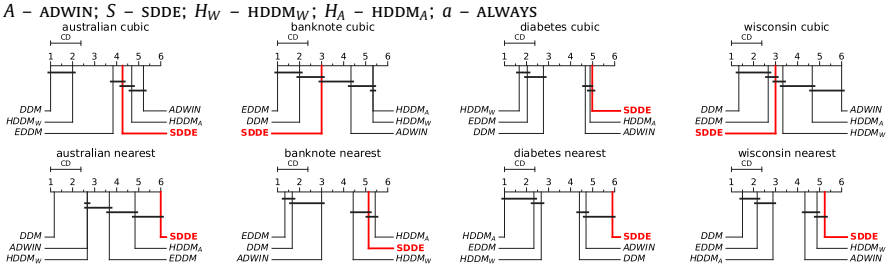
The comparison of the proposed method, *state-of-the-art* detectors, and reference extreme *pseudo-detectors* using both presented metrics and the classification quality, supported by the performed statistical tests, prove the advantage of the proposed method over other detectors. Research revealed the benefits of using metrics other than classification quality to improve drift detection quality measurement.

Future works will focus on the improvements of the SDDE method and further research under more challenging experimental conditions. The method may be equipped with an automatic mechanism of adjusting the *sensitivity* parameter, optimal for the specific data being analyzed. This parameter is critical since it greatly influences the number of detections, impacting the detection quality. Worth analyzing is also the potential of SDDE to detect concept drift occurrence in imbalanced data streams [51]. The problem of concept drift in the case of multi-label streams, which is relatively rarely tackled in the literature, is another potentially captivating research direction [52].

Table 5
Drift Detection ratio (R) error (scaled by 10^{-1}).

		CUBIC							NEAREST						
		D	E	A	S	H_W	H_A	a	D	E	A	S	H_W	H_A	a
		1	2	3	4	5	6	7	1	2	3	4	5	6	7
AUSTRALIAN	3	.200 –	.057 1,5,7	.025 1:2,4:5,7	.060 1,5,7	.096 1,7	**.000** *all*	.099 1	.200 –	.082 1,3,5,7	.200 –	**.003** *all*	.095 1,3,7	.050 1:3,5,7	.099 1,3
	5	.400 –	.029 1,5:7	**.025** *all*	.044 1,5:7	.092 1,7	.067 1,5,7	.099 1	.400 –	.069 1,3,5,7	.150 1	**.000** *all*	.092 1,3,7	.067 1,2:3,5,7	.099 1,3
	7	.600 –	.058 1,5,7	**.040** 1:2,5,7	**.034** 1:2,5,7	.090 1,7	**.040** 1:2,5,7	.098 1	.600 –	.053 1,5,7	.040 1:2,5,7	**.005** *all*	.090 1,7	.040 1:2,5,7	.098 1
BANKNOTE	3	.094 2,7	.098 7	.075 1,2,4,7	.091 1,2,7	.040 1:4,7	**.000** *all*	.099 –	.094 2,7	.097 7	.079 1:2,7	.059 1:3,7	**.050** 1:4,7	**.050** 1:4,7	.099 –
	5	.094 2,7	.097 7	.067 1:2,4,7	.087 1:2,7	.055 1:4,7	**.017** *all*	.099 –	.095 7	.090 1,7	.074 1:2,7	**.000** *all*	.055 1:3,7	.025 1:3,5,7	.099 –
	7	.094 2,7	.095 7	.071 1:2,4,6:7	.084 1:2,7	**.056** *all*	.075 1:2,4,7	.098 –	.092 2,7	.094 7	.072 1:2,7	.032 1:3,5,7	.059 1:3,7	**.000** *all*	.098 –
DIABETES	3	.080 2,5,7	.096 7	.057 1:2,5,7	.046 1:2,5,7	.094 2,7	**.025** 1:3,5,7	.099 –	.063 2,5:7	.094 5:7	.057 1:2,5:7	**.008** *all*	.095 6:7	.099 –	.099 –
	5	.075 2,5,7	.095 7	.064 1:2,5,7	.049 1:2,5,7	.090 2,7	**.000** *all*	.099 –	.081 2,5:7	.095 6:7	.044 1:2,5:7	**.000** *all*	.091 2,6:7	.400 –	.099 6
	7	.077 5,7	.068 1,5:7	.061 1:2,5:7	**.043** *all*	.087 7	.075 1,5,7	.098 –	**.017** 2:3,5:7	.092 6:7	.042 2,5:7	**.010** 2:3,5:7	.088 2,6:7	.250 –	.098 6
WISCONSIN	3	.097 5,7	.079 1,4:5,7	**.040** *all*	.093 1,5,7	.200 –	.050 1:2,4:5,7	.099 5	.095 7	.095 7	**.000** *all*	.043 1:2,5:7	.050 1:2,7	.050 1:2,7	.099 –
	5	.097 7	.097 1,7	**.017** *all*	.088 1:2,7	.025 1:2,4,7	.025 1:2,4,7	.099 –	.096 7	.085 1,7	.075 1:2,7	**.000** *all*	.025 1:3,6:7	.067 1:3,7	.099 –
	7	.096 7	.096 1,7	**.000** *all*	.086 1:2,7	.040 1:2,4,7	.040 1:2,4,7	.098 –	.096 6:7	.093 1,6:7	.053 1:2,6:7	.030 1:3,6:7	**.000** *all*	.133 –	.098 6

D – DDM; E – EDDM; A – ADWIN; S – SDDE; H_W – HDDM$_W$; H_A – HDDM$_A$; a – ALWAYS



CRediT authorship contribution statement

Joanna Komorniczak: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – original draft, Visualization. **Paweł Zyblewski:** Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – original draft. **Paweł Ksieniewicz:** Conceptualization, Methodology, Validation, Investigation, Resources, Writing – review & editing, Visualization, Supervision.

Declaration of competing interest

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

Acknowledgments

This work was supported by the Polish National Science Centre under the grant No. 2017/27/B/ST6/01325 as well by the statutory funds of the Department of Systems and Computer Networks, Faculty of Electronics, Wroclaw University of Science and Technology, Poland.

References

[1] A. Bifet, Classifier concept drift detection and the illusion of progress, in: L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L.A. Zadeh, J.M. Zurada (Eds.), Artificial Intelligence and Soft Computing, Springer International Publishing, Cham, 2017, pp. 715–725.

[2] J. Gama, I. Žliobaite, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM Comput. Surv. 46 (4) (2014).

[3] S. Hashemi, Y. Yang, Flexible decision tree for data stream classification in the presence of concept change, noise and missing values, Data Min. Knowl. Discov. 19 (1) (2009) 95–131.

[4] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Wożniak, F. Herrera, A survey on data preprocessing for data stream mining: Current status and future directions, Neurocomputing 239 (2017) 39–57.

[5] R. Barros, S. Santos, A large-scale comparison of concept drift detectors, Inform. Sci. 451–452 (2018).

[6] M. Bahri, A. Bifet, J. Gama, H.M. Gomes, S. Maniu, Data stream analysis: Foundations, major tasks and tools, WIREs Data Min. Knowl. Discov. 11 (2021).

[7] P. Zyblewski, P. Ksieniewicz, M. Woźniak, Classifier selection for highly imbalanced data streams with minority driven ensemble, in: L. Rutkowski, R. Scherer, M. Korytkowski, W. Pedrycz, R. Tadeusiewicz, J.M. Zurada (Eds.), Artificial Intelligence and Soft Computing, Springer International Publishing, Cham, 2019, pp. 626–635.

[8] J. Komorniczak, P. Zyblewski, P. Ksieniewicz, Prior probability estimation in dynamically imbalanced data streams, in: 2021 International Joint Conference on Neural Networks, IJCNN, IEEE, 2021, pp. 1–7.

[9] I. Zliobaite, Learning under concept drift: An overview, 2010, CoRR abs/1010.4784.

[10] G. Webb, R. Hyde, H. Cao, H.-L. Nguyen, F. Petitjean, Characterizing concept drift, Data Min. Knowl. Discov. 30 (2016).

[11] G. Widmer, M. Kubat, Effective Learning in Dynamic Environments by Explicit Context Tracking, Vol. 667, 1994.

[12] R. Barros, S. Santos, A large-scale comparison of concept drift detectors, Inform. Sci. 451–452 (2018).

[13] J. Gama, P. Medas, G. Castillo, P. Rodrigues, Learning with drift detection, in: A.L.C. Bazzan, S. Labidi (Eds.), Advances in Artificial Intelligence, SBIA 2004, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 286–295.

[14] M. Baena-García, J. Campo-Ávila, R. Fidalgo-Merino, A. Bifet, R. Gavald, R. Morales-Bueno, Early drift detection method, 2006.
[15] A. Bifet, R. Gavaldà, Learning from time-changing data with adaptive windowing, in: Proceedings of the 7th SIAM International Conference on Data Mining, Vol. 7, 2007.
[16] S.H. Bach, M.A. Maloof, Paired learners for concept drift, in: 2008 Eighth IEEE International Conference on Data Mining, 2008, pp. 23–32.
[17] I. Frías-Blanco, J.d. Campo-Ávila, G. Ramos-Jiménez, R. Morales-Bueno, A. Ortiz-Díaz, Y. Caballero-Mota, Online and non-parametric drift detection methods based on hoeffding's bounds, IEEE Trans. Knowl. Data Eng. 27 (3) (2015) 810–823, http://dx.doi.org/10.1109/TKDE.2014.2345382.
[18] S. Micevska, A. Awad, S. Sakr, SDDM: An interpretable statistical concept drift detection method for data streams, J. Intell. Inf. Syst. 56 (2021).
[19] C. Molnar, Interpretable Machine Learning, 2020.
[20] G. Webb, L. Lee, F. Petitjean, B. Goethals, Understanding concept drift, 2017.
[21] D.A. Levin, Y. Peres, Markov Chains and Mixing Times, Vol. 107, American Mathematical Soc., 2017.
[22] J. Kolter, M. Maloof, Dynamic weighted majority: An ensemble method for drifting concepts., J. Mach. Learn. Res. 8 (2007) 2755–2790.
[23] L.L. Minku, X. Yao, DDD: A new ensemble approach for dealing with concept drift, IEEE Trans. Knowl. Data Eng. 24 (4) (2012) 619–633.
[24] L. Du, Q. Song, L. Zhu, X. Zhu, A selective detector ensemble for concept drift detection, Comput. J. 58 (3) (2014) 457–471.
[25] B. Maciel, S. Santos, R. Barros, A lightweight concept drift detection ensemble, 2015.
[26] A. Bifet, Classifier concept drift detection and the illusion of progress, in: L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L.A. Zadeh, J.M. Zurada (Eds.), Artificial Intelligence and Soft Computing, Springer International Publishing, Cham, 2017, pp. 715–725.
[27] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, G. Zhang, Learning under concept drift: A review, IEEE Trans. Knowl. Data Eng. 31 (12) (2018) 2346–2363.
[28] B. Krawczyk, L.L. Minku, J. Gama, J. Stefanowski, M. Woźniak, Ensemble learning for data stream analysis: A survey, Inf. Fusion 37 (2017) 132–156.
[29] N. Street, Y. Kim, A streaming ensemble algorithm (SEA) for large-scale classification, in: Proceedings of the 7Th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001, pp. 377–382.
[30] H. Wang, W. Fan, P.S. Yu, J. Han, Mining concept-drifting data streams using ensemble classifiers, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD '03, ACM, New York, NY, USA, 2003, pp. 226–235.
[31] D. Brzeziński, J. Stefanowski, Accuracy updated ensemble for data streams with concept drift, in: E. Corchado, M. Kurzyński, M. Woźniak (Eds.), Hybrid Artificial Intelligent Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 155–163.
[32] M. Woźniak, A. Kasprzak, P. Cal, Weighted aging classifier ensemble for the incremental drifted data streams, in: Flexible Query Answering Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 579–588.
[33] R. Elwell, R. Polikar, Incremental learning of concept drift in nonstationary environments, IEEE Trans. Neural Netw. 22 (10) (2011) 1517–1531.
[34] R. Polikar, L. Upda, S.S. Upda, V. Honavar, Learn++: An incremental learning algorithm for supervised neural networks, IEEE Trans. Syst., Man, Cybern., Part C (Applications and Reviews) 31 (4) (2001) 497–508.
[35] H.M. Gomes, A. Bifet, J. Read, J.P. Barddal, F. Enembreck, B. Pfharinger, G. Holmes, T. Abdessalem, Adaptive random forests for evolving data stream classification, Mach. Learn. 106 (9) (2017) 1469–1495.
[36] N.C. Oza, S.J. Russell, Online bagging and boosting, in: International Workshop on Artificial Intelligence and Statistics, PMLR, 2001, pp. 229–236.
[37] A. Bifet, G. Holmes, B. Pfahringer, Leveraging bagging for evolving data streams, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2010, pp. 135–150.
[38] H.M. Gomes, J. Read, A. Bifet, Streaming random patches for evolving data stream classification, in: 2019 IEEE International Conference on Data Mining, ICDM, IEEE, 2019, pp. 240–249.
[39] T.K. Ho, The random subspace method for constructing decision forests, IEEE Trans. Pattern Anal. Mach. Intell. 20 (8) (1998) 832–844.
[40] D. Wang, P. Wu, P. Zhao, Y. Wu, C. Miao, S.C. Hoi, High-dimensional data stream classification via sparse online learning, in: 2014 IEEE International Conference on Data Mining, IEEE, 2014, pp. 1007–1012.
[41] A. Cano, B. Krawczyk, Kappa updated ensemble for drifting data stream mining, Mach. Learn. 109 (1) (2020) 175–218.
[42] D. Brzezinski, J. Stefanowski, R. Susmaga, I. Szczech, Visual-based analysis of classification measures and their properties for class imbalanced problems, Inform. Sci. 462 (2018) 242–261.
[43] D. Brzezinski, J. Stefanowski, R. Susmaga, I. Szczech, On the dynamics of classification measures for imbalanced and streaming data, IEEE Trans. Neural Netw. Learn. Syst. (2019) 1–11.
[44] H. Hu, M. Kantardzic, T.S. Sethi, No free lunch theorem for concept drift detection in streaming data classification: A review, Wiley Interdiscipl. Rev.: Data Min. Knowl. Discov. 10 (2) (2020) e1327.
[45] E. Hellinger, Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen., J. Für Die Reine Und Angew. Math. 1909 (136) (1909) 210–271.
[46] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[47] P. Ksieniewicz, P. Zyblewski, Stream-learn–open-source python library for difficult data stream batch analysis, 2020, arXiv preprint arXiv:2001.11077.
[48] J. Montiel, J. Read, A. Bifet, T. Abdessalem, Scikit-multiflow: A multi-output streaming framework, J. Mach. Learn. Res. 19 (72) (2018) 1–5.
[49] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J.F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T.E. Oliphant, Array programming with NumPy, Nature 585 (7825) (2020) 357–362.
[50] J. Komorniczak, P. Ksieniewicz, Data stream generation through real concept's interpolation, in: 30th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2022, (Bruges, Belgium), October 5-7, 2022, 2022.
[51] A. Cano, B. Krawczyk, ROSE: Robust online self-adjusting ensemble for continual learning on imbalanced drifting data streams, Mach. Learn. (2022) 1–39.
[52] G. Alberghini, S.B. Junior, A. Cano, Adaptive ensemble of self-adjusting nearest neighbor subspaces for multi-label drifting data streams, Neurocomputing (2022).

[C2]
Pawel Ksieniewicz. “Processing data stream with chunk-similarity model selection”. W: *Applied Intelligence* (lip. 2022). DOI: 10.1007/s10489-022-03826-4

Applied Intelligence
https://doi.org/10.1007/s10489-022-03826-4

Processing data stream with chunk-similarity model selection

Pawel Ksieniewicz[1]

Accepted: 29 May 2022

Abstract
The classification of data stream susceptible to the concept drift phenomenon has been a field of intensive research for many years. One of the dominant strategies of the proposed solutions is the application of classifier ensembles with the member classifiers validated on their actual prediction quality. This paper is a proposal of a new ensemble method – *Covariance-signature Concept Selector* – which, like *state-of-the-art* solutions, uses both the model accumulation paradigm and the detection of changes in the data posterior probability, but in the integrated procedure. However, instead of ensemble fusion, it performs a static classifier selection, where model similarity assessment to the currently processed data chunk serves as a concept selector. The proposed method was subjected to a series of computer experiments assessing its temporal complexity and efficiency in classifying streams with synthetic and real concepts. The conducted experimental analysis allows concluding the advantage of this proposal over *state-of-the-art* methods in the identified pool of problems and high potential in practical applications.

Keywords Data stream · Classifier selection · Classification · Pattern recognition

1 Introduction

An almost trivial introduction, but at the same time hard to deny, is the often overused statement that *the modern world is filled with data*. The difficult beginning of the third decade of the 21st century has irreversibly transferred the central axis of human existence to the global network of computer systems, which, almost like in W. Gibson's Neuromancer, spans the vast majority of our everyday lives. Most of the time, we organize and process subsequent portions of data in e-mails, instant messages, remote video calls, documents, reports, and notifications. Only to spend our free time accepting more portions of data in the form of *Netflix* or *Youtube* materials tailored to our taste, posts from our social bubble on *Twitter* or music served by *Spotify*. In such times *Machine Learning* drifts its intuitive meaning into a marketing slogan, eagerly taken up by companies such as *Google* or *Amazon*, which is to be a magic panacea for understanding the amount of data we produce and receive every day.

The reality behind this slogan is a much more prosaic complex of difficulties. Typical, classic recognition models target stationary problems, i.e., those describing a particular unchanging concept represented by a finite set of labeled problem instances [1]. The existence of highly numerous data sets, which do not allow for storing the entire dataset in the memory of a computer system, justifies the development of *inductive learning* paradigm with *incremental learning* [2]. Most often using iterative model update procedure based on upcoming data batches. In the case of high-dimensional data, prone to the *curse of dimensionality* [3], methods of *feature selection* and *extraction* are used [4], aimed at reducing the difficulty of the analyzed problem. In the case of data with a high cost of label acquisition, *active* and *semi-supervised learning* paradigms are used [5, 6], allowing the identification of the most challenging objects for the recognition model, thanks to which it is possible to reduce the involvement of human experts in the field. All these methods fit into the common domain of *Big Data* [7], offering solutions for problems characterized by a large amount of data (*volume*). They also have to deal with high speed of data processing (*velocity*), a great *variety* affecting the difficulty with the reliability of labeling (*veracity*) and the potential *value* for the end-user of the system.

✉ Pawel Ksieniewicz
pawel.ksieniewicz@pwr.edu.pl

[1] Department of Systems and Computer Networks, Wroclaw University of Science and Technology, Wybrzeże Stanisława Wyspiańskiego 27, Wroclaw, 50-370, Poland

An additional challenge for *pattern recognition* systems appears when the problems perfectly described by *5V* of *Big Data* also differ by the dynamics of the concepts contained in them. This subject has been dealt with for twenty years in the field of *data stream processing*, defining the characteristics of the variability of the problems *posterior probability* as the phenomenon of *concept drift* [8]. However, classification problems represented by data streams are rarely limited to describing a single point in time, and the systems dedicated to their processing must take into account the temporal changes of class definitions [9].

According to the established taxonomy, this variability may also have different characteristics. First of all, we can talk about *real* and *virtual drifts* [10]. In both of these cases, the distribution of objects in the feature space changes, but in *virtual drift* it does not affect the decision boundary of the model. A particular case of such a situation is the drift in prior probability [11], which apparently does not change the class definitions, but by changing their counts, often leads to noticeable changes in the classifier's decisions and its measurable quality [12]. This type of subject is more widely discussed in the recently popular subfield of imbalanced stream processing [13–15].

The characteristic of its dynamics is a much more frequently analyzed property of *concept drift* [16]. Literature distinguishes *sudden drifts* – consisting in an immediate change of one concept into another [17], *incremental drifts* – where the concept smoothly transforms its distribution [18], and *gradual drifts* – where we deal with a transition period in which two different definitions of the same class occur simultaneously in a changing proportion [19]. The most popular techniques used in dealing with the problems changing over time are built-in mechanisms that allow updating the recognition model taking into account the forgetting of old concepts [20]. There may also be distinguished pre-processing methods dedicated to streams [21] or – the most common in literature – ensemble approaches, using multiple recognition models in line with the wisdom of crowds paradigm [22, 23].

The oldest widely used ensemble method dedicated to processing data streams is the *Streaming Ensemble Algorithm* (SEA) proposed in 2001 by Street and Kim [24]. It is a relatively simple approach based on the accumulation of models built on successive data chunks until reaching the limit of predictors. Exceeding the limit forces the weakest model to be removed from the pool according to the quality calculated on the most recent portion of the labeled data. Further development of this idea led to the *Accuracy Weighted Algorithm* (AWE) [25], which proposes a metric for evaluating models and weighting their supports when obtaining the final ensemble prediction based on the mean square error of the classification. Later proposals most often further developed the paradigm of selecting the models of best quality, such as the *Accuracy Updated Ensemble* (AUE) [26] – correcting the metric from AWE and additionally introducing the possibility of updating the models in the pool, *Recursive Ensemble Approach* (REA) [27] – balancing data with the use of historical samples or *Weighted Aging Ensemble* (WAE) [28] – proposing advanced mechanisms to rejuvenate models and assess the diversity of the constructed pool.

In the current trends in the design of recognition models for the processing of data streams, the use of *Hoeffding Tree* (HTC) varieties based on the *Hoeffding boundary* [29] as the base model gained a significant advantage. It is a family of incremental classification methods that allow to reject the assumption of invariability of a posterior distribution while maintaining high processing efficiency with very large data sets [30]. This makes it a perfect fit for data streams prone to concept drift. Currently, the most frequently used variant of HTC is *Concept-adapting Very Fast Decision Tree learner* (CVFDT) [31], identical to *Very Fast Decision Tree* (VFDT) with moving window, but characterized by a much lower computational complexity. This is what the methods of the current *state-of-the-art* in the field are based on. These include *Leveraging Bagging Classifier* (LBC) [32] – an approach developing *Oza Bagging* [33], increasing resampling based on the Poisson distribution, using input and output detection codes and introducing the *Adaptive Windowing* (ADWIN) [34] drift detector as a pruning tool.

Another *state-of-the-art* approach, *Adaptive Random Forest Classifier* (ARF) [35] is actually fixed on HTC as a base classifier, using the ADWIN detector as well as the LBC, but in splits based on subspaces and building the *background trees*, preparing to recognize the new concept as the detector picks up a drift warning. An extremely interesting alternative to LBC and ARF – outperforming them in imbalanced data streams – is the *Kappa Updated Ensemble* (KUE) [36], also based on resampling using the Poisson distribution, building a diversified subspace band with a fixed size, but implementing pruning based on Kohen-Kappa metric – more sensitive to class imbalance.

Inclusion of methods like ADWIN in the body of *state-of-the-art* induction procedures highlights the importance of another topic of research into dynamic data streams – *concept drift detection* [37, 38]. The dominant majority of methods of this type use the base model that classifies incoming problem instances, analyzing changes in recognition quality. This paradigm has been common since the *Drift Detection Method* (DDM) [39], which analyzes online *error-rate*. The *Early Drift Detection Method* (EDDM) [40] introduced later proposes the *distance-error-rate*, measuring the averaged distances between successive classifier errors. Alternative methods often use tests such as the *Page-Hinkley Test* (PHT) [41] or CUSUM [42, 43], which compares the current quality of a classifier with its average so far [44], or the *Statistical Drift Detection Method* (SDDM) [45]

– which produces metrics based on prior and posterior probabilities. Also dependent on the base model are the popular algorithms such as ADWIN or *Paired Learners* (PL) [46] based on, successively, two windows of variable size while maintaining a single recognition model and implementing a similar mechanics with the strategy of using two models of different reactivity.

This paper proposes a novel algorithm being a product of approaches utilized both by *ensemble classification* algorithms for *data streams* and *drift detectors*. On the one hand, it replaces the *drift detection* with *concept identification*. On the other hand, it constructs an updatable pool of classifiers dedicated to the identified characteristics of the statistical relationship between the attributes of the problem changing in time. The premise of the proposed algorithm is the construction of a diverse pool of classification models, the purpose of which is not the mere assurance of diversity but rather an appropriate adjustment of the prediction to the diversity of concepts previously encountered by the recognition system. An additional guideline is an attempt to abandon the paradigm of model assessment by their quality common in *state-of-the-art* solutions and replace it with an assessment of the scale of similarity between the training and test data.

The main goal of the proposed approach is:

a) to reduce the use of labels in the induction procedure, which will use unsupervised method for the detection of changes in the concept,
b) to expand the potential of using data stream processing methods for other base classifiers than HTC,
c) to reduce the processing time of a single batch in the stream classification.

Although tree-based models are perfect for classifying synthetic problems based on legacy generators and a pool of real benchmark streams, according to Wolpert's theorem, it is impossible to recognize one classification model as universal for all recognition problems. Therefore, it is justified to propose alternative methods that better use the generalization potential of neural models for specific examples analyzed in this work. Such algorithms often allow for convergence in problems described only by quantitative attributes with a normal distribution in significantly less time than CVFDT.

The main contributions of the work are:

- A proposal of the *Covariance-similarity Concept Selector* (CSCS) algorithm dedicated to the classification of *data streams* containing *concept drift*.
- A proposal of procedure for generating data streams utilizing static data, enabling the proper assessment of the quality of the stream classifier in the environment of real concepts.
- A series of computer experiments assessing (*a*) the computational overhead and (*b*) the quality of classification of the proposed method with the comparison of *state-of-the-art* methods on the example of synthetic and real concepts.

2 Methods

The research hypothesis experimentally validated by this paper is that: *"It is possible to distinguish between separate data chunks belonging to the pool of concepts identified based on the signatures determined on their statistical properties"*. The key challenge here is, therefore, to find an appropriate method to establish the unambiguous and easy to cross-compare *signature of the concept*.

The considered method for determining the signature, due to the simple obtaining process for large matrices [47], will be the calculation of the auto-covariance matrix, denoted as K_{XX}. Statistics defines this structure as a square matrix describing the covariance between each pair of elements of a given random vector [48]. Suppose its calculation for a given portion of the training set. Such a case gives a symmetrical matrix with dimensions of $d \times d$. The diagonal of such representation contains information about the variance of each of the d attributes of the problem. Symmetrical rows and columns store generalized information about its notion in multiple dimensions.

It is essential to underline that such an approach – like any information retrieval method – employs the naivety of a particular assumption. In this case, the assumption is the quantitative nature of the attributes. Such problem description enables drifting variability both in the bias and the relationships between the features of the problem.

2.1 Concept signature

This subsection aims to present the processing procedure of the proposed method on a simplified example of a drifting data stream that allows for appropriate visualization. Figure 1 shows auto-covariance matrices computed on twelve consecutive chunks of six data streams containing concept drifts typically distinguishable by the taxonomy (*sudden*, *gradual* and *incremental drift* with both *recurring* and *nonrecurring* concepts). Each data stream contains two concept drifts whose central points lie between the third and fourth and the ninth and tenth chunks. For the clarity of the visualization, to allow the observation of differences between the calculated *concept signatures* with a bare eye, all problems have been simplified to streams with six dimensions, half of which is informative and the other half – repeated [49]. The matrices were visualized in grayscale, with eight-bit depth, to the range limited by the *standard deviation* of the values present in the historical signatures.

As can be observed, the structure of the relationship between the problem attributes, measured by their variance

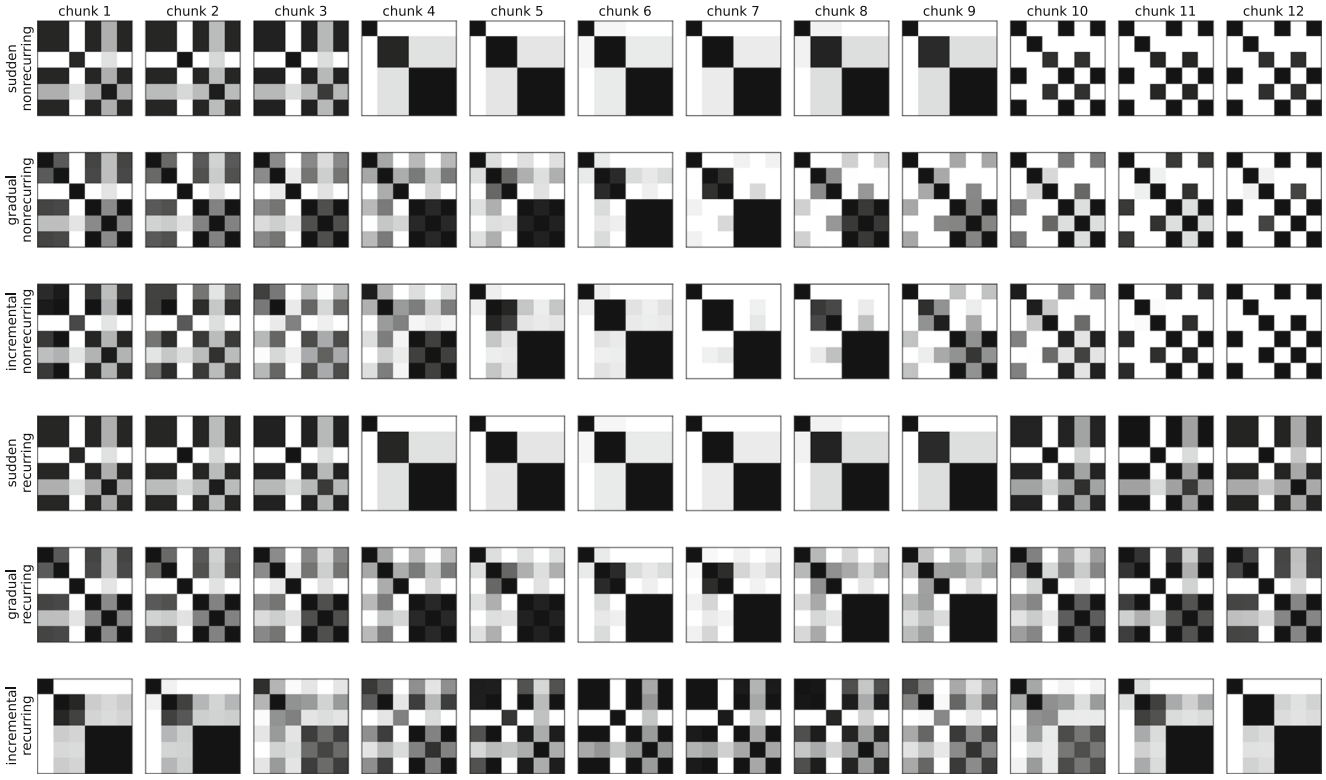


Fig. 1 Visualization of auto-covariance matrices computed on twelve consecutive chunks of six data streams containing concept drifts typically distinguishable by the taxonomy

and cross-variance, changes according to the concept drift dynamics. In the simplest case of *sudden drift*, concept signatures are legible and, in simplified visualization, allow for unequivocal identification of the current concept. On the other hand, in *incremental* and *gradual drifts*, the transition phase (*incremental drift*) or the co-presence phase (*gradual drift*) of two different concepts is noticed in the course of concept changes.

A preliminary experiment verifies whether the changes in the signatures reflect the real nature of stream dynamic. Its results are presented in Figs. 2 and 3. As in the case of auto-covariance matrix visualization, this experiment also uses simplified, six-dimensional data streams.

The streams were expanded to two hundred chunks of five hundred objects to analyze long-run behavior. Each of the six presented plots consists of two columns. The right column (*Covmats*) shows the current signature in the selected chunk. In contrast, the left column (*variance of difference*) shows the value vector of the variance of the difference between the current and each of the historical concept signatures. Dashed lines present actual values, while the red line shows its trend smoothed by the median filter. Additionally, in the form of a dotted line in the background, a *concept change curve* is presented, determined according to the *concept sigmoid spacing* parameter of the synthetic stream generator available in the *stream-learn* package [50]. The plots represent ten points in time, twenty chunks apart, starting with the eighth iteration.

Figure 2 shows the changes in the *variance of differences* vector for data streams with *non-recurring drift*, ie. introducing a new concept for each drift occurrence. As can be seen by observing the first row of visualization (*chunk 7*), the initial phase of the stream in which only one concept is present leads to changes in a much smaller range than the others. At the same time, the vector distribution seems (apart from a small number of observations) to be normal.

The observation of the second row of the visualization (*chunk 27*) allows for the verification of the curve value at the end of the first concept drift. The variance trend at the first drift in each of the dynamics almost perfectly mirrors the dynamics of the *concept change curve*.

The third line of the visualization (*chunk 47*) shows how the stable, equal trend curve of the signature difference variance reflects the last moment of stability of the second concept. Further rows confirm these observations, which is essential, also showing that the differences between the current signature and those for historical concepts differ but are still far from the signatures of the current concept. An interesting observation here is also the fact that in the case of *incremental drifts*, the differences in the transition phases are noticeably lower than in the case of *gradual drift*, which suggests a significant similarity in their transitional concepts.

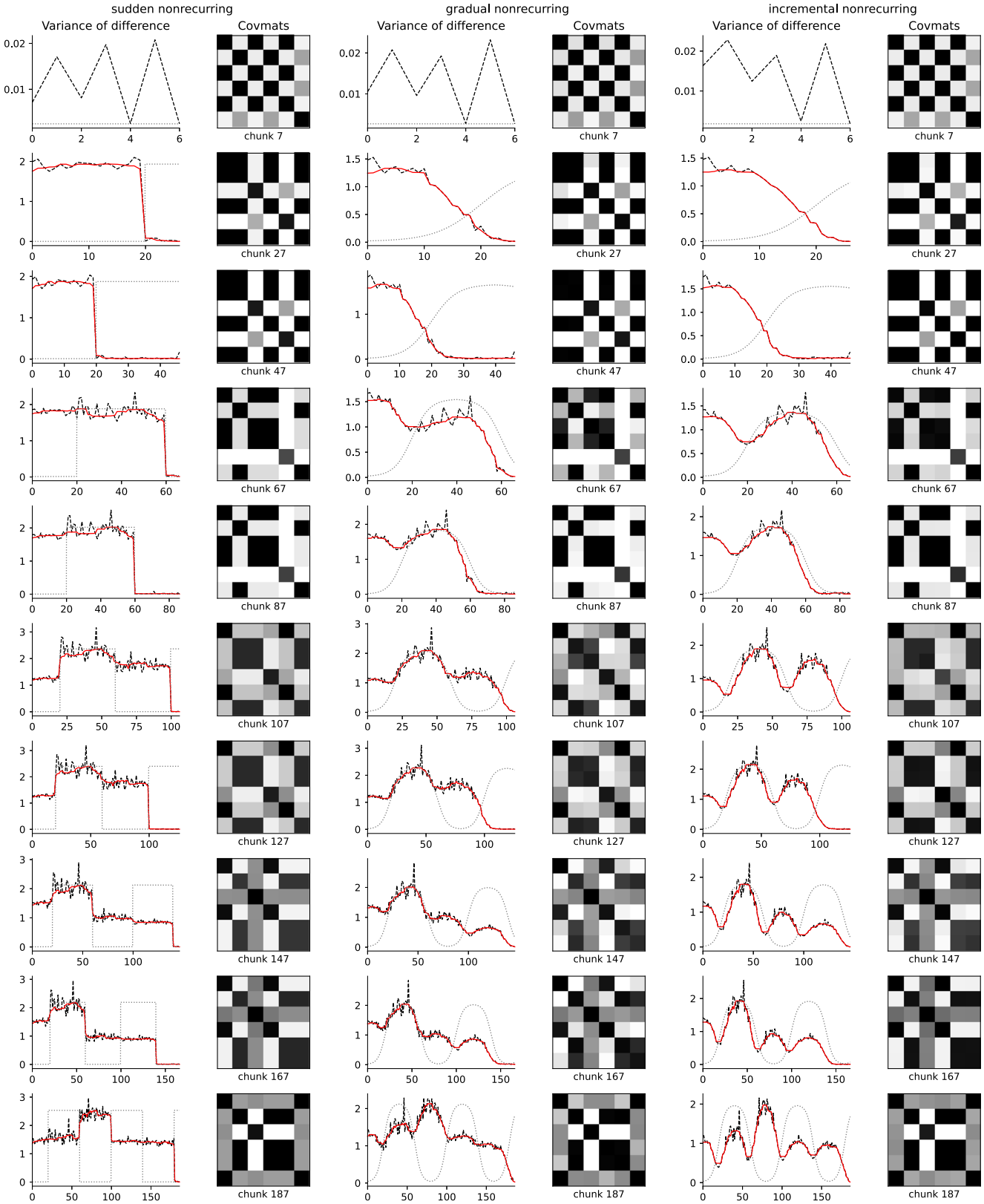


Fig. 2 The changes in the variance of differences vector for data streams with non-recurring drift

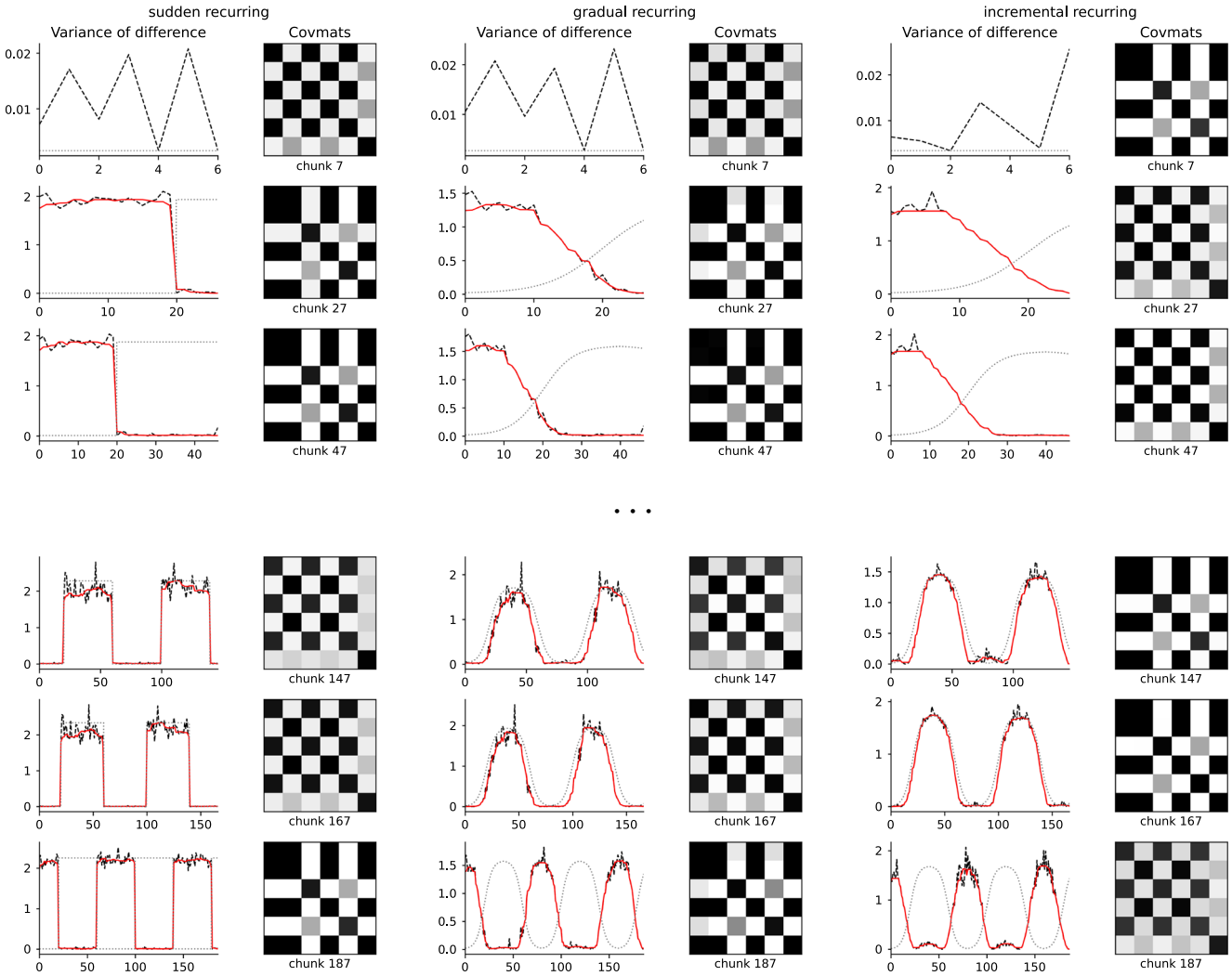


Fig. 3 The changes in the variance of differences vector for data streams with non-recurring drift (shortened to begining and ending of processing)

However, it creates a potential risk, which may lead to the generalization of the transition phase as a uniform concept in the future – in the worst case – leading to the generation and utilization of a random classifier during the drift.

Figure 3 shows the changes in the variance of the differences vector for data streams with *recurrent drift*, i.e., oscillating between the two concepts each time drift occurs. Its analysis makes it possible to replicate all the observations made so far, but without noticing a decrease in the variance in the transition phases between the concepts in the *incremental drift*. Such observation probably comes from a characteristic of *recurrent drifts*, where all transition phases are coherent and, in the context of class distribution, occur with the same or mirrored dynamics of the posterior distribution. *Recurring drift* also allows for an even more precise analysis of differences between the dynamics of transition and the values of the variance of the difference vector.

As we may observe, the red line on the plots almost perfectly reflects the concept change curve, which allows for the initial plausibility of the research hypothesis. Moreover, it leads to a proposal of a method using *concept signatures* – calculated as data chunks auto-covariance matrixes – for effective classification of data streams.

2.2 Covariance-signature concept selector

This subsection presents the context of using the *concept signature* described above in the construction of an effective method for data stream processing. The proposed recognition procedure of *Covariance-signature Concept Selector* (CSCS) uses the achievements of *state-of-the-art* ensemble methods such as ARF, LBC or KUE and builds a pool of classifiers, but, unlike the existing algorithms, does not integrate them, but conducts the selection of the most appropriate model for the currently predicted data chunk [51]. It means that, unlike in the rest of the considered methods, in the CSCS algorithm, a decision is always made by a single model, in line with the approach of static classifier selection.

What is extremely important, the calculation of the concept's signature without pattern labels potentially enables (a) the identification of the concept in the prediction procedure and (b) the selection of an appropriate classifier built on it. As mentioned in the Introduction, the paradigm of the qualitative assessment of the models present in the pool, which aimed at maintaining the highest possible quality of the prediction, is rejected here. Its replacement is the paradigm of the highest similarity between the predictor and the problem. Such change allows both the identification of new concepts and the selection of a model suitable for the current data chunk. The pseudo-code of the CSCS algorithm is presented in Algorithm 1.

Algorithm 1 Covariance-signature Concept Selector – CSCS.

Require:
$\mathcal{DS}$: data stream,
k: number of historical models to store,
n: number of historical covariance matrixes to store,
$\Psi()$: base classifier, $H()$: harmonic mean.
Ensure:
C: set of historical covariance matrixes,
Π: pool of historical models,
M: set of covariance matrixes for historical models.

$C, \Pi, M \Leftarrow \emptyset, \emptyset, \emptyset$
for $DS_i : \{X_i, y_i\}$ **in** $\mathcal{DS}$ **do**
  $c \Leftarrow K_{X_i X_i}$ ▷ COVARIANCE FOR CURRENT CHUNK
  $cv \Leftarrow [\mathbb{V}(c - c') \; \forall \; c' \in C]$ ▷ VARIANCES OF COVARIANCE DIFFERENCES
  $mv \Leftarrow [\mathbb{V}(c - c') \; \forall \; c' \in M]$
  $t \Leftarrow H(cv)$ ▷ NOVELTY TRESHOLD
  if none $mv_j < t$ **or** $\|mv\| = 0$ **then** ▷ NOVEL CONCEPT
    $\Pi \Leftarrow \Pi \cup \Psi(DS_i)$
    $M \Leftarrow M \cup c$
  else ▷ IDENTIFIED CONCEPT
    for any $\Psi_j \in \Pi$ **where** $mv_j < margin$ **do**
      update Ψ_j with X_i, y_i
    end for
  end if
  if $\|M\| > k$ **then** ▷ MODEL PRUNING
    remove oldest element from M
    remove oldest element from Π
  end if
  if $\|C\| > n$ **then** ▷ COVMAT PRUNING
    remove oldest element from C
  end if
end for

Two pruning hyperparameters typical for this type of procedure control the processing: k – specifying how many models might construct the pool and n – specifying how many historical signatures should take part in comparisons [22]. For the correct operation of the method, it is required for the selected base classifier Ψ, like in all *state-of-the-art* algorithms since proposition of AUE, to be updating capable. In line with the method assumptions, it does not have to be able to adapt independently to changes in the concept. Both *Hoeffding Trees* and the *Multi-layer Perceptron* [52] as well as simple methods such as *Gaussian Naive Bayes* [53] can be used here. Alternatively, for non-updateable base classifiers, SEA algorithm may be employed here as a meta-classifier giving this ability to every model. Each model in the pool is dedicated to understanding and generalizing a single, given snapshot of the posterior probability present in the stream's course.

The procedure starts with initializing the empty sets Π, C, and M. The set Π is to store a pool of up to k active classifiers. Set C holds a maximum of n most recent signatures and set M holds a signature for each Ψ currently contained in Π.

For each successive chunk $DS_i : \{X_i, y_i\}$, the concept signature c is first computed as the auto-covariance matrix of the set X_i. Then the vectors cv and mv are determined as the variances of the differences ($\mathbb{V}$) between c and each element of the sets C and M in the way described by Figs. 2 and 3.

In the next step, the *novelty threshold* t is determined, which is the *harmonic mean* of the vector cv. The DS_i concept is considered as yet unknown if none of mv elements are below the threshold t. For unknown concepts, the classifier trained on DS_i is added to the pool Π, and the set M is supplemented with the signature c. Otherwise, each Ψ_j model from Π for which the corresponding ms_j value was lower than the threshold t is considered coherent with the current concept and updated with DS_i. This approach allows the recognition system to adapt to new concepts and update stored models if the induced similarity between their concepts and the current batch of objects is detected.

After the proper part of the processing loop is completed, the procedure carries out the tidying up work. If the module M exceeds k, the oldest elements from sets M and Π are removed. Accordingly, if the module C exceeds n, the oldest element of the set C is removed. Such an approach ensures that the CSCS processing time is independent of the stream length and reduces the number of comparisons necessary to identify similarities between the concepts.

If less than k iterations detect less than n new concepts, the set M will still contain the signatures already removed from C, and the Π set – models paired with them. It is a deliberate and planned action to allow the CSCS algorithm to return to a temporarily obsolete model in the event of

recurrent drift. The purpose of the hyperparameter n is to determine how many concept signatures are present in the set C to correctly estimate the threshold t determining the identification of a new signature representing a concept that does not have a corresponding model in the Π pool. Such design aims to make the number of comparisons made at each method iteration independent of the current course of the stream, ensuring a relatively low time complexity of the proposed solution.

The CSCS prediction procedure for batch processing environment, described by Algorithm 2, is relatively simple and analogous to the training procedure. First, for the given test set X, signature c is determined. Next, the procedure computes the vector of variance of the signature differences of the stored models mv and returns the prediction of the selected model with the smallest mv value.

Algorithm 2 Prediction using CSCS for batch environment.

Require:
X: testing set,
Π: pool of models,
M: set of covariance matrixes for models.
Ensure:
y: set of predictions,

1: $c \Leftarrow K_{XX}$
2: $mv \Leftarrow [\mathbb{V}(c - c') \; \forall \; c' \in M]$
3: $y \Leftarrow \Pi_{argmin(mv)}(X)$

Algorithm 3 Prediction using CSCS for online environment.

Require:
$x \in X$: testing sample in a set of last m testing samples,
Π: pool of models,
M: set of covariance matrixes for models.
Ensure:
y: prediction,

1: $c \Leftarrow K_{XX}$
2: $mv \Leftarrow [\mathbb{V}(c - c') \; \forall \; c' \in M]$
3: $\Psi = \Pi_{argmin(mv)}$
4: $y \Leftarrow \Psi(x)$

In the presented prediction procedure, the decision is not made separately for each object but globally for the entire test set. In the case of using the CSCS method in online processing, it would be necessary to supplement the prediction procedure with a window mechanism, building a concept signature for a given horizon of recent test patterns. So, the online processing (Algorithm 3) procedure uses the last window of given m samples to determine c signature and gives prediction with the same approach to model identification.

3 Experimental evaluation

The principle of the CSCS algorithm may seem very similar to a typical *drift detection* procedure. While this method can detect changes in the concept, it cannot be identified as such a tool. The first difference is the use of an approach that does not rely on the assessment of the predictive capability of the system and does not harvest labels in any phase other than training a new model or updating existing ones. Second, the actual drift detections only occur here in the case of a data chunk describing a new, unknown concept and should not happen during any recurrent drift scenario.

Identification does not necessarily have to refer to an entirely new concept. In the case of incremental or gradual drifts, it may also occur several times in the course of changes, depending on their dynamics. Therefore, the planned experimental evaluation of the method will only concern its ability to construct a reliable recognition system, and it is in this context that the comparison with *state-of-the-art* methods will be prepared. For its needs, (a) 30 data streams containing synthetic concepts with quantitative features with different dynamics of change, (b) 15 data streams containing synthetic concepts with mixed quantitative-qualitative features, and (c) 8 data streams consisting of suddenly drifting real concepts with different dimensionalities were prepared.

3.1 Experimental scenarios and environment

3.1.1 Synthetic concepts from *stream-learn* generators

The relatively most superficial part of the experiment preparations was the generation of 30 synthetic data streams, carried out using the stream generator available in the *stream-learn* package. This generator always produces only quantitative attributes with a given dimension and proportion between informative, redundant, and random attributes. The shared part of the configuration of each of the streams prepared in this way was:

- 250 chunks for 250 objects each,
- 20 informative concept attributes,
- 10 concept drifts in the flow of a stream.

The six separate processing scenarios were flows diversified by type of drift:

1. non-recurring sudden drift,
2. recurring sudden drift
3. non-recurring gradual drift,
4. recurring gradual drift,
5. non-recurring incremental drift,
6. recurring incremental drift.

Each of the scenarios types of data streams generated a pool of problems in five replications differing in a randomly determined *random state* used so that – for optimal comparison – the same concepts were present in the following types of drift.

3.1.2 Synthetic concepts from *MOA* generators

In order to diversify the pool of synthetic problems, the experimental evaluation also used stream generators offered by the *MOA* package. These are solutions commonly used in most comparative experiments in data stream processing. The generators selected for the analysis were:

- *Random Radial Basis Function stream generator* (RBF) – a method for constructing problems with quantitative attributes, configurable in terms of the number of classes, attributes, and source centroids of the classes. Standard hyperparameterization was used here, i.e., binary problems with ten attributes and 50 class centroids.
- *Random Tree stream generator* (RTG) – a method proposed by Hulten and Domingos [54] based on a tree that randomly splits attributes and assigns random labels to leaves. It generates five quantitative and 25 qualitative attributes for a binary problem by default.
- *Agrawal stream generator* (AGR) – the most classical method of stream synthesis, proposed in 1993 by Agrawal et al. [55]. It always generates nine attributes, three of which are qualitative and six are discrete quantitative parameters from a uniform distribution.

Each of the stream generators prepared a pool of problems in five replications differing in a randomly determined *random states*.

The proposed method, due to decisions made to the structure of the concept signature calculation approach, should find its potential mainly in the RBF generator, where the characteristics of the attributes should allow for the correct identification of the concept by its signature. However, the predominance of qualitative or discrete quantitative attributes with a low range will probably impede proper identification, which seems to be a guideline against using this solution in streams similar to AGR or RTG.

3.1.3 Real concepts

The more complex challenge was to prepare a pool of problems with real concepts. In reference works there often appear a relatively short list of real streams, such as *electricity*, *poker-lsn* or *insects* to enumerate a few. Most often, however, they are very specific types of streams, characterized by (*a*) vanishing classes, (*b*) very low dimensionality, (*c*) categorical attributes only, or (*d*) strong class imbalance. Neither of these properties is the primary challenge of the research presented in this paper. Therefore, studies using them for the comparison of methods could be unreliable. For the purposes of the evaluation, a method of obtaining streams containing real concepts determined on the basis of static datasets was proposed.

The `@w4k2/benchmark_datasets` *Github* repository was used here, which contains a specially prepared, unified collection of classification datasets available for use in pattern recognition experiments built based on public UCI and KEEL repositories. The procedure for developing a stream containing real concepts consists of the following five steps:

1. From the available pool of datasets, select all binary problems, the dimensionality of which (number of attributes) is not less than d.
2. For each such dataset, draw 100 of d-dimensional subspaces, and then, using the full data set, the build GNB model on them.
3. If, in any of these cases, the model, when tested on the full training set, achieves at least 75% of *balanced accuracy score*, consider it as fit for use in a stream. Otherwise, ignore the dataset.
4. On all the sets considered as fit for use, perform oversampling using the SMOTE algorithm to obtain six thousand unique objects evenly distributed over both classes of the problem. Add such a set to the end of the stream.
5. If the data stream obtained in such a manner allows obtaining at least 100 chunks of 250 objects each, consider it as fit for the experiment.

The entire procedure of preparing streams with real concepts is carried out by a script publicly available in the `@w4k2/benchmark_datasets`[1] Github repository. It allowed the construction of eight data streams containing sudden drift, with real concepts described by 2 to 9 dimensions.

3.1.4 Experiments set-up

Computer experiments for the needs of the experimental evaluation have been fully implemented in the *stream-learn* environment, which is a set of tools compatible with the *scikit-learn* package. All the experimental, analytical scripts and implementation of the CSCS method are available in the public Github `@w4k2/cscs` repository.[2]

[1] https://github.com/w4k2/benchmark_datasets/blob/master/real_stream.py

[2] https://github.com/w4k2/cscs

In all experiments, two updatable classifiers were used under the configuration of hyperparameters:

- MLP – *Multi-layer Perceptron*:
  - 10 epochs per chunk in Experiments 1 and 2, which gives limit of 625,000 calls for criterion function,
  - 100 epochs per chunk in Experiment 3, which gives limit of 6,250,000 calls for criterion function,
  - 100 artificial neurons in single hidden layer,
  - *ReLU* activation function,
  - *adam* solver,
  - $L2$ penalty of 0.0001,
  - constant learning rate of 0.001,
- HTC – *Hoeffding Tree* or *Concept-adapting Very Fast Decision Tree*:
  - grace period of 250,
  - *information gain* split criterion,
  - split confidence of 0.0000001,
  - tie threshold of 0.05,
  - non-binary splits,
  - pre-pruning enabled,
  - *Naive Bayes Adaptive* leaf prediction.

where all the settings apart from the number of epochs per chunk for MLP, state the standard hyperparametrization. The ratio between the quality achieved by each of the classifiers at default 200 iterations of the optimizer was proportional to 10 iterations for stream-learn generated problems and 100 iterations for MOA generated and real problems, which preliminary research confirmed.

One of the main goals of the experimental evaluation was to verify the usefulness of various base classifiers in various ensemble classification methods in order to diversify the pool of identified *state-of-the-art* methods. Hence, apart from the currently most popular CVFDT, the potential of neural networks in implementing a simple MLP structure was analyzed, which allows for significantly faster inference than *Hoeffding trees*.

Eight standard comparative methods, indicated earlier in the Introduction and selected according to their popularity in the literature, were also used in the evaluation:

1. **REA** – with the limit of 10 classifiers in pool and pruning enabled,
2. **WAE** – with the limit of 10 classifiers in pool,
3. **SEA** – with the limit of 10 classifiers in pool and accuracy score metric.
4. **AWE** – with the limit of 10 classifiers validated with 5 splits and its original quality metric,
5. **AUE** – with the limit of 10 classifiers validated with 5 splits and its original quality metric.
6. **KUE** – with the pool size of 10 classifiers.
7. **ARF** – with the limit of 10 classifiers in pool (only for HTC comparisons).
8. **LBC** – with the limit of 10 classifiers in pool (only for HTC comparisons).

It should be underlined that the ARF and LBC algorithms are included only in comparisons using HTC as the base classifier. Such a decision comes from the limitation of ARF, which can create only decision tree models, while LBC hinders the effective construction of a neural network by relatively frequent resets of the base model.

For CSCS, to ensure a fair comparison, the k hyperparameter limiting pool size has also been set to 10, and hyperparameter n – limiting historical signature storage – was set to 100.

3.2 Experiment 1: computational overhead of streaming methods

The first experiment aimed to assess the temporal complexity of the analyzed recognition methods depending on the size of the problem described by a number of its features and the number of data stream chunks processed. It was supposed to allow for proper verification of the demand of commonly used recognition methods for the computing power of the systems that utilize them. All results on processing time were acquired with machine equipped with *Intel(R) Core(TM) i5 8265U* CPU with *3.7* GHz and *8* GB of *2666* MHz DDR4 RAM.

The previously declared streams pool was extended by an additional six for this experiment. It was not the quality of the constructed classification models that was important, but the processing time of individual methods. These data streams have been extended to 500 chunks representing dimensionality problems from 2 to 100 attributes (in 20 quants). There were ten drifts of the six types defined in previous subsections in each of them. As an additional element of analysis, a truncated version of the CSCS algorithm (CSCS-h) is present in processing for Experiment 1, which ignores n hyperparameter – responsible for the limit of historical signatures stored in the method memory.

Figure 4 shows, in the form of grayscale heatmaps, the average time in milliseconds of single chunk processing over these ranges by each of the compared methods with a neural network as a base classifier. As can be seen, the baseline MLP processing time here uses stable three milliseconds for any class of the problem. Processing time is not significantly dependent on the number of processed chunks or the dimensionality of a problem.

From the reference methods, the SEA has the relatively smallest computational overhead, which allows processing from about 30 milliseconds for two-dimensional problems

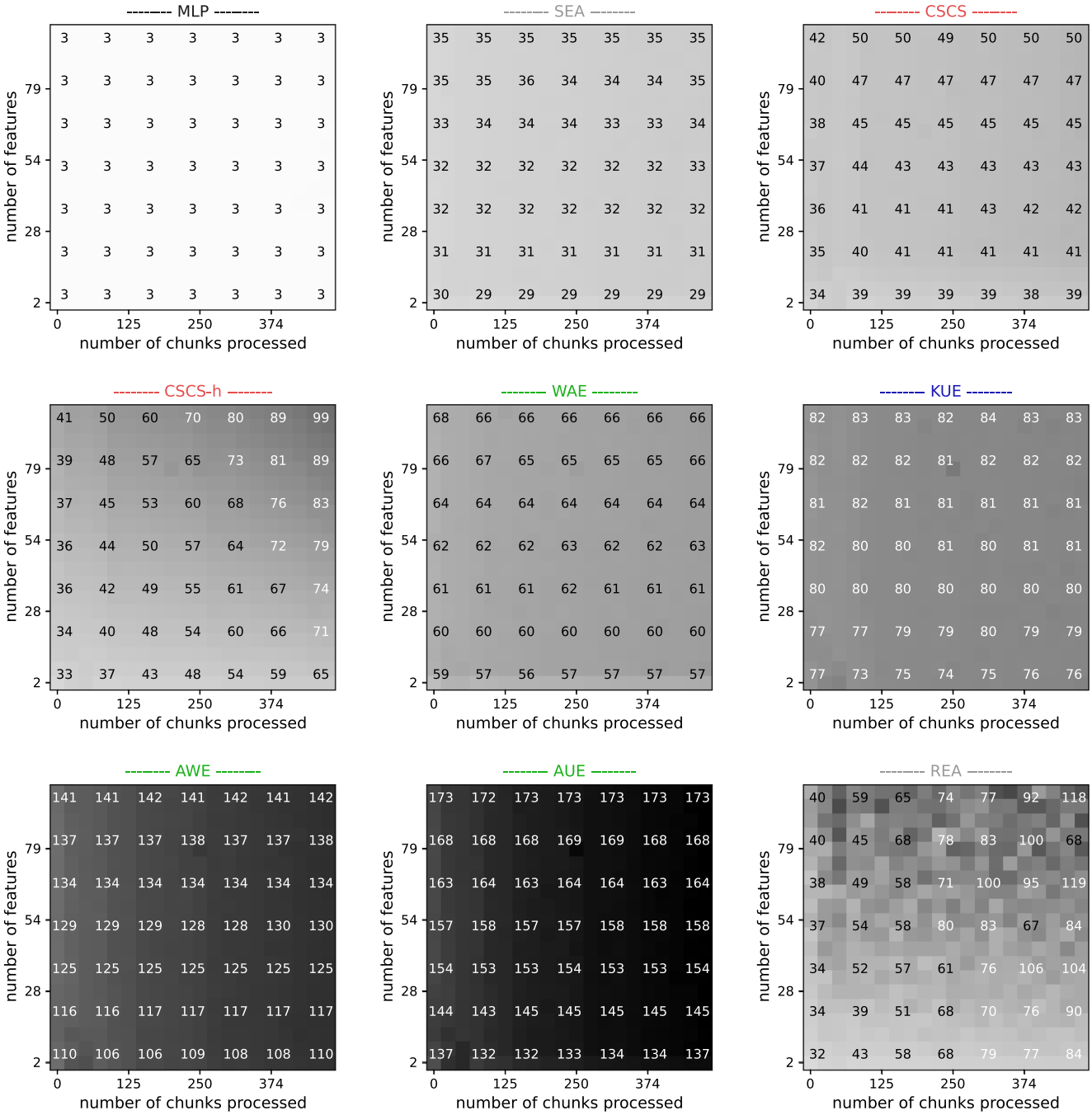


Fig. 4 Time (in milliseconds) to process a single chunk of data by different methods depending on the current run length and the number of attributes on *Multi-Layer Perceptron*

to about 35 milliseconds for 100-dimensional problems. Usage of WAE algorithm increases this time to range 56–68 and KUE to range 73–84. The AWE algorithm increases the processing time even further to about 105 milliseconds for planar problems and about 140 milliseconds for most complex cases. Likewise, the AUE algorithm extends this time to a range from about 130 to about 170 milliseconds. However, all these measurements turn out to be independent of the number of chunks processed so far apart from the very initial stage of processing. A completely separate category is built here by the REA algorithm, which shows strong dependencies on both the dimensionality of the problem and the number of processed chunks, reaching times ranging from 30 to 120 milliseconds.

In this comparison, the CSCS algorithm performs between the level of SEA – the simplest and WAE –

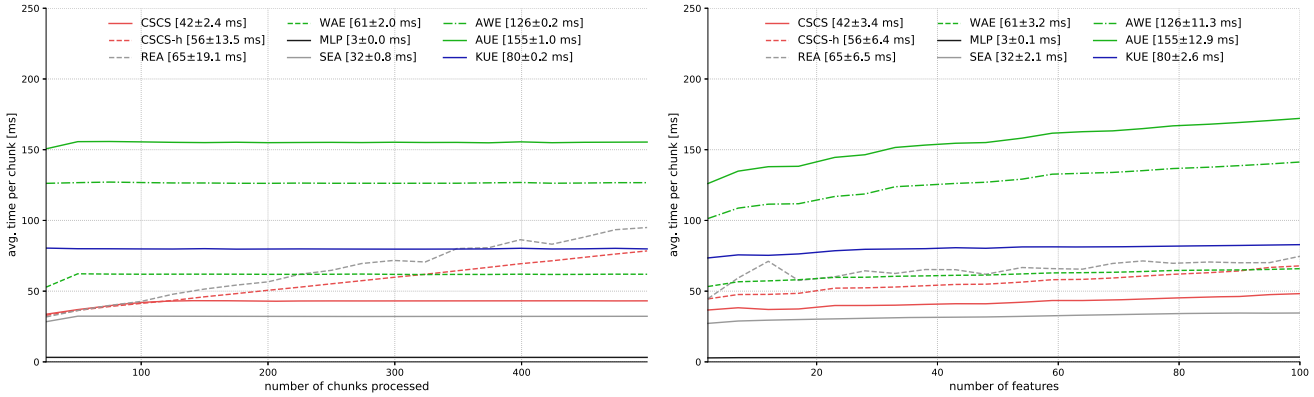


Fig. 5 Time course for processing a single chunk by different methods depending on the current number of chunks processed and the number of attributes on *Multi-Layer Perceptron*

the most optimized ensemble method for classification of data streams used in comparison. With processing times ranging from around 35 to 50 milliseconds, it provides a solution nearly or more than three times faster than the more complex AWE and AUE approaches. However, as can be seen in Fig. 5, in contrast to the competition, this method is not exactly time independent of the number of chunks processed so far. Instead of the complex procedure of assessing the models included in the pool, a less time-consuming statistical analysis of historical signatures is performed in it. As we can see, both on the CSCS-h curve and its heat map, conducting this analysis without limiting the signature horizon would lead to a strong correlation between the stream length and the processing time. At about 1000 chunks, it would equalize the average time needed for AWE and AUE while losing all the time advantage of the new stream analysis strategy. However, the introduction of this limit, after preliminary tests set for 100 signatures, leads to a solution that is slightly slower than SEA, much faster than AWE and AUE, and additionally, after exceeding 100 chunks, time independent of the stream length.

An additional observation that we can make in Fig. 5 is the trend expressed in the relationship between the dimensionality of the problem and the average processing time. As we can see, the standard deviation recorded in the study for the AWE method is 11.3 ms, and for the AUE method – 12.9 ms. Meanwhile, the CSCS method here maintains a standard deviation of 3.4 ms, giving a lower dynamics of processing time increase, which suggests that it may be able to preserve its effectiveness with potentially higher dimensionality.

The same visualizations prepared for the HTC algorithm as the base classifier gives completely different observations. The Hoeffding trees themselves show a strong dependence on the problem's dimensionality, increasing (with low dependence on the number of chunks processed) from about 15 milliseconds for planar problems to about 150 for problems of one hundred dimensions. Compared to the MLP with ten iterations, this gives five times the time for two dimensions and fifty times the time for one hundred dimensions. By making the comparison independent of the stream length (Fig. 7), it can be seen that a significant increase in MLP processing time, by order of magnitude from 10 to 100 iterations, leads to a more computationally demanding solution than HTC only for low-dimensional problems. With the problems typical of stream processing, represented by the stream scenarios, an MLP with 100 epochs is comparable in time to HTC, and an MLP with ten epochs much faster than it.

The computational resource demand hierarchy is also different here than in the case of MLP. The CSCS algorithm, as it may be observed in Fig. 6, generates relatively the lowest overhead here, processing in the range of about 20 to 160 milliseconds per chunk. However, during the time experiment, an upper processing limit was imposed – to filter out methods to slow to converge in the streaming environment – which interrupted learning when the given method exceeded a full second of processing per chunk. This limit was reached by all other ensemble algorithms, including even the relatively most unadorned SEA.

The ARF and LBC algorithms, which are *state-of-the-art* in the processing of balanced streams, reached this threshold in the earliest stages of processing, as can be seen in Fig. 8, achieving it with simple, eight-dimensional problems. The WAE and AUE algorithms remained functional for a bit longer, reaching the threshold around 30 dimensions. With 40 dimensions, the KUE algorithm has become ineffective, with 60 dimensions – the REA and AWE algorithms, and only with 90 - the SEA algorithm. Only the CSCS method and HTC's single base classifier met the limit under all conditions, never exceeding the 20 percent limit (200 milliseconds).

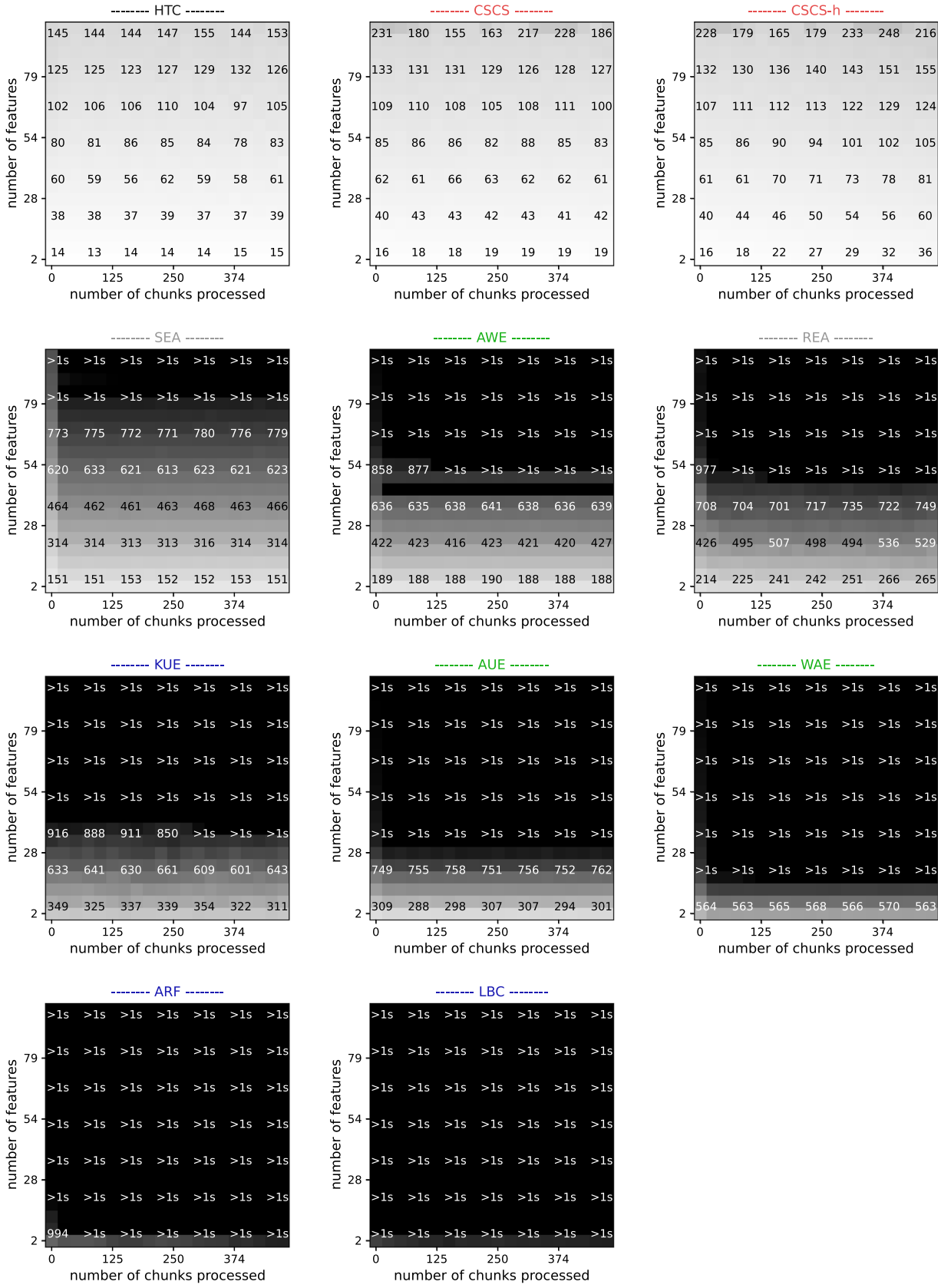


Fig. 6 Time (in milliseconds) to process a single chunk of data by different methods depending on the current run length and the number of attributes on *Hoeffding Tree*

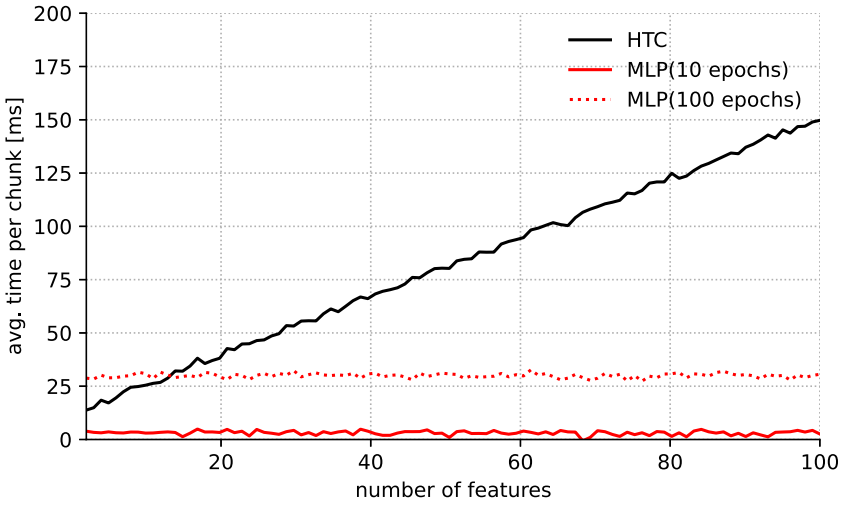


Fig. 7 Comparison of base classifiers training time as a function of dimensionality

3.3 Experiment 2: evaluation on synthetic concepts

The second experiment compared the proposed CSCS method with *state-of-art* algorithms for processing data streams with synthetic concepts. Figure 9 shows quality, as measured by *balanced accuracy score*, on exemplary – representative runs from three non-recurring processing scenarios from *stream-learn* generator for both MLP and HTC base classifier. The X-axis spreads the course of processing in successive chunks, and the Y-axis – quality, presented on a scale from the initial level of a random classifier (50%) to the perfect classification (100%), using the standard *Test-Then-Train* experimental protocol. Chunks in which the CSCS algorithm has detected new concepts are marked on the additional upper X-axis and with the use of dashed black lines. Each of the subfigures contains up to ten pairs of solid lines and hairlines. The solid lines represent the quality trend (the accumulative mean) and the hairlines – the same quality in the chunk for each compared method. The numerical values presented in the legend give the average quality of each method over the entire run of the data stream (Figs. 7 and 8).

For all non-recurring drift cases, with the usage of MLP, the CSCS method has a significant advantage over each of the reference algorithms ranging from 6% for incremental and sudden drifts to 7% for gradual drift. For the cases of gradual and incremental drift, the proportion between the qualities achieved by individual methods is identical. The CSCS algorithm achieves a clearly the best result, the SEA, AWE and AUE algorithms – results differing from each other at the level of thousandths, KUE achieves slightly lower scores, while the base MLP method and REA algorithm clearly stands out qualitatively downwards. In the case of sudden drift, there is a certainly noticeable difference in the SEA, AWE, AUE group, in which AUE achieves a higher quality than the competition. However, this advantage does not exceed one percent, and the result itself is still significantly worse than the one determined for the CSCS.

As can be seen on the hairlines showing the precise quality in each chunk, both smaller drops in quality at the

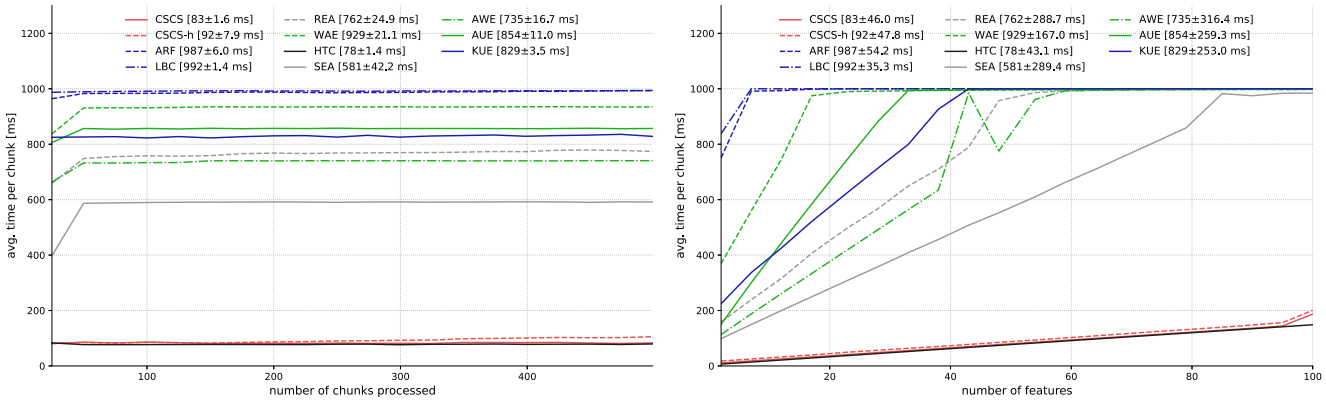


Fig. 8 Time course for processing a single chunk by different methods depending on the current number of chunks processed and the number of attributes on *Hoeffding Tree*

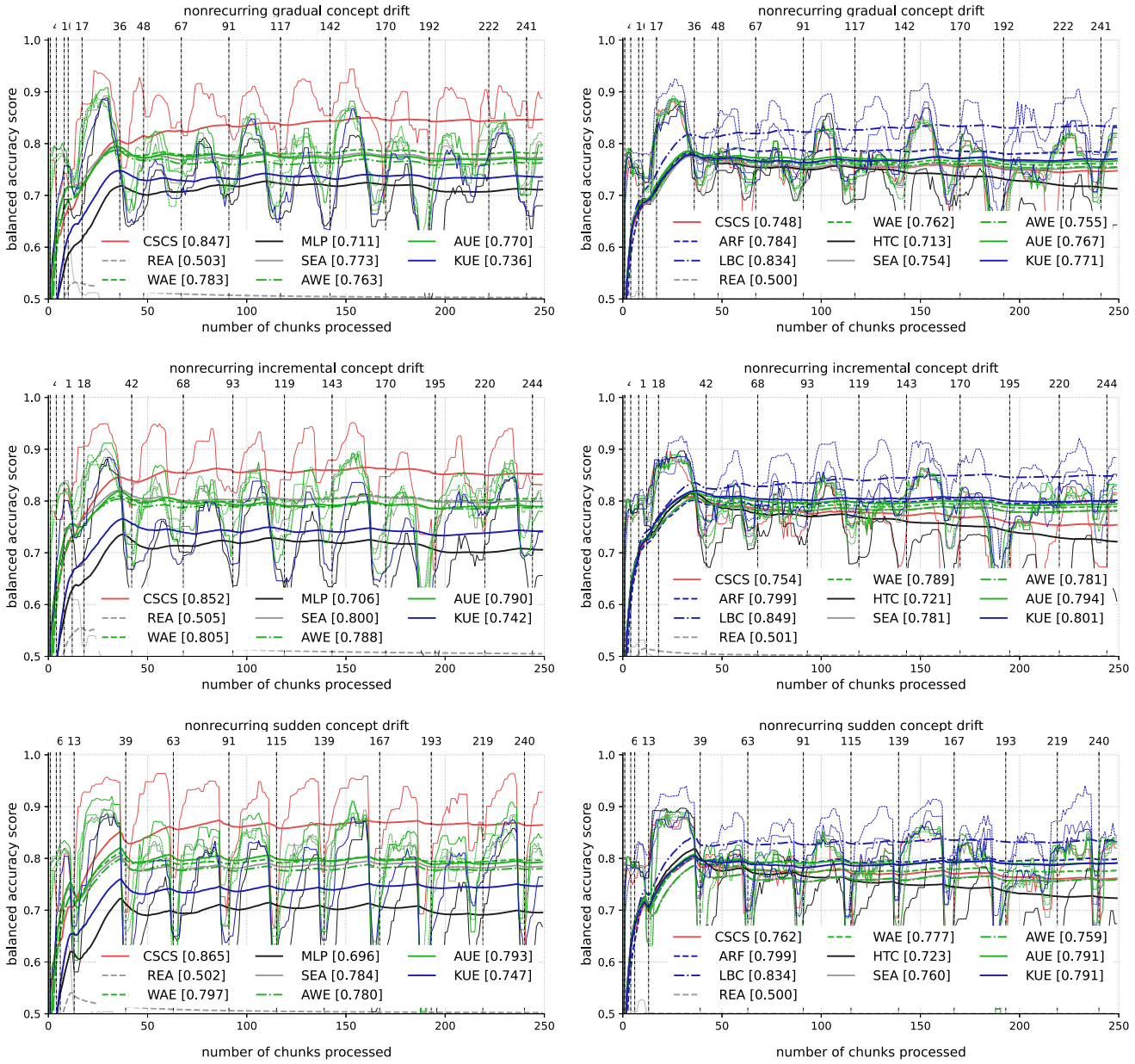


Fig. 9 *Balanced accuracy score* achieved by compared methods on exemplary - representative runs from three nonrecurring drift dynamics for *Multi-Layer Perceptron* (left) and *Hoeffding Tree* (right)

moment of drift and much faster convergence in the phases of the homogeneous concept characterize the CSCS method. What is particularly interesting, in the case of nonrecurrent gradual drift, there are also situations of smooth concept recognition, thanks to which in the initial phase of the drift – which should lead to a decrease in quality – it increases.

Using HTC as a base classifier gives noticeably lower results for these scenarios than MLP. Apart from the base model and mostly random REA, the most noticeable is a clear decline in the quality of CSCS, which seems to be an approach that performs poorly with HTC. A clear advantage over the competition is gained by the LBC algorithm, which, however, still always achieves slightly worse results than the CSCS-MLP pair. It is particularly evident in the exact quality of the grading at each point where MLP converges much faster and leads to higher scores. Two groups can be observed further. Slightly better ARF and KUE and slightly worse AWE, AUE and WAE. The results for REA, which oscillate around the random classifier in each problem, will not be mentioned in further analysis.

For recurrent drift scenarios (Fig. 10) with the MLP as base classifier, more diverse results are achieved. In the case of gradual drift, SEA, AWE and WAE methods create

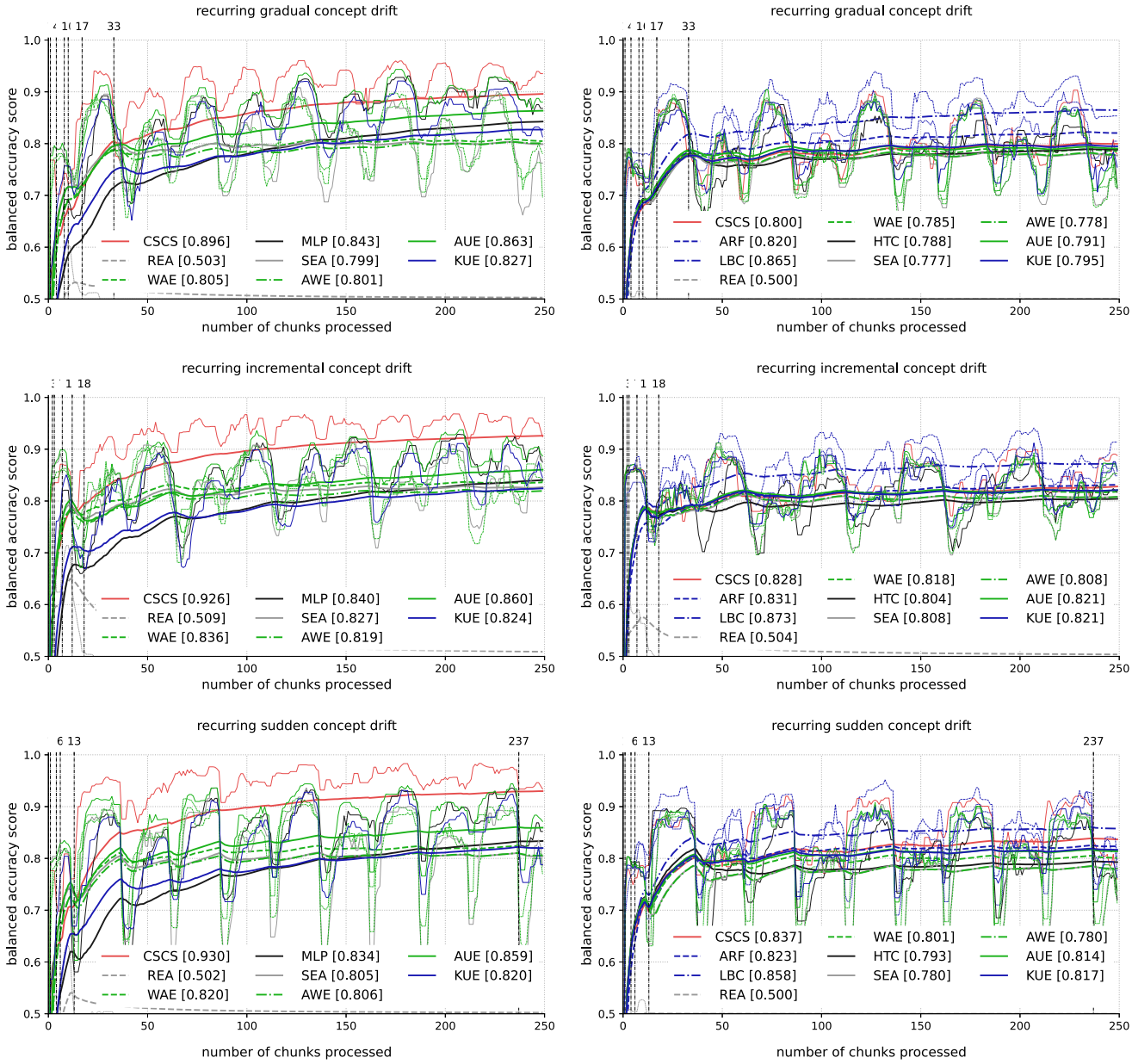


Fig. 10 *Balanced accuracy score* achieved by compared methods on exemplary - representative runs from three recurring drift dynamics for *Multi-Layer Perceptron* (left) and *Hoeffding Tree* (right)

the worst group of solutions, finally reaching about 80% of the balanced accuracy. The base MLP method is initially the worst. However, around the hundredth chunk, it exceeds the predictive capacity of the mentioned group and KUE, stabilizing at a quality level just below AUE and just above KUE. The AUE method, on the finite stream flow, shows its greater utility than the other reference methods and achieves the final quality with an average level of 86%. The CSCS algorithm, except for the initial phase of the first drift, here duplicates the efficiency achieved for nonrecurrent drifts, with (*a*) smaller quality drops throughout the stream course and (*b*) faster recovery after the drift. The results for incremental and sudden drift are similar. However, for sudden drift, after a good knowledge of both concepts contained in the stream after about 100 chunks, the CSCS algorithm gradually eliminates the phenomenon of quality loss after the drift occurs.

In the case of using HTC as the base classifier, the results are similar to those for non-recurring concept drifts. Again, we have a significant drop in CSCS quality to 80 percent, and again the LBC algorithm turns out to be the best in the comparison. Regardless of this, again the CSCS-MLP pair gives results noticeably higher than the *state-of-the-art* LBC-HTC.

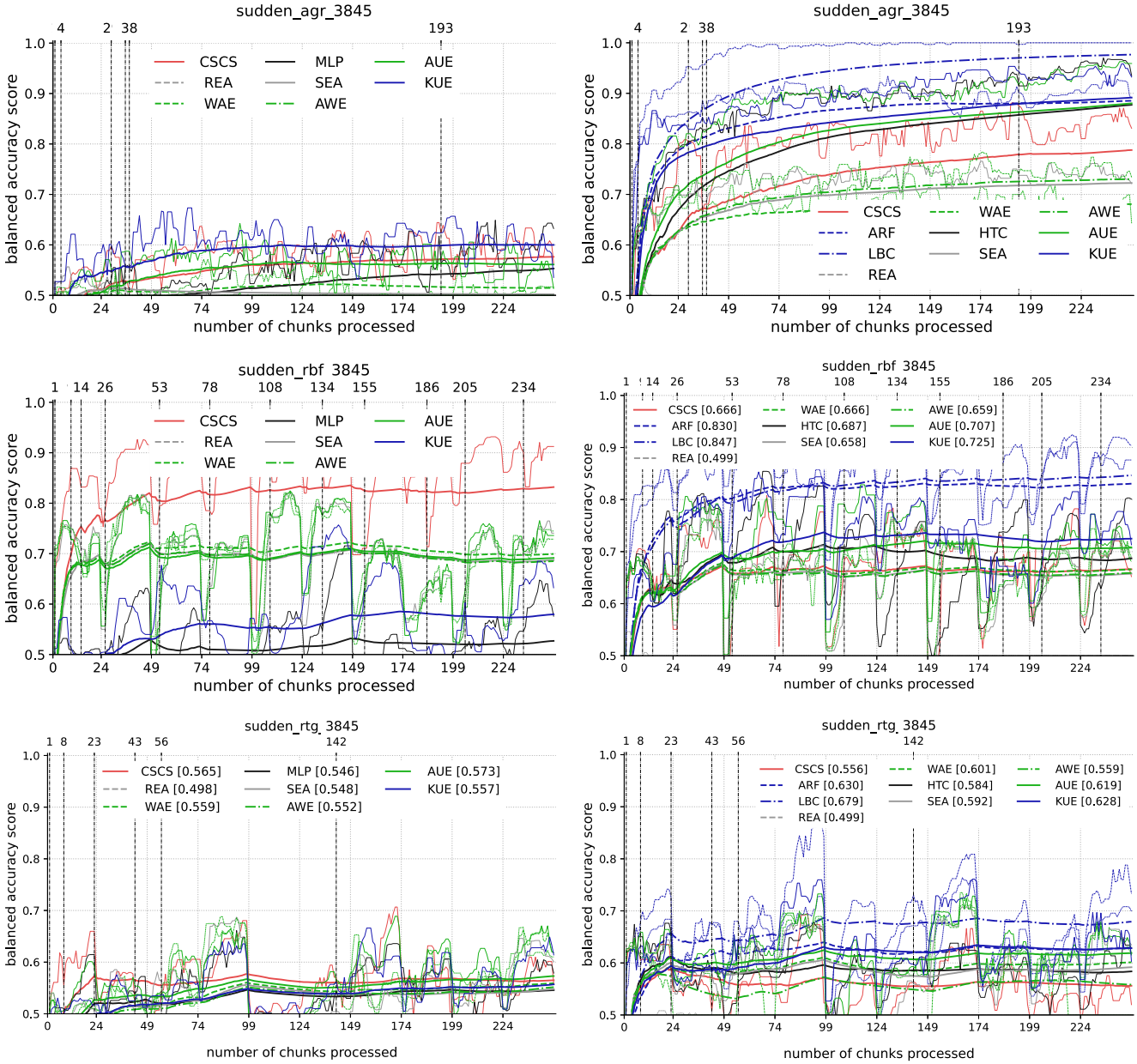


Fig. 11 *Balanced accuracy score* achieved by compared methods on exemplary - representative run from three selected scenarios generated by *MOA* for *Multi-Layer Perceptron* (left) and *Hoeffding Tree* (right)

The results achieved with the standard configuration of methods for flows generated with the use of the MOA packet are noticeably different (Fig. 11). As can be seen, the MLP base classifier achieves in each of these cases a much lower efficiency than HTC, which may explain the special popularity of the latter approach in most of the current research. Due to the low efficiency of neural networks, the LBC-HTC pair turns out to be by far the best solution in the entire rate, which in the case of the AGR generator gains even 10% advantage over the next, ARF-HTC, which achieves comparable generalization power for the RBF generator, but for RTG lays behind the LBC again.

The assumptions about the low efficiency of CSCS for the quality-based RTG and AGR generators were therefore confirmed. However, the behavior of the CSCS along with the MLP algorithm in the case of the RBF generator is interesting. Despite the final efficiency at a level much lower than that of the LBC, there is a significant 10% advantage of CSCS-MLP over other solutions using MLP. It is worth to recall here the observation from Fig. 7, that with about ten problem attributes (default RBF configuration), the HTC processing power demand is equal to the MLP chunk demand of 100 epochs. Meanwhile, the presented results relate to MLP with exactly 10 epochs on chunk.

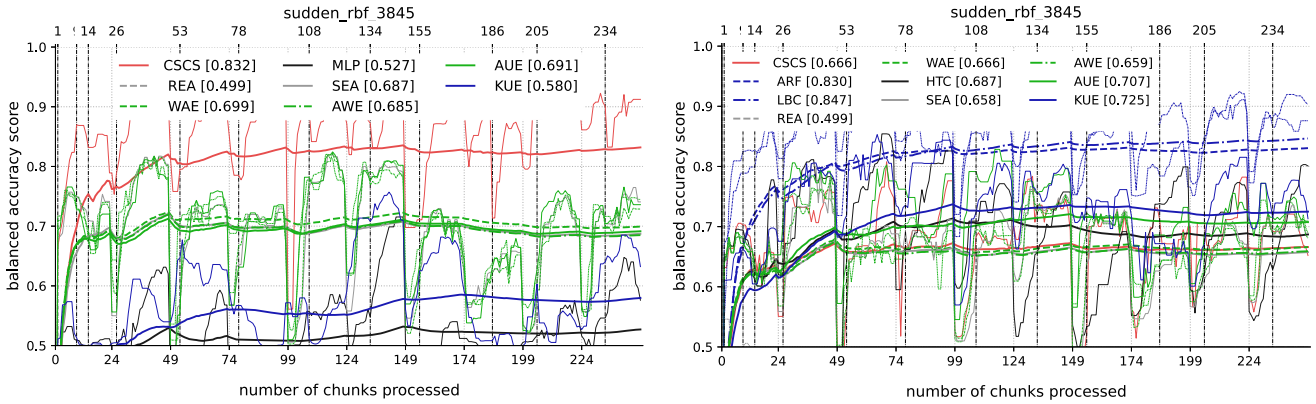


Fig. 12 *Balanced accuracy score* achieved by compared methods on exemplary - representative run from RBF scenario for *Multi-Layer Perceptron* with enhanced training time (left) and *Hoeffding Tree* (right)

Therefore, it would be potentially interesting to compare the effectiveness of HTC against the MLP of one hundred epochs.

This is shown in Fig. 12. As can be seen, after increasing the number of MLP epochs matching the computational demand with HTC, the results are quite different. The CSCS-MLP pair achieves results comparable to the ARF and only slightly weaker than the LBC. The AWE, AUE and WAE methods also benefited from a more complex base model, catching up with their HTC counterparts in terms of maximum efficiency, but showing higher learning dynamics. Only KUE model preserved its weak abilities. In connection with these observations, the MLP configuration with one hundred iterations on chunk was used for evaluation on data streams with real concepts (*Experiment 3*).

In addition to benchmarking the recognition efficiency, it is also important to verify the ability of the CSCS to identify new concepts. Figure 13 shows the concept identification capability in the prediction procedure of the CSCS algorithm for data streams with synthetic concepts generated with *stream-learn* package. The X-axis represents the stream course, and the Y-axis – the absolute identifier of the model included in the Π pool. The scatter plot superimposed on the plot space marks – with black dots – indicates which concept was used to predict which chunk. As in the case of Fig. 9, an additional upper X-axis marks the identification of chunks building the new model. In the background of the illustration, as in Fig. 2, the *concept change curve* of the data stream is highlighted.

As can be seen, the most problematic for the CSCS algorithm is the initial flow of the data stream, in which the identification of a new concept (based on the harmonic mean) is impossible due to the lack of comparative data from concepts other than the first one. In the neighborhood of chunk 25, where the second concept stabilizes, CSCS achieves the proper problem recognition capacity. Identifying new concepts here most often occurs near the end of a gradual or incremental drift or close to sudden drift. In the case of gradual and sudden drift, a momentary disturbance in identification is noticeable, which, after the end of the stable concept phase, temporarily indicates one of the historical models, but this quickly leads to the identification of a new concept and a stable shift of the entire recognition system to it. It is interesting to confirm the assumption in the description of the method that the transient phases of non-recurring incremental drift are very similar. Each transition phase was identified as a common concept #4. Nevertheless, the training phases of the identified model in the algorithm give only seemingly wrong selection, which in the course of drifts builds a transitional model capable of maintaining a higher predictive ability than *state-of-the-art* methods. It is confirmed by the observations made on Figs. 9 and 10.

Figure 14 shows the same analysis but for flows generated by MOA. As can be seen here, the assumptions about the low ability to identify concepts are confirmed in streams with the dominance of categorical attributes because for AGR and RTG, we can observe only a few detections. In the case of AGR streams, only the initial hyper-excitations and a single irrelevant detection downstream are observed. The graph for RTG streams presents slightly better recognition abilities – where there is a higher percentage of quantitative attributes – however, still not all significant changes are detected. Correct detection is achieved by CSCS in the case of RBF streams, which justifies the high CSCS-MLP result obtained after correcting the parameters of the neural network.

The observations made on the figures are also reflected in the numerical results presented in Tables 1 and 2. As we can read from it, the proposed CSCS-MLP method achieves the highest value of balanced accuracy in replication-average for each of the *stream-learn* scenarios. In all types of drift dynamics, it is competed only with the AUE method, which demonstrates greater discriminant power than other

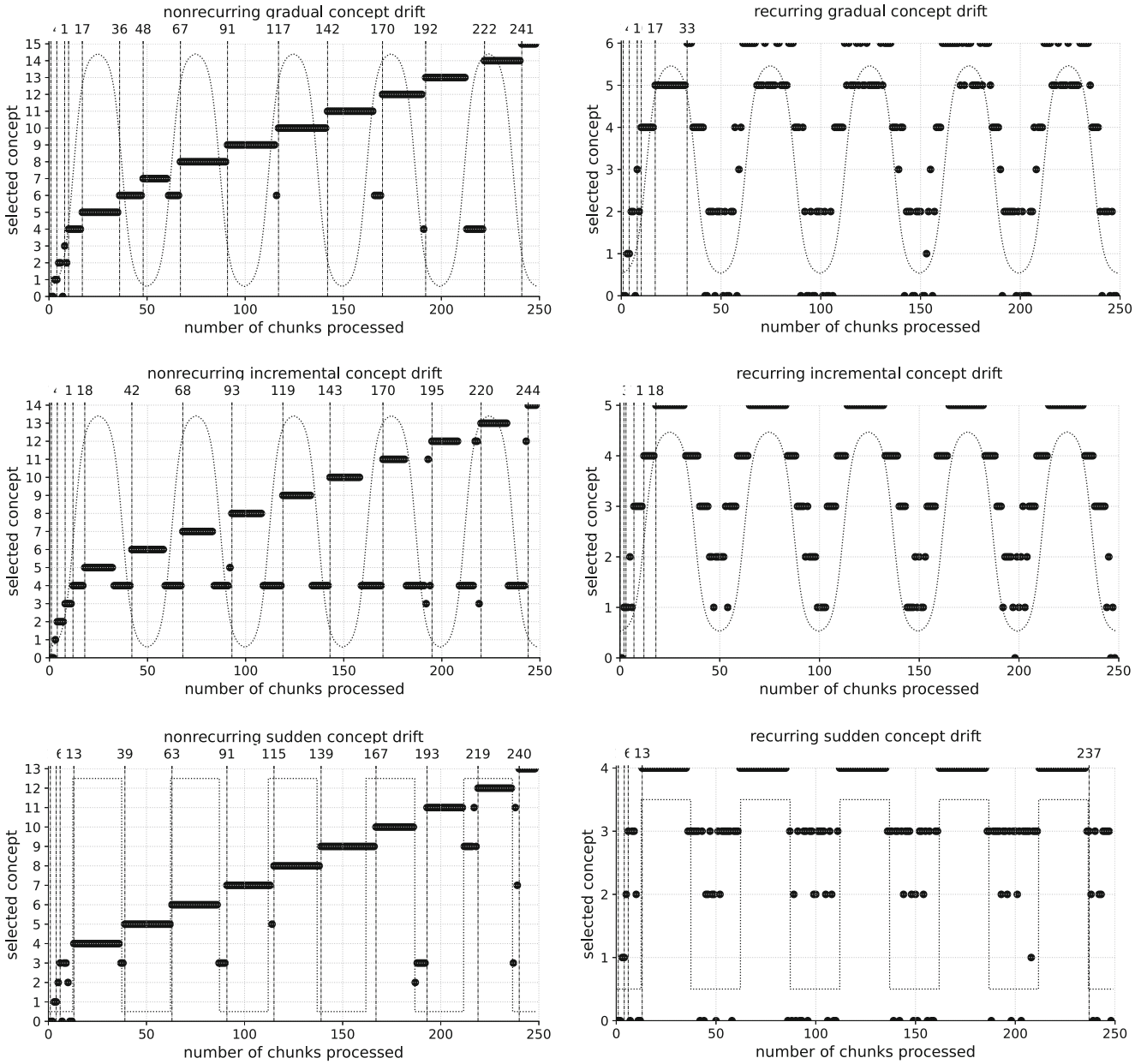


Fig. 13 Concept identification capability in the prediction procedure of the CSCS algorithm for data streams with synthetic concepts generated by *stream-learn* package

methods, but is still about seven percent worse than CSCS. In the case of recurrent drift, it is worth noting a fairly high result of the MLP base method, which shows its ability to adapt to several learnable concepts following the *lifelong-learning* paradigm. The SEA and REA methods, due to the lack of a model updating procedure, are not able to use this ability of the base classifier and, despite the ensemble approach, lead to ultimately worse recognition systems. The AUE algorithm, which is already capable of updating its models, allows a slight improvement over MLP. However, it is the CSCS algorithm – by its ability to dynamically switch between concepts – that makes the best use of this MLP property and achieves an average of 92 percent balanced accuracy on a stream with ten concept drifts.

In turn, solutions based on HTC show a higher generalization capacity for data synthesized by MOA generators. The difference here is particularly strong for AGR and RTG streams, in which the MLP was not able to significantly exceed the level of the random classifier. For the RBF generator, after adjusting the parameters of the base model (RBF*), a significant increase in the efficiency of neural networks can be seen, which in this case makes the CSCS-MLP pairing a method competitive with LBC-HTC and ARF-HTC.

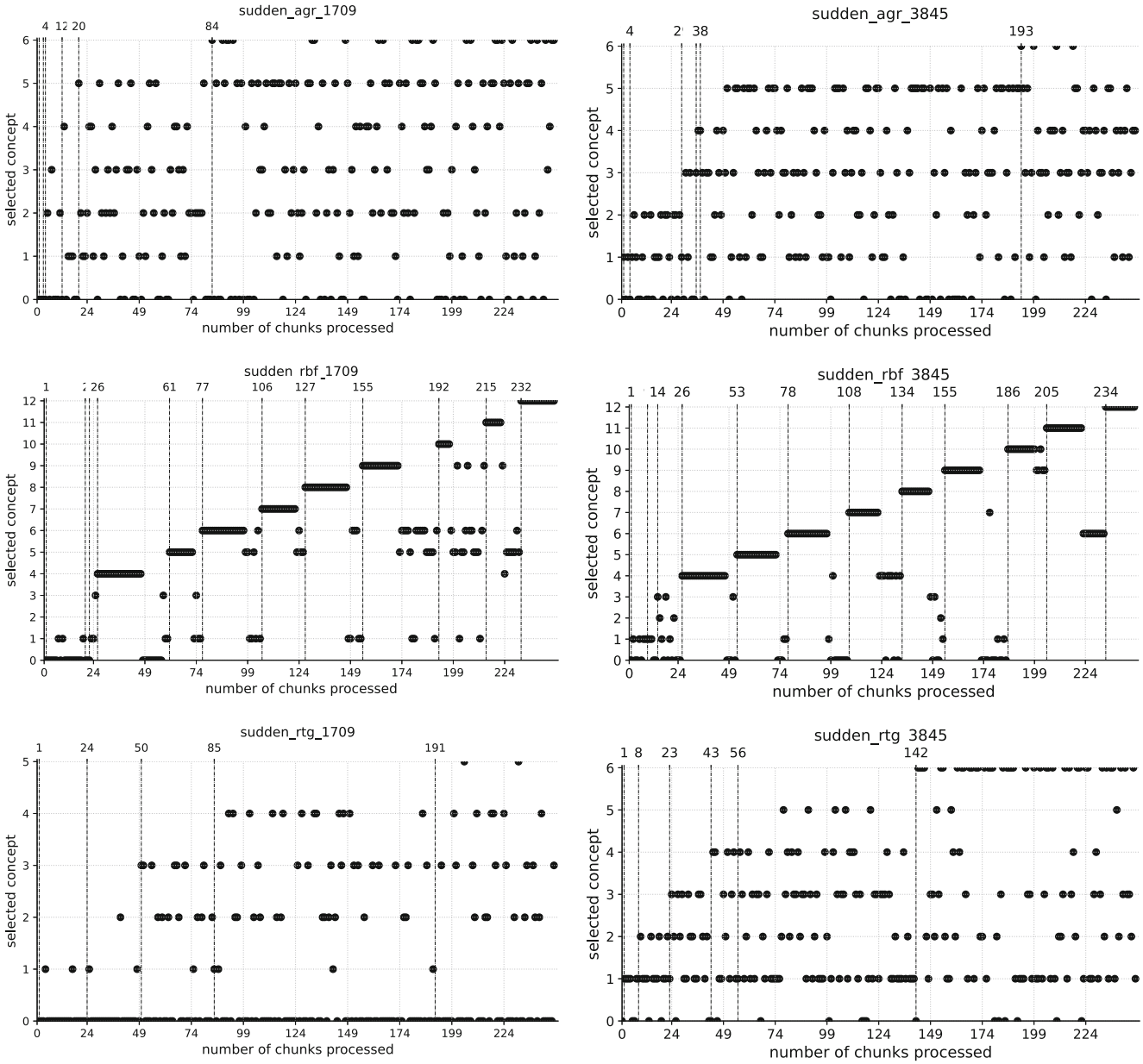


Fig. 14 Concept identification capability in the prediction procedure of the CSCS algorithm for data streams with synthetic concepts generated by *MOA* package

3.4 Experiment 3: evaluation on data streams with real concepts

The final experiment of the conducted evaluation is the analysis of the effectiveness of the compared methods in the processing of data streams with real concepts. Figure 15 presents obtained results analogous to those from Experiment 2. Again the X-axis shows the stream flow, and the Y-axis – accumulative balanced accuracy of the methods.

The basic observation here should be much greater diversity in the quality of various ensemble methods, which no longer group into hard-to-distinguish clusters in the case of real concepts. However, this is only apparent for MLP-based models, where HTC leads to slightly more consistent results. The different dynamics of the initial approach to neural network convergence are particularly interesting here, in which the CSCS algorithm shows the most significant dynamics. However, in some cases, as in the four-dimensional and seven-dimensional problem, the MLP model turns out to be the most effective in the presence of a first concept. However, this state is temporary, and as the concept changes, CSCS is gaining more and more advantages over the competition.

Table 1 Comparative analysis of ensemble methods for classification of data streams with synthetic concepts on *Multi-Layer Perceptron*

	Proposition	Baseline	Ensemble methods					
Drift dype	CSCS	MLP	SEA	REA	AWE	AUE	WAE	KUE
stream-learn generators								
gradual	0.871	0.765	0.770	0.511	0.765	0.810	0.777	0.769
incremental	0.883	0.759	0.796	0.519	0.788	0.816	0.803	0.776
sudden	0.894	0.752	0.778	0.514	0.775	0.814	0.792	0.770
recurring	0.916	0.820	0.781	0.514	0.777	0.843	0.790	0.806
nonrecurring	0.849	0.697	0.782	0.515	0.774	0.784	0.792	0.737
mean	0.883	0.759	0.781	0.514	0.776	0.814	0.791	0.771
MOA generators								
AGR	0.575	0.548	0.503	0.501	0.505	0.564	0.519	0.598
RBF	0.659	0.552	0.565	0.506	0.561	0.587	0.571	0.580
RBF*	0.836	*0.547*	*0.704*	*0.509*	*0.702*	*0.708*	*0.717*	*0.580*
RTG	0.578	0.538	0.544	0.500	0.548	0.570	0.552	0.555
mean	0.604	0.546	0.537	0.502	0.538	0.574	0.547	0.578

Underlined scores are the global best in comparison between MLP and HTC

With the MLP classifier, the AWE, WAE and AUE algorithms represents a uniform front, illustrated by green lines on the plots. They show a significantly higher quality than the KUE algorithm, but also – apart from the initial processing stage – they differ significantly from the CSCS algorithm. When using the HTC classifier, the ARF and LBC algorithms have the highest generalization ability. However, for several components of the stream (chunks 168-192 and 216-240 with four attributes and 192-240 with six attributes), HTC completely loses the potential for recognition, and all models based on it drop significantly to the level of a random classifier.

Interestingly, which was not noticeable in Experiment 2, in the presence of real concepts, the CSCS-MLP algorithm achieves its maximum recognition ability much faster than competing ensemble methods. Its learning curve for each

Table 2 Comparative analysis of ensemble methods for classification of data streams with synthetic concepts on *Hoeffding Tree*

	Prop.	Base.	Ensemble methods							
DT	CSCS	HTC	SEA	REA	AWE	AUE	WAE	KUE	ARF	LBC
stream-learn generators										
gra.	0.760	0.737	0.750	0.507	0.751	0.765	0.759	0.775	0.794	0.843
inc.	0.777	0.750	0.779	0.508	0.779	0.794	0.788	0.800	0.807	0.851
sud.	0.786	0.744	0.755	0.508	0.755	0.789	0.775	0.794	0.805	0.843
rec.	0.801	0.778	0.761	0.508	0.761	0.781	0.773	0.792	0.811	0.852
non.	0.748	0.709	0.762	0.508	0.763	0.784	0.775	0.788	0.794	0.840
mean	0.774	0.744	0.762	0.508	0.762	0.783	0.774	0.790	0.802	0.846
MOA generators										
AGR	0.799	0.901	0.721	0.502	0.727	0.895	0.678	0.872	0.890	0.979
RBF	0.675	0.708	0.675	0.511	0.675	0.725	0.685	0.756	0.840	0.865
RTG	0.559	0.587	0.608	0.501	0.581	0.634	0.618	0.634	0.636	0.684
mean	0.678	0.732	0.668	0.505	0.661	0.751	0.660	0.754	0.788	0.843

Underlined scores are the global best in comparison between MLP and HTC

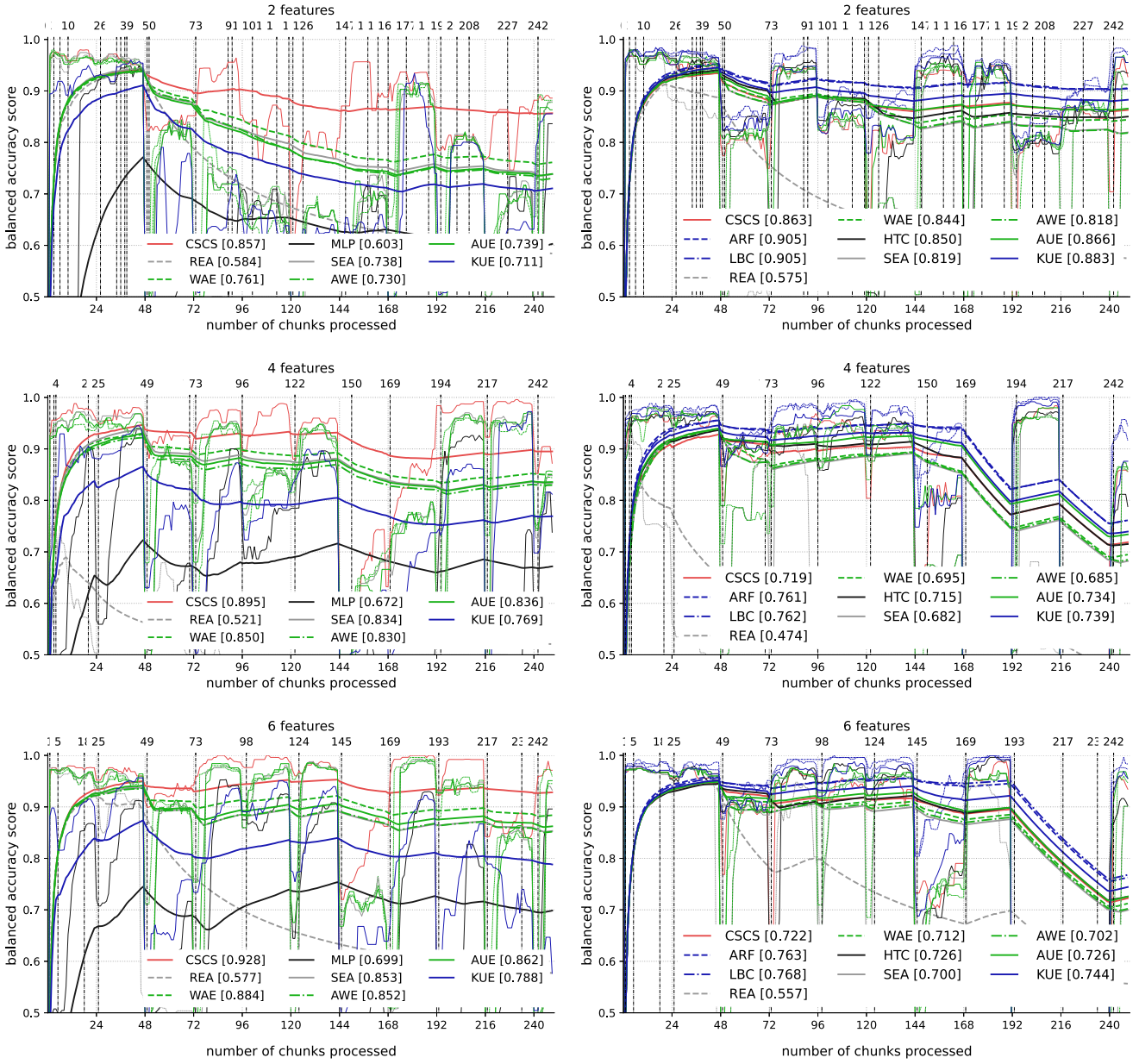


Fig. 15 *Balanced accuracy score* achieved by compared methods on runs from processing scenarios with real concepts

of the concepts contained in the stream is noticeably more dynamic and allows for faster achievement of neural network convergence. In most of the analyzed streams, the CSCS-MLP pairing is better than the other ensemble methods almost from the very beginning of processing, even despite the earlier problems in distinguishing between concepts.

Relatively low-dimensional real concepts also make it possible to evaluate the effectiveness of concept detection depending on the size of the representation of the chunk signature. Appropriate analysis can be made based on Fig. 16 showing the ability to identify the concept depending on the dimensionality of the stream. Problems with a small number of attributes build a small covariance matrix, so they should also hinder the proper recognition of their properties by the CSCS algorithm.

As can be seen in the case of the problem with two attributes, the CSCS algorithm is characterized by the high variation in concept identification, relatively often deciding to "jump" between models. However, the training of all models recognized as consistent with the current concept does not affect the final quality of recognition. Its only influence is observable in reducing the dynamics of the learning curve of such a recognition system. Moreover, it is done while still achieving a significantly better prediction than the competition. In the case of problems with four to six attributes, the recognition is unambiguous after the initial

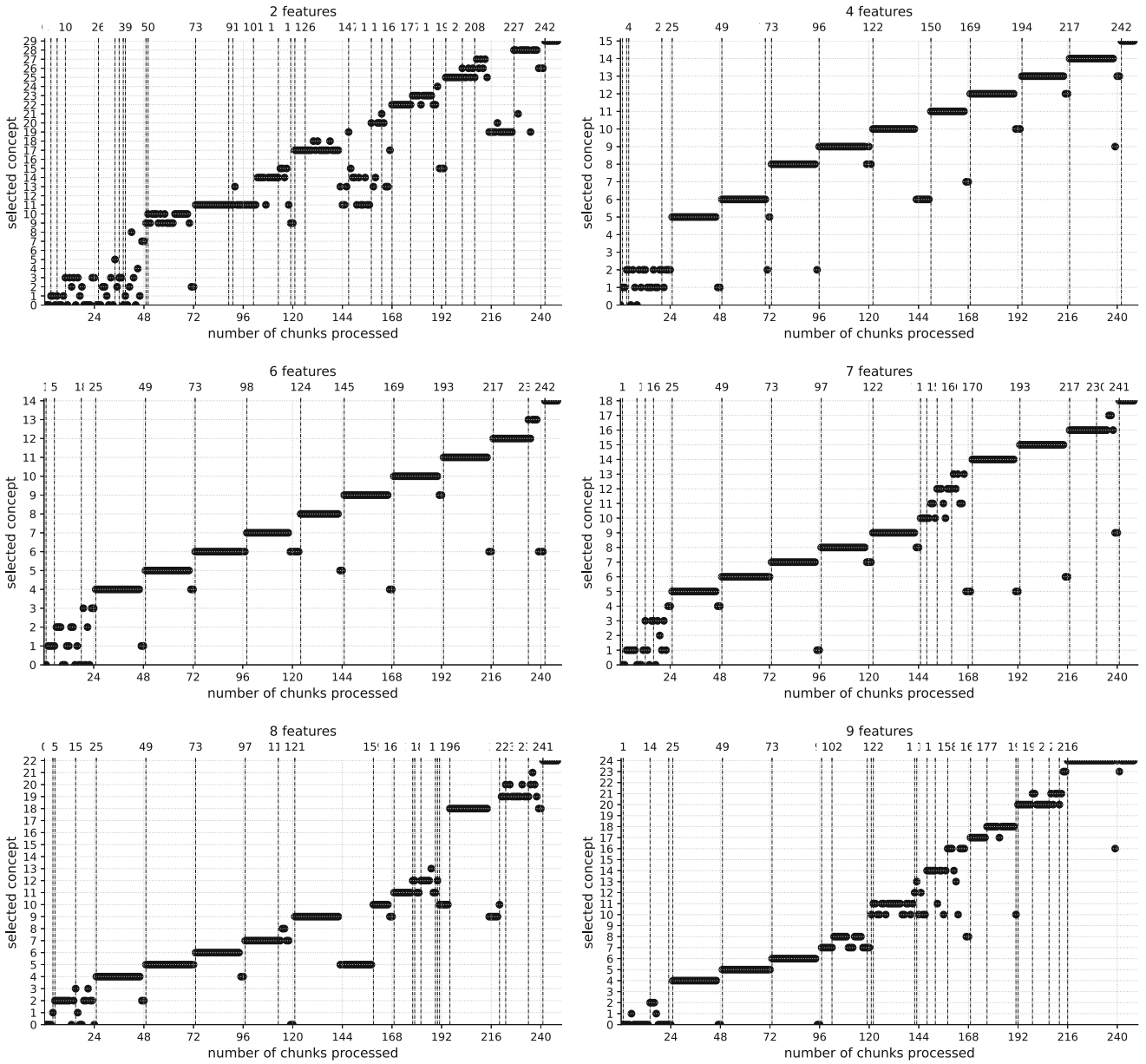


Fig. 16 Concept identification capability in the prediction procedure of the CSCS algorithm for data streams with real concepts

phase and allows a quick switch to a new concept. Apparent difficulties occur at a later stage of processing in the problems with seven, eight, and nine attributes. However, they are mainly due to the inclusion in the stream of concepts with similar dependencies between the attributes. The basis of which was the datasets available in the KEEL repository, diversified by combining classes from a multi-class problem into a dichotomy. However, it does not change the fact that also, in these cases, the CSCS algorithm shows a significant advantage in the quality of recognition over the competition.

This is confirmed by the numerical results contained in the Tables 3 and 4. Methods dedicated to HTC (ARF and LBC) achieve the best results only for low-dimensional data consisting of two or three attributes. For each of the other cases, the balanced accuracy achieved by CSCS-MLP turns out to be much higher than for *state-of-the-art* methods. The difference between other MLP models is, in some cases, more than 10 percent in favor of CSCS, and with the extreme differences between MLP and HTC (7 attributes), it is almost 20 percent.

Table 3 Comparative analysis of ensemble methods for classification of data streams with real concepts on *Multi-Layer Perceptron*

	Proposition	Baseline	Ensemble methods					
N. of features	CSCS	MLP	SEA	REA	AWE	AUE	WAE	KUE
2	0.860	0.605	0.741	0.587	0.733	0.742	0.764	0.713
3	0.915	0.642	0.806	0.574	0.803	0.809	0.833	0.749
4	0.899	0.674	0.837	0.523	0.833	0.839	0.853	0.772
5	0.908	0.635	0.814	0.549	0.819	0.826	0.856	0.781
6	0.932	0.701	0.856	0.580	0.855	0.866	0.887	0.791
7	0.919	0.649	0.841	0.574	0.836	0.850	0.869	0.767
8	0.913	0.659	0.830	0.596	0.833	0.840	0.861	0.782
9	0.902	0.634	0.807	0.557	0.808	0.811	0.838	0.747
mean	0.906	0.650	0.817	0.567	0.815	0.823	0.845	0.763

Underlined scores are the global best in comparison between MLP and HTC

Globally, when analyzing streams with real concepts in the cross-section from 2 to 9 dimensions, the CSCS-MLP turns out to be the best solution, which achieves 91 percent balanced accuracy. Subsequent methods are WAE-MLP reaching 85 percent and the AUE-MLP group, LBC-HTC, SEA-MLP, ARF-HTC, and AWE-MLP averaging about 82 percent balanced accuracy.

4 Discussion

The evaluation carried out in three experiments makes it possible to substantiate the statement about the advantage of updating the models already available in the pool of classifiers over the obligatory building of a new model on each new data chunk for a broad group of problems described by quantitative features. Importantly, this allows for a significant reduction in the processing time (*E1*) with a simultaneous advantage in the quality of prediction resulting from ignoring obsolete models, which was observed both for synthetic streams (*E2*) and for streams with real concepts (*E3*). Despite the increase in the calculation time of the base model for the processing of streams with real concepts, the CSCS method – even in such a case – is characterized by significantly lower complexity than the current *state-of-the-art* solutions (*E1*).

The approach proposed in the CSCS algorithm – based on keeping obsolete models in a pool of classifiers – is a very promising strategy for recurrent concept drift. Using the neural network as the underlying processing model allows to use of its natural property to adapt to the problem, and by identifying the concept, it does not desensitize the base model to subsequent drifts [56]. In addition, this approach also allows for the potential use of training strategies using classifiers that do not natively follow the drift of the concept. Finally, referring to Wolpert's theorem, it should be noted

Table 4 Comparative analysis of ensemble methods for classification of data streams with real concepts on *Hoeffding Tree*

	Prop.	Baseline	Ensemble methods							
d	CSCS	MLP	SEA	REA	AWE	AUE	WAE	KUE	ARF	LBC
2	0.867	0.854	0.823	0.577	0.822	0.869	0.847	0.886	0.909	0.908
3	0.904	0.893	0.865	0.566	0.866	0.907	0.876	0.911	0.939	0.938
4	0.722	0.718	0.685	0.476	0.687	0.737	0.698	0.742	0.764	0.765
5	0.810	0.810	0.750	0.531	0.748	0.815	0.777	0.827	0.853	0.858
6	0.725	0.729	0.702	0.559	0.705	0.729	0.715	0.747	0.766	0.771
7	0.688	0.688	0.678	0.430	0.674	0.696	0.685	0.705	0.726	0.728
8	0.791	0.769	0.768	0.509	0.766	0.795	0.784	0.805	0.836	0.833
9	0.686	0.679	0.674	0.531	0.674	0.697	0.693	0.718	0.746	0.747
mean	0.774	0.768	0.743	0.522	0.743	0.780	0.760	0.793	0.817	0.818

Underlined scores are the global best in comparison between MLP and HTC

that this method allows the potential extension of stream processing applications to problems in which *Hoeffding trees* have significant difficulties with achieving high and stable generalization abilities.

5 Conclusion

The *Covariance-signature Concept Selector* algorithm – proposition of this article – allowed, through the integration of paradigms typical for ensemble learning and drift detectors, to construct models characterized by greater efficiency in the data stream classification than methods that are currently *state-of-the-art* in the field in the recognized group of problems with quantitative features. However, this does not change that it is a solution characterized by several elements that constitute a potential development area.

In the first place, CSCS algorithm allows for efficient concept identification only for streams where drifts occur. In the case of stationary streams, typical of incremental learning, any CSCS identification of the concept will only lead to a slowdown of the training procedure. Therefore, the proposed algorithm cannot be considered recommended in the above situations. Similarly, the CSCS is not an optimal solution in the initial stage of processing, when only one stable concept is available. Instead, it should be recommended for data streams with relatively frequent concept changes and streams with recursive concept drift.

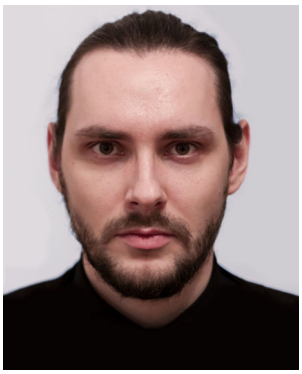
Additionally, in future work, it is possible to further reduce the processing time of the CSCS algorithm to or even below the SEA level by changing its implementation and introducing rather a matrix than the iterative computation of the variance of differences in the covariance matrix. Among additional future work, it will also be worth considering a proposed method for classifying *slow-drift* data streams. Due to its ability to recognize the transitional stages of a concept, this method could work for sets that have only one – gradual or incremental – concept drift starting its dynamics in the first chunk and ending in the last chunk.

Acknowledgements This work was supported by the Polish National Science Centre under the grant No. 2017/27/B/ST6/01325 as well by the statutory funds of the Department of Systems and Computer Networks, Faculty of Electronics, Wroclaw University of Science and Technology.

References

1. Alpaydin E (2020) Introduction to machine learning. MIT press
2. Wu Y, Chen Y, Wang L, Ye Y, Liu Z, Guo Y, Fu Y (2019) Large scale incremental learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 374–382
3. Köppen M (2000) The curse of dimensionality. In: 5th Online World conference on soft computing in industrial applications (WSC5), vol 1, pp 4–8
4. Khalid S, Khalil T, Nasreen S (2014) A survey of feature selection and feature extraction techniques in machine learning. In: 2014 Science and information conference. IEEE, pp 372–378
5. Ienco D, Bifet A, Žliobaitė I, Pfahringer B (2013) Clustering based active learning for evolving data streams. In: International conference on discovery science. Springer, pp 79–93
6. Berthelot D, Carlini N, Goodfellow I, Papernot N, Oliver A, Raffel C (2019) Mixmatch: A holistic approach to semi-supervised learning. arXiv:1905.02249
7. Zhou L, Pan S, Wang J, Vasilakos AV (2017) Machine learning on big data: opportunities and challenges. Neurocomputing 237:350–361
8. Žliobaitė I (2010) Learning under concept drift: an overview. arXiv:1010.4784
9. Gaber MM, Zaslavsky A, Krishnaswamy S (2007) A survey of classification methods in data streams. Data Streams, 39–59
10. Sobolewski P, Woźniak M (2013) Comparable study of statistical tests for virtual concept drift detection. In: Proceedings of the 8th international conference on computer recognition systems CORES 2013. Springer, pp 329–337
11. Ksieniewicz P (2021) The prior probability in the batch classification of imbalanced data streams. Neurocomputing 452:309–316
12. Komorniczak J, Zyblewski P, Ksieniewicz P (2021) Prior probability estimation in dynamically imbalanced data streams
13. Grzyb J, Klikowski J, Woźniak M (2021) Hellinger distance weighted ensemble for imbalanced data stream classification. J Comput Sci 51:101314
14. Ghazikhani A, Monsefi R, Yazdi HS (2013) Recursive least square perceptron model for non-stationary and imbalanced data stream classification. Evolv Syst 4(2):119–131
15. Zyblewski P, Sabourin R, Woźniak M (2019) Data preprocessing and dynamic ensemble selection for imbalanced data stream classification. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, pp 367–379
16. Gama J (2012) A survey on learning from data streams: current and future trends. Progress Artif Intell 1(1):45–55
17. Manoj Kumar MV, Thomas L, Annappa B (2015) Capturing the sudden concept drift in process mining. Algorithms & theories for the analysis of event data (ATAED'15, Brussels, Belgium, June 22-23, 2015), p 132
18. Brzezinski D, Stefanowski J (2013) Reacting to different types of concept drift: the accuracy updated ensemble algorithm. IEEE Trans Neural Netw Learn Syst 25(1):81–94
19. Liu A, Zhang G, Lu J (2017) Fuzzy time windowing for gradual concept drift adaptation. In: 2017 IEEE International conference on fuzzy systems (FUZZ-IEEE). IEEE, pp 1–6
20. Krawczyk B, Woźniak M (2015) One-class classifiers with incremental learning and forgetting for data streams with concept drift. Soft Comput 19(12):3387–3400
21. Ramírez-Gallego S, Krawczyk B, García S, Woźniak M, Herrera F (2017) A survey on data preprocessing for data stream mining: Current status and future directions. Neurocomputing 239:39–57
22. Krawczyk B, Minku LL, Gama J, Stefanowski J, Woźniak M (2017) Ensemble learning for data stream analysis: a survey. Inform Fus 37:132–156
23. Kuncheva LI (2004) Classifier ensembles for changing environments. In: International workshop on multiple classifier systems. Springer, pp 1–15
24. Street WN, Kim Y (2001) A streaming ensemble algorithm (sea) for large-scale classification. In: Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining, pp 377–382

25. Wang H, Fan W, Yu PS, Han J (2003) Mining concept-drifting data streams using ensemble classifiers. In: Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining, pp 226–235
26. Brzeziński D, Stefanowski J (2011) Accuracy updated ensemble for data streams with concept drift. In: International conference on hybrid artificial intelligence systems. Springer, pp 155–163
27. Chen S, He H (2011) Towards incremental learning of nonstationary imbalanced data stream: a multiple selectively recursive approach. Evolv Syst 2(1):35–50
28. Woźniak M, Kasprzak A, Cal P (2013) Weighted aging classifier ensemble for the incremental drifted data streams. In: International conference on flexible query answering systems. Springer, pp 579–588
29. Hoeffding W (1963) Probability inequalities for sums of bounded random variables. Amer. ž Statist Assoc J, 1329
30. Muallem A, Shetty S, Pan JW, Zhao J, Biswal B (2017) Hoeffding tree algorithms for anomaly detection in streaming datasets: a survey. J Inf Secur 8:4
31. Hulten G, Spencer L, Domingos P (2001) Mining time-changing data streams. In: Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining, pp 97–106
32. Bifet A, Holmes G, Pfahringer B (2010) Leveraging bagging for evolving data streams. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, pp 135–150
33. Oza NC, Russell SJ (2001) Online bagging and boosting. In: International workshop on artificial intelligence and statistics. PMLR, pp 229–236
34. Bifet A, Gavalda R (2007) Learning from time-changing data with adaptive windowing. In: Proceedings of the 2007 SIAM international conference on data mining. SIAM, pp 443–448
35. Gomes HM, Bifet A, Read J, Barddal JP, Enembreck F, Pfharinger B, Holmes G, Abdessalem T (2017) Adaptive random forests for evolving data stream classification. Mach Learn 106(9):1469–1495
36. Cano A, Krawczyk B (2020) Kappa updated ensemble for drifting data stream mining. Mach Learn 109(1):175–218
37. Gonçalves Jr PM, de Carvalho Santos SilasGT, Barros RobertoSM, Vieira DaviCL (2014) A comparative study on concept drift detectors. Expert Syst Appl 41(18):8144–8156
38. Barros RSM, Santos SGTC (2018) A large-scale comparison of concept drift detectors. Inf Sci 451:348–370
39. Gama J, Medas P, Castillo G, Rodrigues P (2004) Learning with drift detection. In: Brazilian symposium on artificial intelligence. Springer, pp 286–295
40. Baena-Garcıa M, del Campo-Ávila J, Fidalgo R, Bifet A, Gavalda R, Morales-Bueno R (2006) Early drift detection method. In: Fourth international workshop on knowledge discovery from data streams, vol 6, pp 77–86
41. Page ES (1954) Continuous inspection schemes. Biometrika 41(1/2):100–115
42. Alippi C, Roveri M (2006) An adaptive cusum-based test for signal change detection. In: 2006 IEEE international symposium on circuits and systems. IEEE, pp 4–pp
43. Severo M, Gama J (2006) Change detection with Kalman filter and cusum. In: International conference on discovery science. Springer, pp 243–254
44. Srivastava MS, Wu Y (1993) Comparison of Ewma, Cusum and Shiryayev-Roberts procedures for detecting a shift in the mean. Ann Stat, 645–670
45. Micevska S, Awad A, Sakr S (2021) Sddm: an interpretable statistical concept drift detection method for data streams. J Intell Inform Syst 56(3):459–484
46. Bach SH, Maloof MA (2008) Paired learners for concept drift. In: 2008 Eighth IEEE international conference on data mining. IEEE, pp 23–32
47. Bose A, Bhattacharjee M (2018) Large covariance and autocovariance matrices. CRC Press, USA
48. Park KI, Park M (2018) Fundamentals of probability and stochastic processes with applications to communications. Springer
49. Guyon I, Gunn S, Ben-Hur A, Dror G (2004) Result analysis of the nips 2003 feature selection challenge. Advances in Neural Information Processing Systems, 17
50. Ksieniewicz P, Zyblewski P (2020) stream-learn–open-source python library for difficult data stream batch analysis. arXiv:2001.11077
51. Zyblewski P, Sabourin R, Woźniak M (2021) Preprocessed dynamic classifier ensemble selection for highly imbalanced drifted data streams. Inform Fus 66:138–154
52. Hinton GE (1990) Connectionist learning procedures. 555–610
53. Chan TF, Golub GH, LeVeque RJ (1982) Updating formulae and a pairwise algorithm for computing sample variances. In: COMPSTAT 1982 5th symposium held at Toulouse 1982. Springer, pp 30–41
54. Domingos P, Hulten G (2003) A general framework for mining massive data streams. J Comput Graph Stat 12(4):945–949
55. Agrawal R, Imielinski T, Swami A (1993) Database mining: a performance perspective. IEEE Trans Knowl Data Eng 5(6):914–925
56. Ksieniewicz P, Woźniak M, Cyganek B, Kasprzak A, Walkowiak K (2019) Data stream classification using active learned neural networks. Neurocomputing 353:74–82



Pawel Ksieniewicz received the M.Sc. and Ph.D. degrees from the Wroclaw University of Science and Technology in 2013 and 2017, respectively. He is an Assistant Professor with the Wroclaw University of Science and Technology. Most of his papers concern classification of difficult data, focusing on processing data streams, multidimensional data representation, imbalanced data, and image processing.

[C3]
Joanna Komorniczak, Pawel Zyblewski i Pawel Ksieniewicz. “Prior Probability Estimation in Dynamically Imbalanced Data Streams”. W: *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, lip. 2021. DOI: 10.1109/ijcnn52387.2021.9533795

Prior Probability Estimation in Dynamically Imbalanced Data Streams

1st Joanna Komorniczak
Department of Systems and Computer Networks
Wroclaw Univ. of Science and Technology
Wrocław, Poland
e-mail: 241245@student.pwr.edu.pl
ORCID: 0000-0002-1393-3622

2nd Paweł Zyblewski
Department of Systems and Computer Networks
Wroclaw Univ. of Science and Technology
Wrocław, Poland
e-mail: pawel.zyblewski@pwr.edu.pl
ORCID: 0000-0002-4224-6709

3st Paweł Ksieniewicz
Department of Systems and Computer Networks
Wroclaw Univ. of Science and Technology
Wrocław, Poland
e-mail: pawel.ksieniewicz@pwr.edu.pl
ORCID: 0000-0001-9578-8395

2021 International Joint Conference on Neural Networks (IJCNN) | 978-1-6654-3900-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/IJCNN52387.2021.9533795

***Abstract*—Despite the fact that real-life data streams may often be characterized by the dynamic changes in the *prior class probabilities*, there is a scarcity of articles trying to clearly describe and classify this problem as well as suggest new methods dedicated to resolving this issue. The following paper aims to fill this gap by proposing a novel data stream taxonomy defined in the context of *prior class probability* and by introducing the *Dynamic Statistical Concept Analysis* (DSCA) – *prior probability* estimation algorithm. The proposed method was evaluated using computer experiments carried out on 100 synthetically generated data streams with various class imbalance characteristics. The obtained results, supported by statistical analysis, confirmed the usefulness of the proposed solution, especially in the case of *discrete dynamically imbalanced data streams* (DDIS).**

***Index Terms*—data stream, imbalanced data, dynamically imbalanced data stream, pattern recognition**

I. Introduction

The modern world is often described by both fiction [1] and scientific authors as a never-ending, diverse data stream controlling our lives. The reality increasingly confirms this supposition in the era of the digitization of our everyday life – accelerated by the coronavirus pandemic – in which most of our interpersonal contacts, cultural works we receive, our financial transactions or even home lighting control are carried out with streaming information sent via computer networks [2].

The data stream processing is a problem widely discussed in the literature [3]. The two main difficulties recurring in the analyzes are (*a*) unusually large data volumes, which forces the authors of processing algorithms to adapt to the rule of single processing of each pattern coming from a data stream [4], [5], and (*b*) the constantly changing probabilistic characteristics of the data stream, most often interpreted as a *concept drift* phenomenon, involving changes in the *posterior* class distributions in the time domain [6].

The vast majority of works in the field of non-stationary data stream processing deal with problems of changes in the *posterior* probability [7], relatively rarely addressing the topic of imbalanced streams, and in particular, dynamically imbalanced streams, i.e. those characterized by changes in the *prior probability* [8]. Meanwhile, a large part of data streams – with network streams as a representative example – can be characterized primarily by a huge volume in a relatively short period, which means that the influence of concept drift is stretched over time, reduced and sometimes even negligible. In such cases, changes in the prior class distribution become much more important [9].

The above-mentioned data may therefore be defined as static in terms of the concept, with dynamic proportion of individual classes, depending on the point in time at which we make the snapshot. The very type of differences in *prior probability* can also vary. There exists possible cases of problems that are globally balanced (on the entire analyzed data stream course), but are characterized by a strong disproportion of the class count in local data portions (*batches*).

Information about the prior distribution of problem classes may have a real impact on the quality of classification models built in the stream environment. The usefulness of such knowledge in statistically significant improvement of the imbalanced data stream classification has been substantiated in [10]. However, the mentioned work deals with only one simple method of *prior probability* estimation, which, cannot be generalized to all types of imbalanced data streams. We would like to propose the following taxonomy of data streams in the context of *prior class probability* (presented also in Figure 1):

- **BS** (*balanced streams*)
  The global prior and each of the local priors has a **proportional** and **dependent** class distribution,

978-0-7381-3366-9/21/$31.00 ©2021 IEEE

- **SIS** (*statically imbalanced streams*)
  The global prior and each of the local priors are characterized by a **disproportionate** but **constant and dependent** class distribution,
- **DIS** (*dynamically imbalanced streams*)
  Among which we can distinguish two subcategories:
  - **CDIS** (*continuous dynamically imbalanced streams*)
    The global prior **may differ** from local priors, whose class distribution is **independent**, but changes **continuously**, allowing for the observation of trends in its changes,
  - **DDIS** (*discrete dynamically imbalanced streams*)
    The global prior **may differ** from local priors, whose class distribution is **independent** and changes **discreetly**, making it impossible to observe trends in its changes.

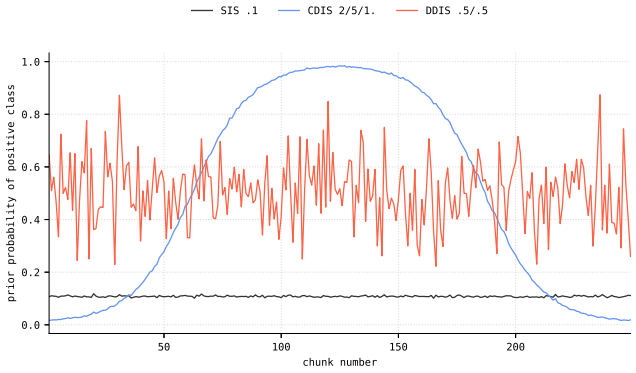


Fig. 1. Positive class *prior probabilities* for each data chunk of the streams from each category from introduced taxonomy.

The main contributions of this work are as follows:

- The proposition of a novel data stream taxonomy in the context of *prior class probability*.
- The proposition of an optimization method of *prior probability* estimation in dynamically imbalanced data streams from all four distinguished categories, based on multi-criteria regression using neural networks.
- Experimental evaluation of the proposed DSCA algorithm based on diverse data streams and a comparison with the *state-of-art* methods.

II. Dynamic Statistical Concept Analysis

This chapter describes the structure of *Dynamic Statistical Concept Analysis* (DSCA) model proposed in the work, which performs the *prior probability* prediction through *representation learning* [11].

Dynamic Statistical Concept Analysis is a novel method for determining *prior probabilities* in binary imbalanced data stream, designed primarily for DIS. It uses two *Multi-layer Perception (MLP) regressors* that are able to detect relationships between statistical characteristics of a data chunk, such as the *mean value* and *standard deviation* of the attributes in the context of processed chunk, and the class proportions.

The entire DSCA procedure is described in Algorithm 1. The model obtains information about the data chunk $\mathcal{DS}_k$ for the currently processed, k^{th} data chunk. Basic statistical characteristics is collected from each data portion – such as the *mean value* of the features and their *standard deviation*. This conglomerate contains approximated and reduced information about the entire batch, constructing r – a new *representation* of chunk.

Algorithm 1 DSCA pseudocode

Input:
Stream of data chunks $\{\mathcal{DS}_1, \mathcal{DS}_2, \ldots, \mathcal{DS}_k\}$,
Symbols:
$\mathcal{DS}_k$ – data chunk,
r – data chunk representation,
$\mathcal{R}$ – incremental chunk representation dataset,
P – real prior probability,
P' – predicted prior probability,
e – regressor epoch limit,
$\mathcal{C}_j^k$ – number of samples from class j in k^{th} chunk,
$\mathcal{C'}_j^k$ – predicted number of samples from class j in k^{th} chunk,
w – window width

$\mathcal{R} \leftarrow \varnothing$
$reg_0 \leftarrow$ cloned base regressor for negative class
$reg_1 \leftarrow$ cloned base regressor for positive class
for all data chunk $\mathcal{DS}_k \in Stream$ **do**
  $P_k \leftarrow \mathcal{C}_0^k \div (\mathcal{C}_0^k + \mathcal{C}_1^k)$ ▷ Real prior
  $r \leftarrow [mean(\mathcal{DS}_k), std(\mathcal{DS}_k)]$ ▷ Representation
  if $\mathcal{R} = \varnothing$ **then**
    $iter \leftarrow e$
  else
    $\mathcal{C'}_0^k \leftarrow reg_0(r)$ ▷ Estimated class count
    $\mathcal{C'}_1^k \leftarrow reg_1(r)$
    $P'_k \leftarrow \mathcal{C'}_0^k \div (\mathcal{C'}_0^k + \mathcal{C'}_1^k)$ ▷ Estimated prior
    $err \leftarrow |P_{k-1} - P'_k|$
    $iter \leftarrow e \times err$
  end if
  $\mathcal{R} \leftarrow \mathcal{R} + r$
  $R \leftarrow$ last w of $\mathcal{R}$
  $C \leftarrow$ last w of $\mathcal{C}$
  while $iter > 0$ **do**
    $reg_0 \leftarrow reg_0(R, C_0)$ ▷ Model update
    $reg_1 \leftarrow reg_1(R, C_1)$
    $iter \leftarrow iter - 1$
  end while
end for

For subsequent chunks of the data stream, (*i*) representation r extracted from the $\mathcal{DS}_k$ ($\mathcal{R}$), (*ii*) real and predicted *prior probabilities* (P and P') and (*iii*) real and predicted counts of samples from each of the problem classes (C and C') are stored. The model uses regressors, trained with the last

w *representations* R, and the corresponding class counts C^k respectively for negative and positive class. The window width w is set by the parameter.

The model works for streams processed with the *test-then-train* protocol, which means that for currently processed chunk $\mathcal{DS}_k$ a prior prediction P' was made in the previous step. Depending on the error err between the prediction and the true proportions between the classes, the number of regressor training iterations $iter$ is calculated – the bigger the err variable, the more epochs. Upper limit of iterations e is set by the parameter. For the first data chunk, regressors are trained to the maximum number of epochs. At the beginning of stream processing, errors will be larger, which will result in longer training time, however, after a certain amount of data processed, errors will decrease and therefore will the training time. Such dependency is presented in Figure 2.

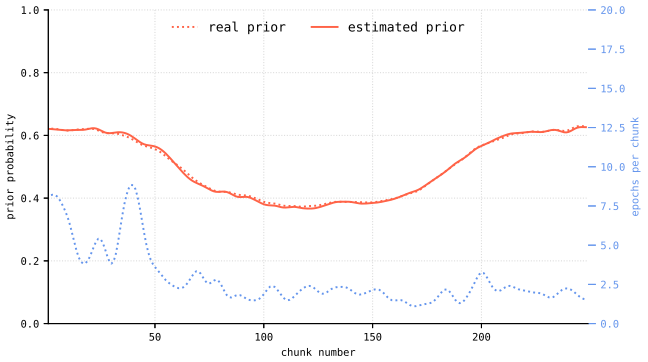


Fig. 2. Predicted and real *prior probability* for each data chunk of the stream with number of DSCA training epochs per given chunk.

During *prior probability* prediction procedure, described by lines 10-12 of Algorithm 1, where only feature values of the $\mathcal{DS}_k$ set are known. The responses of both regressors are used as predictions of individual classes count (C') in the current representation r, finally determining the predicted prior probability (P') as the proportions between the predicted number of the negative class samples and the predicted sum of the patterns in $\mathcal{DS}_k$.

III. Experiments design

The following section presents the research hypotheses, goals of the planned experiments, as well as the set-up of the conducted research.

a) Research hypotheses*:*

Using the taxonomy proposed in Section 1 and based on the observations from the works published in the field, we would like to verify the following experimental hypotheses:

- **H1** In the BS and SIS categories, it is possible to stabilize the information about the *prior probability* in each successive chunk by analyzing the distribution of classes in historical data (from the batches processed so far).
- **H2** In the CDIS category, it is possible to estimate the information about the *prior probability* in each successive chunk by analyzing the trend of the class distribution in historical data.
- **H3** In the DDIS category – due to the discrete changes in the class distribution – the estimators efficient in the remaining categories (BS, SIS and CDIS) will indicate low predictive ability.
- **H4** It is possible to propose a method of efficient prior probability estimation for the DDIS category.

In the experimental evaluation we will try to validate hypotheses H1-3 and verify the effectiveness of the proposed method in order to validate the hypothesis H4.

b) Goals of the experiments*:*

In order to verify the presented research hypotheses, two separate experiments were designed:

Experiment 1 – Hyperparameter optimization

The aim of the first experiment was to optimize hyperparameter n of the reference *prior probability* estimation methods, which denotes the number of previous data chunks, that are taken into consideration during the prior estimation process. The four parameterized methods were:

- **Linear** – methods dedicated to BS i SIS streams:
  - **Mean** – calculates the mean *prior probability* of n previous data chunks.
  - **Linear** – calculates the weighted mean *prior probability* of n previous data chunks. The newer the data chunk is, the more weight it is given.
- **Regressive** – methods dedicated to CDIS streams:
  - **Linear Regression (LR)** – with default hyperparametrization.
  - **Random Forest Regression (RF)** – with 100 trees.

Three n values were examined – consecutively 5, 50 and 250 (all) data chunks.

Experiment 2 – Comparative analysis

The aim of the second experiment was to compare the behavior of the proposed DSCA algorithm to the previously parameterized methods. One additional reference approach, called **Previous** is introduced, which assigns the current data chunk the same *prior probability* as in the previous one.

c) Experimental set-up*:*

To evaluate the proposed method performance, a total of 100 synthetic binary data streams with static concept were generated using the *stream-learn* package. Each data stream was composed of two hundred and fifty thousand instances (250 data chunks, 1000 instances each) described by 8 informative attributes and contained label noise at the level of 1%. During the generation process, 10 different stream types were distinguished, based on the (*a*) *prior class probability* taxonomy introduced in Section 1 and (*b*) class imbalance characteristics.

- **BS** (*balanced streams*) – a single data stream with balanced class distribution,

- **SIS** (*statically imbalanced streams*):
  - *the imbalance ratio* (IR) – successively 2.5, 5 and 10% of minority class,
- **CDIS** (*continuous dynamically imbalanced streams*):
  - *number of drifts* – 2,
  - *concept sigmoid spacing* – 5,
  - IR *amplitude* – successively 25, 50 and 100%,
- **DDIS** (*discrete dynamically imbalanced streams*):
  - *mean value* – 50%,
  - *standard deviation* – successively 10, 25 and 50%,

Additionally, each of the above-mentioned data streams was replicated 10 times with different *random states* to allow for a statistical analysis of obtained results using the *Student's t-test* [12].

As the conducted experiments deal with the *prior probability* estimation, the evaluation is based mainly on the *mean absolute error* (MAE) [13] regression loss calculated using *test-then-train* evaluation protocol [14]. Additionally, to facilitate the visual analysis of obtained results, figures presenting the accumulated difference between the actual and estimated *prior probability* value in each data chunk are presented. All experiments were implemented in the Python programming language, based on the *scikit-learn* [15] and *stream-learn* [16] API's, and can be replicated according to the code published on the *GitHub* repository[1].

IV. Experimental evaluation

This section presents the results of the conducted experiments. The tables present the results of a statistical analysis based on MAE and performed using *Student's t-test* ($p = .05$) on 10 stream replications. The numbers below the average MAE value indicate which of the n parameter values (Table I) or *prior probability* estimation methods (Table II) performed statistically significantly worse than the one in question.

A. Experiment 1 – Hyperparameter optimization

The results of *Experiment 1* are presented in Table I. It is noticeable that the effective value of *n* depends on the stream type:

- For *statically imbalanced streams* (SIS) and *balanced streams* (BS) the results present similar, due to the fact that the streams are stable, the greatest errors occur at the value of *n* equal to 1, which takes into account only previous data chunk. In case of these streams, 50 previous *prior probabilities* of data are sufficient to determine a stable *prior probability* level, hence no statistically significant improvement between *n* value equal to 50 and the analysis of all previous chunks.
- For *continuous dynamic imbalanced streams* (CDIS), the best results occurred for *n* equal to 1, where only proportions in classes of previous chunk were analyzed. This is due to the high instability of the stream while maintaining continuity. *Prior probability* in the analyzed chunk will be relatively close to the previous value. For RF method in streams with an amplitude of 25% and 50%, there was no statistically significant difference in the operation of the prediction.
- For *discrete dynamic imbalanced streams* (DDIS), the optimal *n*-value is the total number of chunks. Since the stream in the whole domain is stable and large fluctuations in *prior probabilities* occur between successive chunks, *n* equal to 1 will not operate well, and *n* equal to 50 still does not give a statistically better result than the analysis of all past *prior* values.

[1]https://github.com/w4k2/stream-dsca

TABLE I
Results of Experiment 1: average mean absolute error for each of tested *n* value

	5 (1)	50 (2)	All (3)	5 (1)	50 (2)	All (3)
	BS			SIS .03		
Mean	0.014 —	0.013 1	0.013 1	0.005 —	0.005 —	0.005 —
Lin	0.014 —	0.013 1	0.013 1	0.005 —	0.005 —	0.005 —
LR	0.021 —	0.014 1	0.014 1	0.008 —	0.005 1	0.005 1
RF	0.016 —	0.016 —	0.016 —	0.005 —	0.006 —	0.005 —
	SIS .05			SIS .10		
Mean	0.006 —	0.006 —	0.006 1	0.009 —	0.008 1	0.008 1
Lin	0.007 —	0.006 1	0.006 1	0.009 —	0.008 1	0.008 1
LR	0.010 —	0.006 1	0.006 1	0.013 —	0.009 1	0.008 1
RF	0.007 —	0.007 —	0.007 —	0.010 —	0.010 —	0.009 —
	CDIS 2/5/.25			CDIS 2/5/.50		
Mean	0.015 all	0.050 3	0.084 —	0.017 all	0.097 3	0.167 —
Lin	0.015 all	0.036 3	0.070 —	0.016 all	0.067 3	0.138 —
LR	0.021 3	0.020 all	0.075 —	0.020 all	0.032 3	0.146 —
RF	0.016 —	0.016 —	0.016 —	0.015 —	0.015 —	0.015 —
	CDIS 2/5/1.			DDIS .5/.1		
Mean	0.024 all	0.192 3	0.332 —	0.086 —	0.080 1	0.079 all
Lin	0.019 all	0.131 3	0.274 —	0.088 —	0.080 1	0.080 all
LR	0.015 all	0.044 3	0.226 —	0.131 —	0.085 1	0.083 all
RF	0.014 all	0.015 —	0.015 —	0.096 —	0.096 —	0.095 all
	DDIS .5/.25			DDIS .5/.5		
Mean	0.209 —	0.194 1	0.193 all	0.330 —	0.312 1	0.311 all
Lin	0.212 —	0.194 1	0.193 all	0.334 —	0.313 1	0.312 1
LR	0.300 —	0.204 1	0.200 all	0.407 —	0.323 1	0.318 all
RF	0.231 —	0.231 —	0.229 all	0.357 —	0.357 —	0.354 all

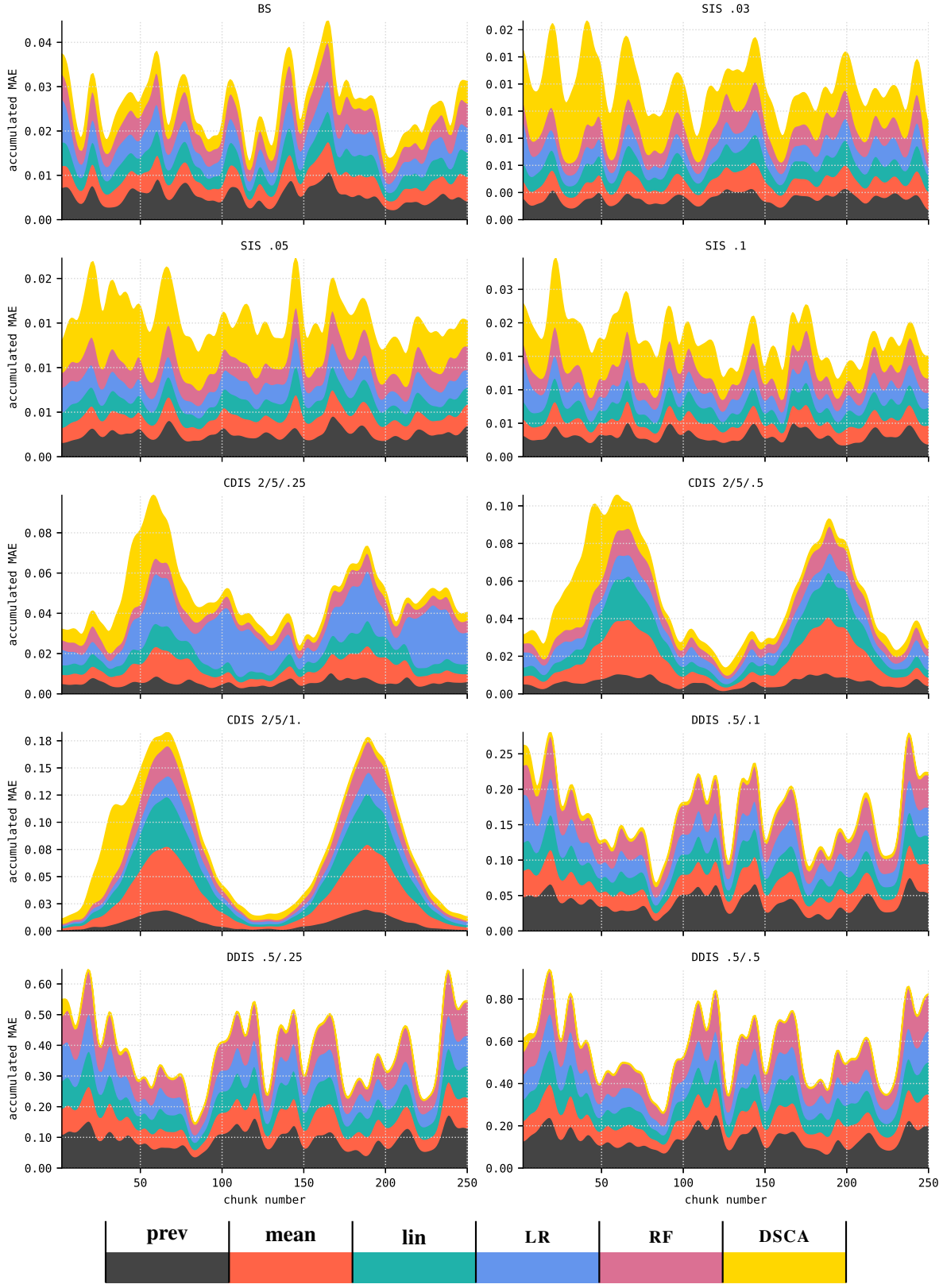


Fig. 3. Results of *Experiment 2* – accumulated difference between the estimated and actual *prior probability* value in each data chunk throughout stream analysis for all *prior probability* estimation methods and all examined stream types.

TABLE II
RESULTS OF EXPERIMENT 2: AVERAGE MEAN ABSOLUTE ERROR FOR EACH OF *prior probability* ESTIMATION METHODS

	LINEAR			REGRESSIVE		
	PREV (1)	MEAN (2)	LIN (3)	LR (4)	RF (5)	DSCA (6)
BS	0.018	0.013	0.013	0.014	0.016	0.011
	—	1, 4, 5	1, 4, 5	1, 5	1	all
SIS .03	0.006	0.005	0.005	0.005	0.005	0.010
	6	1, 5, 6	1, 5, 6	1, 5, 6	1, 6	—
SIS .05	0.008	0.006	0.006	0.006	0.007	0.013
	6	1, 5, 6	1, 5, 6	1, 5, 6	1, 6	—
SIS .1	0.011	0.008	0.008	0.008	0.010	0.012
	—	1, 5, 6	1, 5, 6	1, 5, 6	1	—
CDIS 2/5/.25	0.018	0.015	0.015	0.020	0.016	0.012
	4	1, 4, 5	1, 4, 5	—	1, 4	all
CDIS 2/5/.5	0.017	0.018	0.016	0.020	0.015	0.012
	2, 4	4	1, 2, 4	—	1, 2, 3, 4	all
CDIS 2/5/1.	0.013	0.026	0.021	0.015	0.015	0.013
	2, 3, 4, 5	—	2	2, 3	2, 3, 4	2, 3, 4, 5
DDIS .5/.1	0.111	0.079	0.079	0.080	0.094	0.010
	—	1, 4, 5	1, 4, 5	1, 5	1	all
DDIS .5/.25	0.267	0.192	0.193	0.193	0.229	0.012
	—	1, 5	1, 5	1, 5	1	all
DDIS .5/.5	0.403	0.310	0.310	0.311	0.354	0.013
	—	1, 5	1, 5	1, 5	1	all

B. Experiment 2 – Comparative analysis

Results of *Experiment 2* are presented in Figure 3. The graphs show the accumulated difference between the estimated and actual *prior probability* value in each data chunk throughout stream analysis for all *prior probability* estimation methods and all examined stream types.

It is noticeable, that the DSCA model learns best during a *prior probability* changes in subsequent chunks, therefore:

- For all *statically imbalanced streams* (SIS) the model does not perform better than simpler, linear methods.
- However, for *balanced streams* (BS), which are characterized by minor changes in *prior probabilities*, DCSA has higher prediction quality than for SIS. Predictive ability results from averaging two random *MLP regressor* models in DSCA that give a balanced prediction.
- For *continuous dynamic imbalanced streams* (CDIS) – the greater the changes in amplitude, the faster DSCA learns. Model's error at the beginning of stream processing is larger and decreases over subsequent data chunks. Such trend is also visible in Figure 2 and affects number of epochs of model's regressors training.
- For *discrete dynamic imbalanced streams* (DDIS) model converges almost instantly.

DSCA performs best for CDIS and DDIS, which seem to be the most difficult for other *prior probability* prediction methods. The final results of *Experiment 2* are also presented in Table II, extending the analysis with statistical tests. DSCA is statistically significantly better than all other tested methods for DDIS and two CDIS with amplitude of 25% and 50%. Results of CDIS with amplitude of 100% do not allow to select best method – DSCA and PREV performed equally well. However, it is worth noticing, that final scores of *prior probability* prediction methods were calculated over entire stream, whereas DSCA for CDIS needs more than 50 data chunks to converge and operate at the highest level.

V. CONCLUSIONS

The main purpose of this work was to introduce the original data stream taxonomy in the context of *prior class probability* as well as to propose a novel *prior probability* estimation method dedicated to *dynamically imbalanced streams* (DIS). This goal was achieved by introducing the *Dynamic Statistical Concept Analysis* (DSCA) model, which predicts the *prior probability* using *Multi-layer Perception regressors* to detect relationships between statistical characteristics of a data chunk and the number of class instances. Research conducted on the total of 100 data streams divided into 10 types in terms of *prior class probability* and class imbalance characteristics confirmed the usefulness of the proposed method, especially in the case of the *discrete dynamically imbalanced streams* (DDIS). Statistical analysis further confirmed the obtained results.

In connection with the performed statistical analysis, we can adopt (with the probability resulting from the statistical test parameters) the experimental hypotheses 1, 2 and 3, which results directly from the conducted observations. Experimental hypothesis 4 can be adopted for synthetic data with the presented DDIS characteristics.

Certain modifications to the DSCA model can be introduced to improve results at the start of stream processing, before the model is properly trained. Depending on the type of analyzed stream, up to the DSCA convergence point, the proportions of the classes from the previous data chunk can be taken as a prediction of *prior probability* — in case of CDIS, or the prediction of another method of determining *prior probability*, analyzing the previously processed chunks — in case of DDIS.

Further works will focus mainly on employing *prior probability* estimation methods in the dynamically imbalanced data stream classification task as well as on the *prior probability* estimation in multiclass imbalanced data streams.

ACKNOWLEDGMENT

This work was supported by the *Polish National Science Centre* under the grant No. 2017/27/B/ST6/01325.

REFERENCES

[1] W. Gibson, "Neuromancer," *Ace*, 1984.

[2] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," *IDC iView: IDC Analyze the future*, vol. 2007, no. 2012, pp. 1–16, 2012.

[3] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, "Ensemble learning for data stream analysis: A survey," *Information Fusion*, vol. 37, pp. 132 – 156, 2017.

[4] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2000, pp. 71–80.
[5] L. I. Kuncheva, "Classifier ensembles for changing environments," in *International Workshop on Multiple Classifier Systems*. Springer, 2004, pp. 1–15.
[6] J. Gao, B. Ding, W. Fan, J. Han, and S. Y. Philip, "Classifying data streams with skewed class distributions and concept drifts," *IEEE Internet Computing*, vol. 12, no. 6, pp. 37–49, 2008.
[7] H.-L. Nguyen, Y.-K. Woon, and W.-K. Ng, "A survey on data stream clustering and classification," *Knowledge and information systems*, vol. 45, no. 3, pp. 535–569, 2015.
[8] S. Wang, L. L. Minku, and X. Yao, "Resampling-based ensemble methods for online class imbalance learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 5, pp. 1356–1368, May 2015.
[9] N. Rout, D. Mishra, and M. K. Mallick, "Handling imbalanced data: A survey," in *International Proceedings on Advances in Soft Computing, Intelligent Systems and Applications*. Springer, 2018, pp. 431–443.
[10] P. Ksieniewicz, "The prior probability in the batch classification of imbalanced data streams," *Neurocomputing*, Nov. 2020.
[11] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
[12] E. Alpaydin, *Introduction to machine learning*. MIT press, 2020.
[13] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance," *Climate research*, vol. 30, no. 1, pp. 79–82, 2005.
[14] J. Gama, *Knowledge Discovery from Data Streams*, 1st ed. Chapman & Hall/CRC, 2010.
[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[16] P. Ksieniewicz and P. Zyblewski, "stream-learn–open-source python library for difficult data stream batch analysis," *arXiv preprint arXiv:2001.11077*, 2020.

[C4]
Paweł Ksieniewicz. “The prior probability in the batch classification of imbalanced data streams”. W: *Neurocomputing* 452 (wrz. 2021), s. 309–316. DOI: 10.1016/j.neucom.2019.11.126

Neurocomputing 452 (2021) 309–316

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

The prior probability in the batch classification of imbalanced data streams

Paweł Ksieniewicz [a]

[a] Department of Systems and Computer Networks, Wrocław University of Science and Technology, Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland

ARTICLE INFO

Article history:
Received 31 May 2019
Revised 27 October 2019
Accepted 26 November 2019
Available online 20 November 2020
Communicated by Chennai Guest Editor

Keywords:
Pattern recognition
Classification
Imbalanced data
Data streams
Concept drift

ABSTRACT

In the diversity of contemporary decision-making tasks, where the data is no longer static and changes over time, data stream processing has become an important issue in the field of pattern recognition. In addition, most of the real problems are not balanced, representing their classes in various improportions. Following paper proposes the *Prior Imbalance Compensation* method, modifying on-the-fly predictions made by the base classifier, aiming at mapping *prior* probability in the statistics of assigned classes. It is intended to be a less computationally complex competition for popular algorithms such as SMOTE, solving this problem by oversampling the training set. The proposed method has been tested using computer experiments on the example of a set of various data streams, leading to promising results, suggesting its usefulness in solving this type of problems.

1. Introduction and related works

Pattern recognition methods are a collection of fundamental tools for solving problems faced by artificial intelligence, employing the popular interest that has been growing in recent times. As a rule, we divide them into two groups [1]. The first of them – *supervised learning* – is responsible for the description of new, yet unknown cases, on the basis of a set of patterns already known by labels [2]. In the case of the second group – *unsupervised learning* – the whole analysis takes place on a set of objects without any prior description.

The most popular problem of supervised learning is the classification, which, unlike the regression estimating the continuous value, assigns new objects to the set of discrete classes. The algorithms for solving this task have been developing intensively since the beginning of this field and have already formed a large group of methods, ranging from simple solutions such as the *Naive Bayes Classifier* [3] or *k-Nearest Neighbors* [4], through *decision trees* [5] and *forests* [6], *Support Vector Machines*, up to *neural networks*, with particular emphasis on the most-recently-fashionable *deep convolutional networks* [7].

1.1. Imbalanced data

The assumption of the most of classification algorithms is the equal occurrence of each of the considered classes [8]. This becomes problematic in the case when *imbalance ratio*, called also *prior* probability, is disturbed and one of the problem classes occurs much frequently than the others [9]. Data of this type are called *imbalanced data*. Due to the fact that the dominant majority of real decision problems, i.e. medical diagnostics, SPAM or fraud detection, presents imbalanced data, where, what should be emphasized, a less numerous class is the key from the perspective of the problem. Therefore, it became necessary to develop appropriate methods for counteracting the tendency of classifiers to favor the majority class [10].

Most of the proposed solutions to the problem of imbalanced data may be assigned to one of three groups. The first, theoretically the simplest, are the mechanisms built directly in the process of classifier training, modifying their model to align the impact of all classes of the problem, for example by the use of appropriate loss function. The second and the most common approach is the appropriate preprocessing of training data to align the presence of problem class patterns in it. Above simple *random oversampling* and – *undersamplig* one should distinguish here the SMOTE [11] – algorithm generating synthetic samples, along with its numerous variants, and ADASYN [12], extending it to include the distribution of the majority class in the synthesis. The last, but not least, approach are hybrid methods [13], using group of diversified classifiers in the construction of the decision system [14] connected by the prior-sensitive decision rule [15].

https://doi.org/10.1016/j.neucom.2019.11.126
0925-2312/

1.2. Data streams and concept drift

Important in the context of real problems of classification is also the aspect of knowledge historicity. A classification that is completely correct at a given point in time may lose its validity in the future and eventually turn out to be wrong. Therefore, it is naïve to assume that once-trained model, used for a long time will induct an error – which once estimated – will not increase over time, and the classifier itself, will not be outdated.

In many problems, we do not deal with static data set, and in addition to attributes, objects are characterized by their location in time. Such cases are called data streams and we process them, in principle, in one of two ways. In the first of them – online processing – each incoming object is analyzed separately, one by one and in this mode it drives the updating of the classification model. However, it is a very computationally intensive approach, and so-called batch processing is used much more frequently. The principle of batch processing is that incoming patterns are accumulated in so-called chunks and processed not pattern by pattern, but group by group.

Among the solutions to the stream classification problem, the most popular are approaches that allow for partial model fitting, i.e. modifying the existing model with information extracted from upcoming data, like WINNOW [16] or VFDT [17], and the ensemble approach, especially popular in batch processing [18]. Employing the incremental learning methods requires implementation of forgetting mechanisms, either as built-in capabilities of algorithm [19] or as dataset weighting or windowing [20].

In such ensembles, successive members of the committee are built on the basis of subsequent chunks, making it possible to weigh the influence of the member decision on the final prediction according to their quality determined on the latest data, and to trim the committee in order to eliminate obsolete models [21].

The already mentioned aspect of knowledge historicity introduces an additional complication in the problem of data stream classification. Outdating of models with passing time is the result

Table 1
Average results of *Gaussian Naive Bayes* classifier depending on the processing parameters. Bolded are the **measurements significantly better** in the pair of the base method and modified by PIC.

MINORITY	BALANCED ACCURACY				F-SCORE			
CLASS	GRADUAL		SUDDEN		GRADUAL		SUDDEN	
PERCENTAGE	GNB	PIC	GNB	PIC	GNB	PIC	GNB	PIC
5%	0.627	**0.664**	0.694	**0.718**	**0.966**	0.959	**0.964**	0.961
10%	0.673	**0.707**	0.741	**0.759**	**0.943**	0.934	**0.944**	0.939
20%	0.722	**0.744**	0.779	**0.790**	**0.899**	0.890	**0.908**	0.903
50%	**0.766**	**0.766**	**0.805**	**0.804**	**0.764**	**0.758**	**0.802**	0.796

Table 2
Average results of *k-NN* classifier depending on the processing parameters. Bolded are the **measurements significantly better** in the pair of the base method and modified by AIC.

MINORITY	BALANCED ACCURACY				F-SCORE			
CLASS	GRADUAL		SUDDEN		GRADUAL		SUDDEN	
PERCENTAGE	k-NN	PIC	k-NN	PIC	k-NN	PIC	k-NN	PIC
5%	0.657	**0.763**	0.693	**0.787**	**0.979**	0.973	**0.981**	0.975
10%	0.756	**0.829**	0.787	**0.846**	**0.968**	0.963	**0.971**	0.966
20%	0.842	**0.874**	0.861	**0.886**	**0.952**	0.948	**0.956**	0.952
50%	**0.900**	**0.900**	**0.909**	0.908	**0.900**	0.898	**0.909**	0.905

Table 3
Average results of *Support Vector Classifier* depending on the processing parameters. Bolded are the **measurements significantly better** in the pair of the base method and modified by AIC.

MINORITY	BALANCED ACCURACY				F-SCORE			
CLASS	GRADUAL		SUDDEN		GRADUAL		SUDDEN	
PERCENTAGE	SVM	PIC	SVM	PIC	SVM	PIC	SVM	PIC
5%	0.624	**0.797**	0.670	**0.820**	**0.978**	0.977	**0.981**	0.979
10%	0.730	**0.844**	0.769	**0.862**	**0.968**	0.967	**0.972**	0.970
20%	0.827	**0.879**	0.851	**0.891**	**0.952**	0.950	**0.957**	0.954
50%	**0.900**	**0.899**	**0.908**	0.907	**0.900**	0.897	**0.908**	0.904

Table 4
Average results of *Random Forest Classifier* depending on the processing parameters. Bolded are the **measurements significantly better** in the pair of the base method and modified by AIC.

MINORITY	BALANCED ACCURACY				F-SCORE			
CLASS	GRADUAL		SSUDDEN		GRADUAL		SUDDEN	
PERCENTAGE	RFC	PIC	RFC	PIC	RFC	PIC	RFC	PIC
5%	0.643	**0.738**	0.696	**0.776**	**0.977**	0.971	**0.979**	0.974
10%	0.739	**0.805**	0.785	**0.834**	**0.965**	0.959	**0.968**	0.963
20%	0.828	**0.855**	0.856	**0.875**	**0.945**	0.940	**0.951**	0.946
50%	**0.884**	**0.883**	**0.899**	0.898	**0.883**	**0.881**	**0.898**	0.895

of the phenomenon called *concept drift* [22]. Among the concept drifts you may distinguish between *sudden drift*, where the change between class distributions occurs rapidly at precise point, as well as *incremental* or *gradual drift*, where the concepts of classes are changing smoothly [22,18]. Solutions to this problem are focused either on the drift detection, signaling the need to rebuild the model, or on the classifier ensemble. It is also important to mention the propositions how to react to detected drifts, like DWM [23], STAGGER [24], or GT2FC [25]. Appearances of *concept drift* in *data streams* have become a challenge for plethora of practical solutions, such as computer systems security [26,27], medical diagnosis [28] or fraud detection [29].

1.3. Contributions

The following work is intended to achieve the following goals:

- Proposal of a strategy for interpreting the support obtained on a batch of data by the probabilistic base classifier in a manner that takes into account the prior probability.

Table 5
Summary of the classification quality for each of the eight analyzed classifiers. Bolded are the **measurements significantly better** in the competition.

	GRADUAL DRIFT							
MINORITY CLASS	Base method				PIC			
PERCENTAGE	GNB	k-NN	SVM	RFC	GNB	k-NN	SVM	RFC
5%	0.627	0.657	0.624	0.643	0.664	0.763	**0.797**	0.738
10%	0.673	0.756	0.730	0.739	0.707	0.829	**0.844**	0.805
20%	0.722	0.842	0.827	0.828	0.744	0.874	**0.879**	0.855
50%	0.766	**0.900**	**0.900**	0.884	0.766	**0.900**	**0.899**	0.883
	SUDDEN DRIFT							
5%	0.694	0.693	0.670	0.696	0.718	0.787	**0.820**	0.776
10%	0.741	0.787	0.769	0.785	0.759	0.846	**0.862**	0.834
20%	0.779	0.861	0.851	0.856	0.790	0.886	**0.891**	0.875
50%	0.805	**0.909**	**0.908**	0.899	0.804	0.908	0.907	0.898

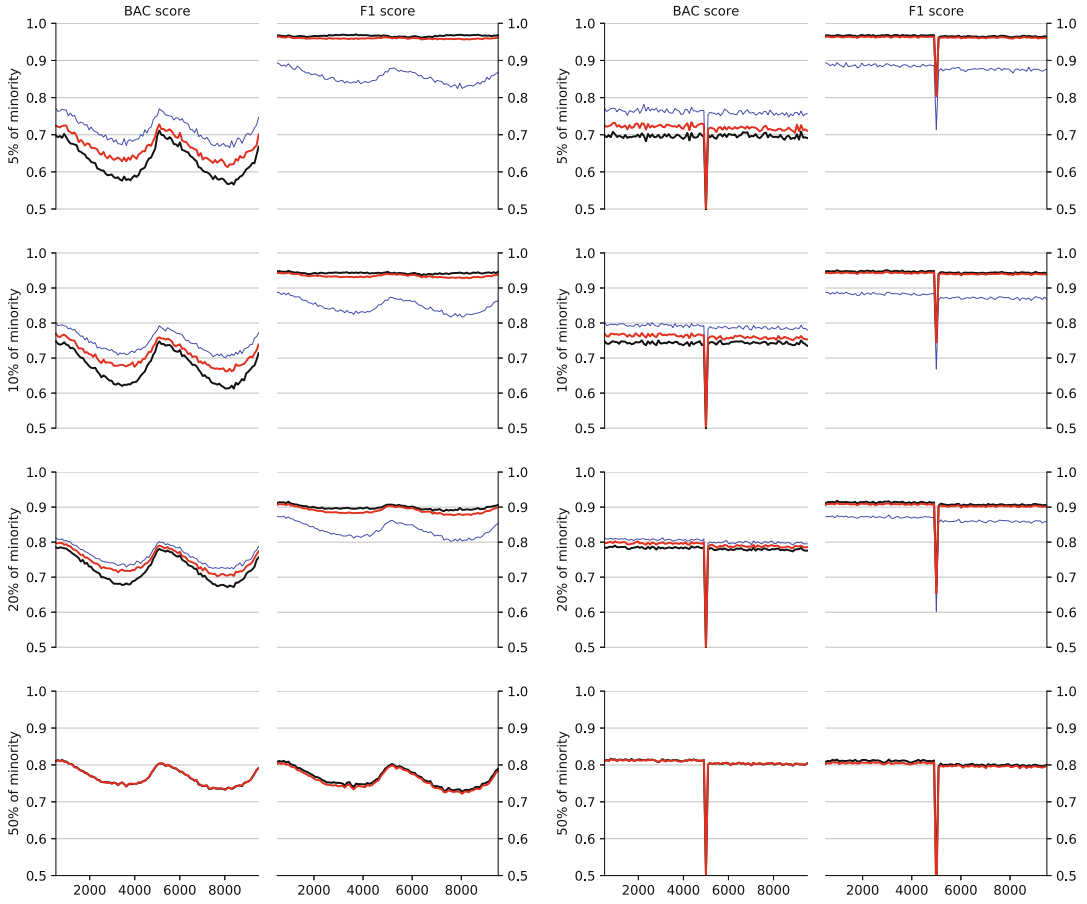


Fig. 1. Comparison of *balanced accuracy* and *F1 score* for bare *Gaussian Naive Bayes classifier* (black) and its proposed PIC version (red). *Gradual drift* on the left and *sudden drift* on the right side. Rows indicating different imbalance ratios (highly imbalanced stream on top, balanced data on bottom). For comparative purposes a blue line shows results with use of SMOTE oversampling in the same processing. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

- Experimental evaluation of the proposed approach on the example of a collection of benchmark data streams with various level of class imbalance, compared to the method without modification.

2. Employing prior probability of training set in batch classification

The method proposed in the following paper may be used together with any approach to batch processing of stream data, so for the clarity of the description and objectivity of experiments, it is presented with the simplest system of this type. We assume in it, in accordance with the principle of *test-and-train* evaluation, that for each incoming data chunk a single new model is built and it is used to predict the next batch of samples.

It is important to point out that the prerequisite for applying the described approach is to use the probabilistic model as a base classifier. This means that the answer of the trained algorithm to a new object is to be not only a direct assignment to one of the problem classes, but a *support vector* (*posterior probability* for the pattern) determining the chance of belonging to each class separately. Examples of such algorithms for data quantitative features may be *Gaussian Naive Bayes*, the *k-NN classifier* (with particular emphasis on its variant assigning distance-based weight to the set of nearest neighbors) or *Support Vector Machines* in its probabilistic interpretation.

2.1. Training

The first step of processing is to determine the local class distribution in the training set, which we carry out with each learning process, and thus in the considered case, with each subsequent chunk. The *prior* is the averaged value of this distribution on the current course of data stream. In contrast to the most of popular approaches, we do not modify the training set in any way (which is the case with all oversampling methods) but only extract simple statistical information from it.

2.2. Prediction

The second element of processing is the minimal modification of the prediction method based on the obtained supports. In the classic approach, the prediction is made independently for each pattern from the chunk, and the object is assigned to the class whose probability value was the largest. In the case of batch processing, however, we do not make prediction for a single sample, but for a set of patterns. This creates the potential to use information about the *prior* probability of the problem.

Sorting predicted objects, according to the support obtained for them, allows for assigning them to classes, using *posterior* probability, but keeping the class proportions consistent with *prior* probability. However, the proposed modification of predictions takes place only in cases where the *minority class* is underrepresented. In the case where the percentage of predictions in its favor is greater than according to stored *prior* probability, there is no artificial reduction. This is a delicate bias of the model in a favor of a more valuable class from the perspective of most classification problems.

2.3. The prior imbalance compensation

The proposed method, for the needs of the rest of this work, called *Prior Imbalance Compensation* (PIC), has a low computational overhead and was designed for data streams in which new objects appear with high frequency and in large numbers, or in cases where the imbalance ratio is very high (percentage of a minority class with less than 5% of objects), where in the absence of a sufficient number of minority examples it is not possible to generate synthetic patterns through the SMOTE algorithm or similar synthesizer. Overview of the proposed method is presented in Algorithm 1.

Algorithm 1: Prior Imbalance Compensation

Require:
data stream,
n – data chunk size,
training_procedure() – classifier training procedure,
classifier() – classification model,
$F_0()$ – support function of minority class used by *classifier*(),
$F_1()$ – support function of majority class used by *classifier*()
PREPARING INITIAL MODEL
1: *chunk* ← initial chunk in data stream
2: *a_priori* ← percentage of minority class samples in *chunk* 3: *classifier* ← *training_procedure*(*chunk*)
STREAM PROCESSING LOOP
4: **for***chunk* **in** data stream**do**
5: *prediction* ← $F_1() > F_0()$
PRIOR IMBALANCE COMPENSATION
6: *support_order* ← *argsort*($F_0()$) 7: *minority_treshold* ← *a_priori* ∗ *n*
8: *prediction*[*support_order* < *minority_treshold*] ← 0
ESTABLISHING A NEW MODEL
9: *local_a_priori* ← percentage of minority class samples in *chunk*
10: *a_priori* ← mean of all previous *local_a_priories*
11: *classifier* ← *training_procedure*(*chunk*)
12: **end for**

3. Experiments set-up

A key issue in planning experiments on imbalanced data is the selection of appropriate evaluation measures. The use of classical accuracy in this case will give a completely non-quantifiable result, because for example, in the case of a blind classifier always returning the prediction towards the majority class, it will respond with its percentage in the test set, which will give – with imbalance ratio 1:20 – a very good, though hypocritical result of 95%.

From the standard metrics for imbalanced problems, two measures were selected for the experiment:

- F1-score, returning a ratio between the doubled sum and the product of precision and recall.
- Balanced accuracy score, which is the average recall for each class of problem.

An improvement in *balanced* accuracy in the absence of *F1-score* degeneration will be recognized as a promising result.

In the experiments, a standard *Test-and-train* approach to evaluation of data streams was used. It consists of, after training the initial model on the first chunk, on subsequent testing on the upcoming data chunk and transferring it to the next learning loop until the end of the data stream.

Four algorithms were selected as the base classifiers for processing:

- GNB – *Gaussian Naive Bayes*,
- **k-**NN – *k-Nearest Neighbors* with $k = 5$ in version with weights of neighbors according to their Minkowski distance from sampled point in decision space,
- SVM – *Support Vector Machines* in its probabilistic interpretation,
- RF – *Random Forest* with 20 estimators and *Gini* criterion.

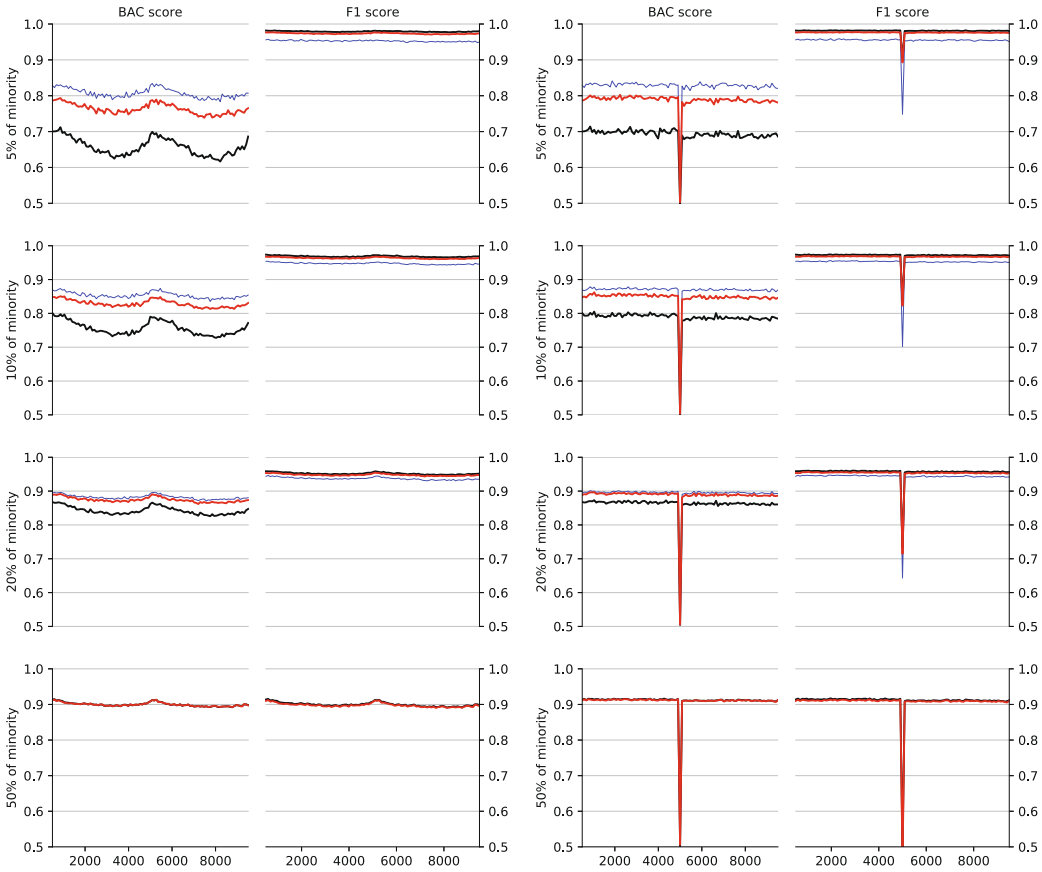


Fig. 2. Comparison of *balanced accuracy* and *F1 score* for bare *k-NN classifier* (black) and its proposed PIC version (red). *Gradual drift* on the left and *sudden drift* on the right side. Rows indicating different imbalance ratios (highly imbalanced stream on top, balanced data on bottom). For comparative purposes a blue line shows results with use of SMOTE oversampling in the same processing. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

To ensure the most reliable estimation of the solutions quality, one hundred data streams consisting of 50,000 objects broken into 100 chunks of 500 objects (5 million in total) were generated, each containing two concepts established according to the rules of generating synthetic classification problems used to create the Madelon set [30]. Concept changes are made in two ways, using *sudden drift* and *incremental drift*. For each of these combinations, four variants of streams were prepared, with different *imbalance ratios* (successively 5, 10, 20 and 50% share of the minority class).

In total, it gives 800 data streams used in the evaluation, and the presented results, for the stabilization of the sampling and measurability, represent the average of each combination of the imbalanced ratio and the type of drift from the hundred runs.

The statistical dependence test results were based on the *Wilcoxon test*, and the whole implementation of the experiments was carried out in *Python 3*, based on the *scikit-learn* [31] library. The whole code necessary to repeat the experiments contained in this work is in the public *git* repository[1].

4. Experimental evaluation

4.1. Goals

The main goal of conducted experimental evaluation was to verify the impact of *Prior Imbalance Correction* on discriminative power of four probabilistic classifiers on collection data streams with various types of concept drifts and various scales of data imbalance.

4.2. Results

Illustrations 1–4 illustrate the *balanced accuracy* and the *F1 score* for all previously described combinations of data streams for the selected four base classifiers. They are supplemented, for comparative purposes, with quality readings marked with a blue line for use of oversampling in the same processing with the SMOTE algorithm. All the charts, due to the considered binary problem, were scaled from the level of the *random classifier* (50%) to the end of the scale (100%).

Clean concepts in *gradual drifts* locate near the beginning of the graph, its end and the exact center, and the greatest mixing during the *concept drift* takes place at points 3000 and 8000. The *sudden drifts* occur exactly at 5000 points.

Analogously, Tables 1–4 present a summary of the results averaged for each stream type, both for the base classifiers and their versions modified by the PIC. Table 5 provides a summary of the classification quality for each of the eight analyzed classifiers (four base classifiers with and without PIC modification) in each of the analyzed groups of data streams, along with an analysis of the statistical dependence of the achieved results.

4.3. Observations

Analyzing the results achieved for GNB as the base classifier (Table and Fig. 1), it may be observed that for the imbalanced

[1] https://github.com/w4k2/apriori-stream

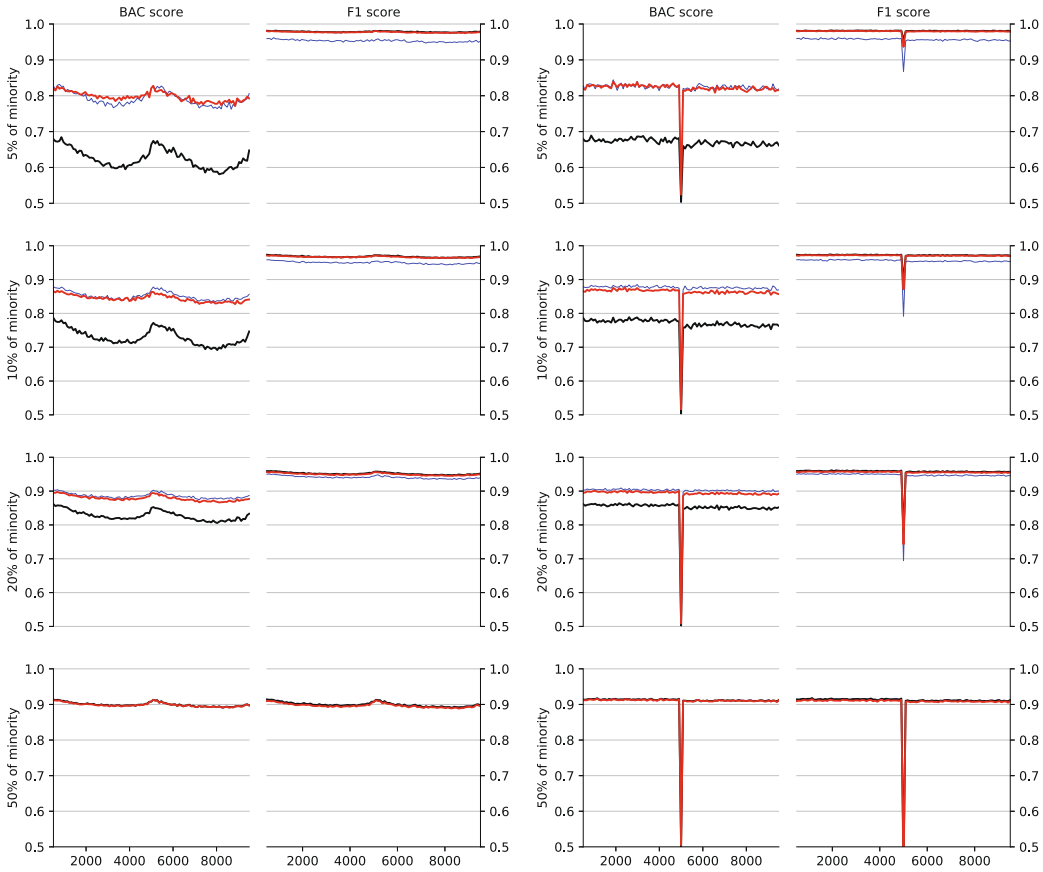


Fig. 3. Comparison of *balanced accuracy* and *F1 score* for bare *Support Vector Classifier* (black) and its proposed PIC version (red). *Gradual drift* on the left and *sudden drift* on the right side. Rows indicating different imbalance ratios (highly imbalanced stream on top, balanced data on bottom). For comparative purposes a blue line shows results with use of SMOTE oversampling in the same processing. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

streams (regardless of the *imbalance ratio*), both gradual and sudden, the PIC method is characterized by a few percent improvement in the *balanced accuracy*, especially noticeable (and growing to about ten percent) in the moments of the strongest mixing of patterns during gradual drift. The *F1-score* measure shows only a slight degradation at the level of a few promilles.

It is interesting to compare the quality achieved by PIC (red) and SMOTE (blue line). Oversampling allows for even stronger increase of *balanced accuracy*, but it strongly degenerates *F1-score*.

A similar situation occurs for the use of the *k*-NN algorithm (Table and Fig. 2), where the improvement of PIC against the base method is even more noticeable for *balanced accuracy*, with equally low *F1-score* degeneration. The relation between PIC and SMOTE remains unchanged.

Of particular interest is the behavior of algorithms interpreted with PIC using the SVM algorithm (Table and Fig. 3), which, according to the observations contained in Table 5, achieves the statistically significant highest results from all competitors in the case of imbalanced streams, regardless of the scale of unbalancing. Modification of the PIC in this case gives results very similar to SMOTE, without the need for oversampling. In the case of a very strong imbalance ratio (5%), the results achieved, both according to the *balanced accuracy* and *F1-score* metrics, PIC turns out to be better than SMOTE.

A reasonable explanation for the above observation may be the principle of SMOTE algorithm operation, which does not seem to be an ideal solution in the case of *linear classifiers*, or the ones using *radial basis function* kernel. Employing it in processing moves the decision boundary of the constructed model in the direction taking into account the class imbalance, but by the relatively high impact of synthetic patterns, it always distorts its shape with a greater or lesser negative effect on the quality of classification. Approach, as in PIC, just scaling the prediction threshold at a given level, also makes the desired shift of the decision boundary, but does not affect its shape in any manner, which gives a solution more fitted to real training data and closer to the optimal one. Employment of RFC as a base algorithm (Table and Fig. 4), in the relation of processing methods, gives results similar to those achieved by SVM.

5. Conclusions

This work proposed the *Prior Imbalance Compensation* (PIC) method for use in batch processing of imbalanced data streams. It consists in determining the missing predictions of a minority class in the case of under-representation in the prediction of a chunk.

The algorithm achieves promising results, showing not only the advantage over the used base methods, but also the over more computational-costly method of SMOTE, without the need to generate synthetic samples that carry the risk of introducing incorrect patterns into the training set.

A possible disadvantage of the proposed method is the naive assumption of *prior* constancy in the entire flow of the data stream. Due to the promising results of the presented research, its variant resistant to variable *imbalance ratio* and the version adapted to online classification will be implemented in the near future.

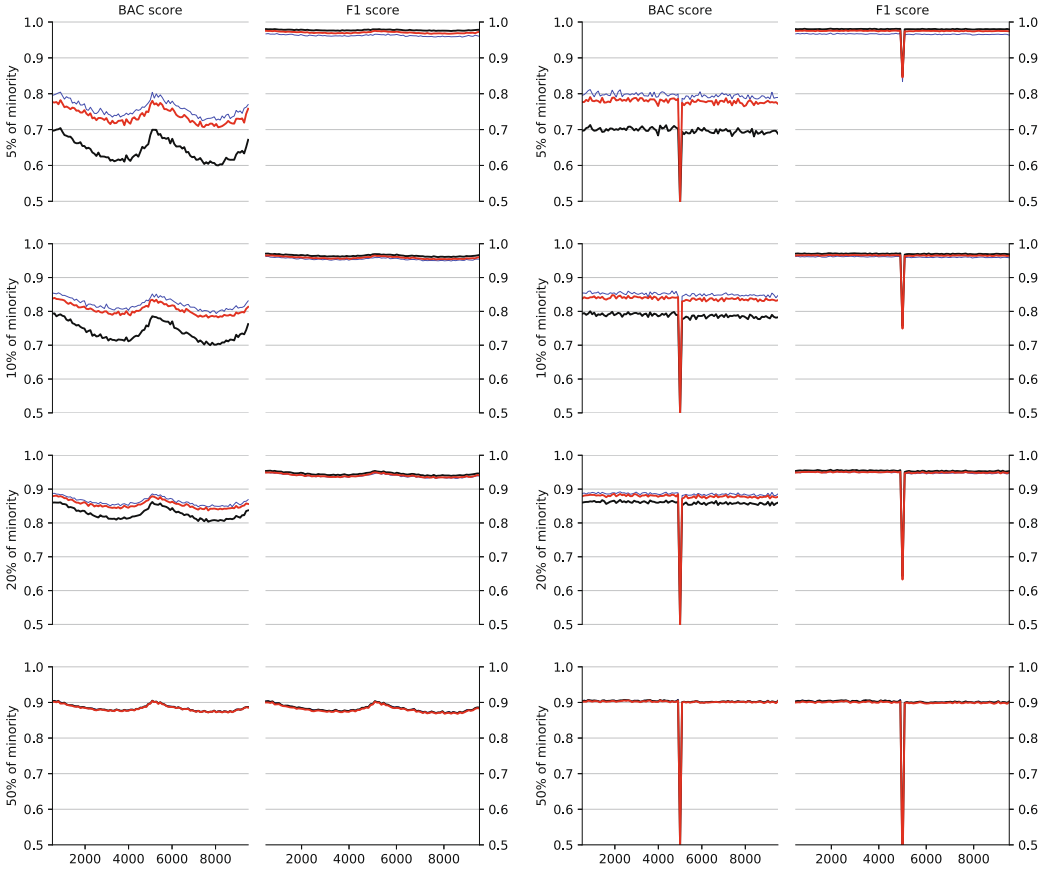


Fig. 4. Comparison of *balanced accuracy* and *F1 score* for bare *Random Forest Classifier* (black) and its proposed PIC version (red). *Gradual drift* on the left and *sudden drift* on the right side. Rows indicating different imbalance ratios (highly imbalanced stream on top, balanced data on bottom). For comparative purposes a blue line shows results with use of SMOTE oversampling in the same processing. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Declaration of Competing Interest

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

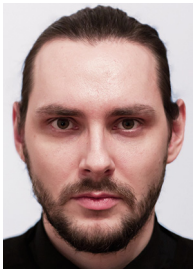
Acknowledgement

This work is supported by the Polish National Science Center under the Grant No. 2017/27/B/ST6/01325 as well the statutory funds of the Department of Systems and Computer Networks, Faculty of Electronics, Wrocław University of Science and Technology.

References

[1] T. Mitchell, B. Buchanan, G. DeJong, T. Dietterich, P. Rosenbloom, A. Waibel, Machine learning, Annu. Rev. Comput. Sci. 4 (1990) 417–433.
[2] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., Wiley, New York, 2001.
[3] P. Cheeseman, M. Self, J. Kelly, J. Stutz, W. Taylor, D. Freeman, AutoClass: a Bayesian classification system, in: Machine Learning: Proceedings of the Fifth International Workshop, Morgan Kaufmann, 1988..
[4] D. Wettschereck, T.G. Dietterich, An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms, Mach. Learn. 19 (1995) 5–27.
[5] D. Jankowski, K. Jackowski, B. Cyganek, Learning decision trees from data streams with concept drift, Procedia Comput. Sci. 80 (2016) 1682–1691.
[6] T.K. Ho, The random subspace method for constructing decision forests, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 832–844.
[7] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436.
[8] Y. Sun, A.K.C. Wong, M.S. Kamel, Classification of imbalanced data: a review, Int. J. Pattern Recogn. Artif. Intell. 23 (2009) 687–719.
[9] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (2009) 1263–1284.
[10] N.V. Chawla, N. Japkowicz, A. Kotcz, Special issue on learning from imbalanced data sets, ACM Sigkdd Explor. Newslett. 6 (2004) 1–6.
[11] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, Smote: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.
[12] H. He, Y. Bai, E.A. Garcia, S. Li, Adasyn: adaptive synthetic sampling approach for imbalanced learning, in: Proceedings of the International Joint Conference on Neural Networks, pp. 1322–1328..
[13] S. Wang, X. Yao, Diversity analysis on imbalanced data sets by using ensemble models, in: 2009 IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2009 – Proceedings, pp. 324–331..
[14] P. Ksieniewicz, Combining random subspace approach with smote oversampling for imbalanced data classification, in: Hybrid Artificial Intelligent Systems – 14th International Conference, HAIS 2019, Leon, Spain, September 4–6, 2019, Proceedings..
[15] P. Ksieniewicz, M. Woźniak, Imbalanced data classification based on feature selection techniques, in: International Conference on Intelligent Data Engineering and Automated Learning, Springer, pp. 296–303..
[16] N. Littlestone, Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm, Mach. Learn. 2 (1988) 285–318.
[17] P. Domingos, G. Hulten, Mining high-speed data streams, in: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '00, ACM, New York, NY, USA, 2000, pp. 71–80..
[18] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM Computing Surveys in Press, 2013..
[19] P. Ksieniewicz, M. Woźniak, B. Cyganek, A. Kasprzak, K. Walkowiak, Data stream classification using active learned neural networks, Neurocomputing (2019).
[20] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM Comput. Surv. 46 (2014) 44:1–44:37..
[21] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, 2004.
[22] G. Widmer, M. Kubat, Learning in the presence of concept drift and hidden contexts, Mach. Learn. 23 (1996) 69–101.
[23] J. Kolter, M. Maloof, Dynamic weighted majority: a new ensemble method for tracking concept drift, in: Data Mining, 2003. ICDM 2003. Third IEEE International Conference on, pp. 123–130..

[24] J.C. Schlimmer, R.H. Granger Jr., Incremental learning from noisy data, Mach. Learn. 1 (1986) 317–354.
[25] A. Bouchachia, C. Vanaret, GT2FC: an online growing interval type-2 self-learning fuzzy classifier, IEEE Trans. Fuzzy Syst. 22 (2014) 999–1018.
[26] T. Lane, C.E. Brodley, Approaches to online learning and concept drift for user identification in computer security, in: R. Agrawal, P.E. Stolorz, G. Piatetsky-Shapiro (Eds.), Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98), New York City, New York, USA, August 27–31, 1998, AAAI Press, 1998, pp. 259–263..
[27] J.R. Méndez, F. Fdez-Riverola, E.L. Iglesias, F. Díaz, J.M. Corchado, Tracking Concept Drift at Feature Selection Stage in SpamHunting: An Anti-spam Instance-Based Reasoning System, Springer, Berlin Heidelberg, Berlin, Heidelberg, pp. 504–518..
[28] A.A. Beyene, T. Welemariam, M. Persson, N. Lavesson, Improved concept drift handling in surgery prediction and other applications, Knowl. Inf. Syst. 44 (2015) 177–196.
[29] A.D. Pozzolo, G. Boracchi, O. Caelen, C. Alippi, G. Bontempi, Credit card fraud detection and concept-drift adaptation with delayed supervised information, in: IJCNN, IEEE, 2015, pp. 1–8.
[30] I. Guyon, Design of experiments of the nips 2003 variable selection benchmark, in: NIPS 2003 Workshop on Feature Extraction and Feature Selection..
[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.



Paweł Ksieniewicz is an assistant professor of computer science at the Department of Systems and Computer Networks, Wroclaw University of Science and Technology, Poland. He received M.Sc. and Ph.D. degrees in computer science from the Wroclaw University of Science and Technology in 2013 and 2017, respectively. His research focuses on pattern classification and machine learning methods and computer vision.

[C5]
Paweł Ksieniewicz i in. “Fake News Detection from Data Streams”. W: *2020 International Joint Conference on Neural Networks (IJCNN)*. 2020, s. 1–8. DOI: 10.1109/IJCNN48605.2020.9207498

Fake News Detection from Data Streams

1st Paweł Ksieniewicz
Department of Systems and Computer Networks
Wroclaw Univ.of Science and Technology
Wrocław, Poland
e-mail: pawel.ksieniewicz@pwr.edu.pl
ORCID: 0000-0001-9578-8395

2nd Paweł Zyblewski
Department of Systems and Computer Networks
Wroclaw Univ.of Science and Technology
Wrocław, Poland
e-mail: pawel.zyblewski@pwr.edu.pl
ORCID: 0000-0002-4224-6709

3rd Michał Choraś
Institute of Telecommunications and Computer Science
UTP Univ.of Science and Technology
Bydgoszcz, Poland
e-mail: chorasm@utp.edu.pl

4th Rafał Kozik
Institute of Telecommunications and Computer Science
UTP Univ.of Science and Technology
Bydgoszcz, Poland
e-mail: kozikr@utp.edu.pl
ORCID: 0000-0001-7122-3306

5th Agata Giełczyk
Institute of Telecommunications and Computer Science
UTP Univ.of Science and Technology
Bydgoszcz, Poland
e-mail: agata.gielczyk@utp.edu.pl
ORCID: 0000-0002-5630-7461

6th Michał Woźniak
Department of Systems and Computer Networks
Wroclaw Univ.of Science and Technology
Wrocław, Poland
e-mail: michal.wozniak@pwr.edu.pl
ORCID: 0000-0003-0146-4205

***Abstract*—Using fake news as a political or economic tool is not new, but the scale of their use is currently alarming, especially on social media. The authors of misinformation try to influence the users' decisions, both in the economic and political sphere. The facts of using disinformation during elections are well known. Currently, two fake news detection approaches dominate. The first approach, so-called fact or news checker, is based on the knowledge and work of volunteers, the second approach employs artificial intelligence algorithms for news analysis and manipulation detection. In this work, we will focus on using machine learning methods to detect fake news. However, unlike most approaches, we will treat incoming messages as stream data, taking into account the possibility of concept drift occurring, i.e., appearing changes in the probabilistic characteristics of the classification model during the exploitation of the classifier. The developed methods have been evaluated based on computer experiments on benchmark data, and the obtained results prove their usefulness for the problem under consideration. The proposed solutions are part of the distributed platform developed by the H2020 SocialTruth project consortium.**

***Index Terms*—stream analysis, fake news, distributed architecture**

I. Introduction

The idea of using fake news to achieve political or economic goals is not new and was used, among others, in ancient Rome, where Octavian spread false information about Mark Antony, who was allegedly a drunkard, a womanizer and a toy in the hands of Queen Cleoparty[1]. We may also read how the disinformation could be used in the Holy Bible, e.g., in the resurrection description[2].

Unfortunately, for several years the problem of using fake news has become more and more nagging and recently concerned the presidential campaigns in the USA and France. Fake news could also be used as a political weapon, which may promote desirable ideas or behaviours on the ground. The Russian Federation widely uses it as part of the information war, which has been followed by an EU response, resulting in the creation of the EUvsDisinfo website[3] to monitor and counteract disinformation campaigns.

Another example is the systematic activity of the Lithuanian government, which allows temporary shutting down servers for 48 hours without a court order if they are used to spread fake news[4]. These activities are supported by civic movements, the so-called net of elves, which manually check the news published on the Internet and report the detected violations to the authorities.

Currently, more and more websites are being created that are trying to help assess the accuracy of the information, the so-called fact-checkers. Unfortunately, it is not possible to check them manually with such a massive amount of new messages. Hence, more and more hope is placed in automatic fake news

[1]Kaminska, Izabella, A lesson in fake news from the info-wars of ancient Rome, Financial Times, January 17, 2017, https://www.ft.com/content/aaf2bb08-dca2-11e6-86ac-f253db7791c6?mhq5j=e3

[2]"So the soldiers took the money and did as they were instructed. And this story has been widely circulated among the Jews to this very day." (Matthew 28:15)

[3]https://euvsdisinfo.eu/

[4]Michael Peel, Fake news: How Lithuania's 'elves' take on Russian trolls, Financial Times, Feb. 4, 2019, https://www.ft.com/content/b3701b12-2544-11e9-b329-c7e6ceb5ffdf

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

detection systems using machine learning methods. The main reason for the rapid increase in the use of disinformation is the ability to use not only traditional mainstream media but also social media, like Twitter or Facebook. It is worth noting that its popularity can rapidly grow according to the rule that *false news spreads faster and more comprehensive*. Its extensive spread has a severe negative impact on media users and society.

The main goal of publishing such information with malicious content is to attract readers, which could increase publisher rank and popularity, which consequently increases revenues form adds. It is worth mentioning that there is no single definition of what fake news is and what it is not. Shu et al. proposed the following taxonomy [1]:

- satire news with proper context and hoaxes,
- rumours,
- conspiracy theories,
- misinformation that is created unintentionally.

Interesting are studies analyzing which types of attributes prove to be useful in the context of fake news classification [1]. Among the standard solutions present in the research, we may distinguish the analysis of creators and readers of texts, document content, stylometric analysis and verification of positioning the document in social networks [2]. Image analysis that focuses on false video information is also a promising approach [3]. Equally interesting seems to be the work of Conroy at al. [4] proposing a distinction between approaches to linguistic (semantic, rhetorical, discourse and simple probabilistic recognition models) and social (analysis of the author's behaviour within the social network or analysis of the built context by all of his posts). Castillo et al. [5] bases the construction of recognition models on users' behaviour in the context of posting and forwarding content depending not only on their content but also references to other documents. Ferrara et al. [6] analyze various data representations, and Afroz et al. [7] different variations of stylometric metrics.

Sharma et al. [8] discussed several topics related to the problem in question and pointed to the possibility of using the SCAN (Scientific Content Analysis) method to solve it. Jin et al. [9] proposed the compelling approach for the task of automatic news verification, departing from text analysis in favour of image data. Zhang et al. [2] pointed to the dynamic nature of social media messages and proposed analyzing them in the context of streaming data. Horne and Adali [10] employed SVM to distinguish between fake, authentic and satire messages. This kind of classifier has also been used by Cheng et al. for users classification, using semantic analysis and behavioural feature descriptors to detect potentially fake online posts [11].

Interesting comparative studies by Gravanis et al. [12] evaluated several linguistic features based classification approaches. They presented the results showing that known classifier models, especially ensembles, may be successfully used as fake news detectors. Bondielli et al. [13] pointed out that while anomaly detection and clustering methods could be used for fake news detection, this problem is usually reduced to the classification task. Atodiresi et al. [14] considered an NLP tools-based approach to tweet analysis. The authors defined this problem as a regression task and thus were able to assign a credibility value to each message.

Unfortunately, in most works, the authors consider the problem of fake news detection is a classic problem of data analysis, without taking into account their streaming nature. What is more, it should be taken into account that the profile of messages classified as fake news may change over time, i.e., we are dealing with a phenomenon known as concept drift. This is since, as with other information security problems, such as the detection of unwanted mail, the authors of fake news are aware that publishing them is becoming more difficult because automatic detection systems will detect them. Thus, it can be expected that their profile will change over time to deceive these systems, and therefore requires authors of these types of systems to equip them with mechanisms for adapting to changes in probabilistic characteristics of the fake news detection task. This work will attempt to develop fake news detection algorithms based on a data stream where we will not assume its stationary nature [15]. To the best of the authors' knowledge, there is no work treating fake news detection as a problem of streaming data classification. Although some authors note that social media data are of such nature, only Wang and Terano [16] use techniques adequate to analyze data streams. However, their approach is limited to relatively short streams and does not potentially take into account the non-stationary nature of the data.

The main contributions of this paper are as follows:

- Formulating the problem of fake news detection as a data stream classification task.
- Proposing a novel pattern classification methods based on feature extraction techniques, which address the detection of fake news in streaming data from social media.
- An extensive experimental analysis backed-up by the statistical tests.

II. SocialTruth Platform Architecture

The proposed machine learning solutions constitute text verification services, one of the critical elements of the *SocialTruth* platform. From a broader perspective, it is crucial to explain the environment where the proposed solutions will operate and how they will bring benefits for the end-users, which are all kind of actors that need to cope with fake news challenges.

Therefore, in this section, we give a general overview of the *SocialTruth* project platform, which has been depicted in Figure 1. The technology stack has been decomposed into the following logical elements that have been detailed in the next subsections:

- physical elements (nodes) and their orchestration,
- verification services,
- messaging and event processing.

The data is ingested into the platform either by the user (journalist, author, reader) over HTTP(S) protocol or using

dedicated crawlers (data connectors) that send data over the binary protocol to the *Apache Kafka* framework. The *Apache Kafka* is a distributed streaming platform implementing the publish-subscribe model. Once the ingested data is published to one of the *Kafka* topics, it can be simultaneously consumed by various verification services and stream processing applications. Once the services When the services finalize their computations, they make the results available on another *Kafka* topic, which can be consumed by other services again. Such kind of processing pipeline is called choreography pattern. When the services finalize their computations, they make the results available on another *Kafka* topic, which can be consumed by other services again. Such kind of processing pipeline is called choreography pattern. It is a contradiction to the orchestration pattern, which introduces a central entity (orchestrator) controlling the execution of each stage in the pipeline. These two approaches have their advantages. We use a mix of both when implementing web service handling the HTTP requests.

A. Physical nodes comprising the system

The first and the most bottom layer in the technology stack constitutes the orchestration framework. It is laid down on top of an infrastructure composed of virtual and hardware machines. This layer is intended to implement automated resource management, and thus it facilitates the entire platform with such capabilities as flexibility, scalability, and fault tolerance. It is the responsibility of the orchestration layer to effectively deploy the services on the available computational nodes (both physical and virtual). It is achieved thanks to containers that are sandboxes that contain the implemented service together with all the software dependencies (libraries and execution environment). In such a form, the services can be easily migrated between the computational nodes and deployed. In the proposed solution, we have used *Docker Swarm* system.

B. Verification services

All the fake news detection mechanism (presented in this paper) are the instances of text verification services. As such, they are the critical building blocks of the system and are deployed as a micro-services. Micro-service is an independently deployable component, which (in the proposed architecture) is packed as a *Docker* container. Each micro-service is focused on providing single functionality. Moreover, each functional service provides an API that allows other clients to interact with the service in synchronous (e.g. REST) and asynchronous ways (e.g., events). Each service can also have client API for interacting with other components/services (e.g., databases). Moreover, the service can subscribe (listen) to a notification sent from other components in the system. In the proposed architecture, we heavily use asynchronous event-based communication in favour of synchronous calls. This allows us to avoid tight coupling between the verification services and other components in the platform. In that regard, each verification service subscribes to a dedicated topic and produces results on another one. In the diagram of the architecture, we deliberately depicted image and meta-verification services. These constitute essential elements of the platform as news commonly is accompanied by images to support the content. However, this matter is out of the scope of this paper.

C. Messaging and event processing

As we mentioned before, in the described system, we have adapted *Apache Kafka*. It is a distributed streaming platform, which enables both real-time event processing and event-driven communication between various components. From the architectural point of view, *Apache Kafka* constitutes a flexible and efficient way to integrate all the components, both existing tools as well as the new ones developed during the project or by the community.

Moreover, on top of the *Apache Kafka* system, we deploy the *Complex Event Processing* (CEP) engine. We use it to pre-process the data before storing and presenting it to the end-user or the verification service. For example, we use this technology to join data streams produced by verification services belonging to the same type. In that regard, we can present the user analysis results obtained from different classifiers.

III. Methods

As we noticed in the introduction to the following work, the overwhelming majority of machine learning research in the field of fake news detection relies on the extraction of linguistic features. The main subject covered in the study presented in this work is, however, the analysis of approaches to feature extraction for the needs of the data stream classification task for the problem above. At the entry point, each of the patterns constructing the stream, representing individual articles, contains a thousand attributes determined by a simple solution of *Count Vectorizer* – applied to isolate the impact of diverse linguistic analysis methods on the quality of results achieved.

The dataset developed for the experiment is based on the *Getting Real about Fake news* set, containing 13,000 articles marked as fake news by *BS Detector Chrome Extension* users. In each of them, we have the title, content and timestamp. To develop the classification model, it was supplemented with the same number of articles from sources commonly considered to be verified and reliable, selected in a similar period indicated by timestamps of data from the original set. The dataset developed in this way was then ordered following the publication dates, to develop a data stream enabling to perform the reliable experiments.

A. Dimensionality reduction

A thousand attributes of a learning set constitute a problem of very high dimensionality. Many features may show mutual statistical dependence, and many of them may prove to be utterly unimportant in the context of the classification problem under consideration, showing no relationship with the actual problem labels. It turns out that it is necessary to reduce the scale of the problem or modify it using methods of

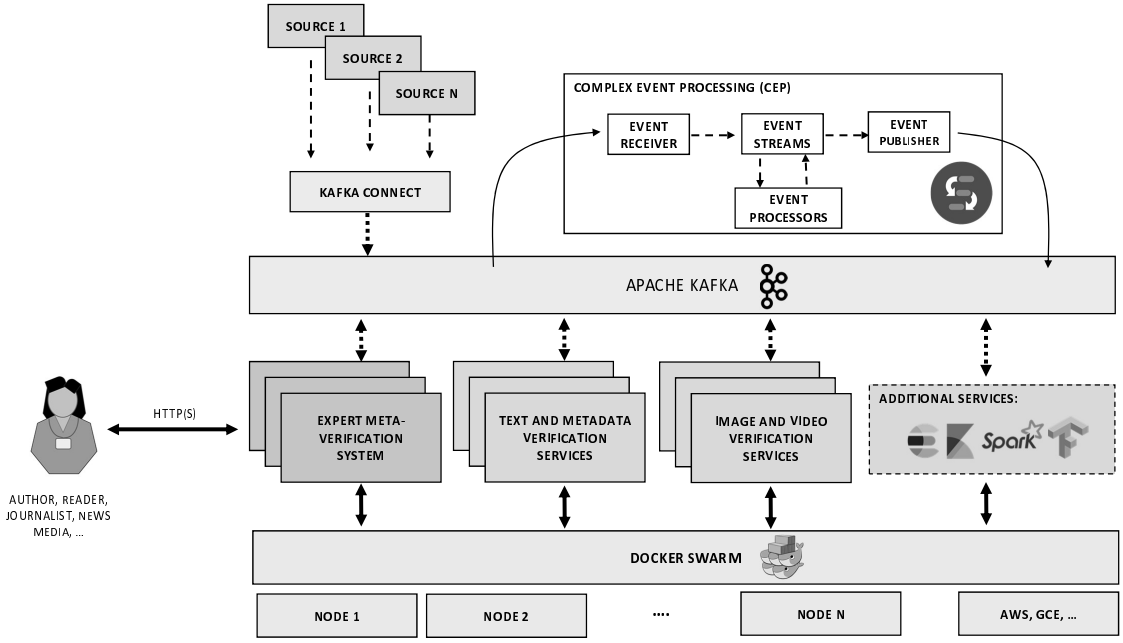


Fig. 1: SocialTruth Platform - general overview of the architecture and technology stack

feature extraction and selection. In the conducted experimental evaluation, three strategies of this type were applied.

a) PCA*:* The first considered strategy was the *Principal Components Analysis*. For the set of observations, the coordinate system is rotated in such a way as to maximize the variance of subsequent attributes. It leads to an increasing percentage of the explained variance. It allows the transformation of data to representations with a lower number of features than the input set by the combination of real attributes of the problem. It is a state-of-art solution to extract features for multidimensional data.

b) COUNT VECTORIZER*:* The second method used in experiments is based on the base approach of feature extraction – *Count Vectorizer*. To take into account the same impact of document titles and content, each of the subsets of features was normalized using the *standard normalization*, followed by the selection of the features represented by the most significant number within the data used to construct the extraction model.

c) FEATURE SELECTION*:* The last method used was a filter-based feature selection. Methods from this group use statistical techniques to assess the relationship between each feature and problem classes [17]. The scores obtained are then used to select the most significant features. In this case, as the correlation measure, the *Chi-Squared test* was used.

Each of the considered methods of reducing the feature space, due to the streaming nature of the processing, was implemented in the form of a model fitted based on the first portion of data supplied to the classification system (the first chunk). Subsequent portions of data were reduced under the model prepared in this manner, so they showed the same, processed characteristics. However they did not pose a problem of *data peeking*, due to the lack of influence on the feature extraction model.

B. Strategies to train the classifier on a data stream

Each of the three proposed dimensionality reduction strategies was analyzed in three different *state-of-art* methods for constructing classification models in data streams.

*a) Streaming Ensemble Algorithm (*SEA*):* Proposed by Street and Kim in [18], SEA constructs a classifier ensemble of a fixed size, by training a new base classifier on each observed data chunk. This approach is separate from the commonly used approaches with updated models [19]. In case of exceeding the fixed pool size, the worst performing model according to a given metric is removed. The final decision of the ensemble is produced according to the sum rule [20].

*b) Online bagging (*OB*):* Ensemble learning algorithm proposed by Oza in [21] and based on offline *Bagging*. It maintains a classifier pool in which, with the arrival of the new sample, each base estimator is trained on it K times, where K comes from the $Poisson(\lambda = 1)$ distribution.

*c) Single model (*SM*):* In addition to the classifier ensembles, the natural ability of selected classifiers to adapt to partial fitting was also tested, where a single model is constructed. However, each incoming data chunk is used to modify its properties with knowledge acquired based on a new class distribution. This approach, apart from the classifiers that effectively provide the forgetting mechanisms, is not

immune to the concept drift phenomenon. Nevertheless, unlike ensemble methods, it is not strongly dependent on the size of a single chunk of data used in processing [22].

C. Base classifiers for data stream processing

Each of these processing methods also requires the selection of a base classifier, which – due to the consideration of processing using single models – must meet the requirement to be able to conduct a partial fit of the already built recognition model. Three classification algorithms meeting this condition were selected.

- GNB *Gaussian Naive Bayes* – without prior probabilities,
- MLP *Multi-layer Perceptron* – with one hidden layer build on 100 neurons, using ReLU activation function and stochastic gradient-based optimizer.
- HT *Hoeffding Tree* – using *gini* split criterion and *Naive Bayes Adaptive* prediction mechanism.

IV. Experimental setup

The entire experimental evaluation was implemented using *Python* libraries, based on the *scikit-learn* [23] module in the implementation of two base classifiers and all feature reduction methods, on the *stream-learn* [24] module in data stream processing, calculating evaluation metrics and employed classifier ensembles of stream processing and on the *scikit-multiflow* [25] module in the implementation of *Hoeffding Tree*.

The implementation of the analyzed processing methods, supplemented with a module of datastream generation prepared following the description of static data included in Section II, together with the analytical script used to generate all tables and illustrations contained in the following section, is publicly available on the GIT repository (https://github.com/w4k2/fakestreams).

During the process of methods evaluation, the *state-of-art Test-Then-Train* methodology was used, which involves alternating testing of algorithms on an incoming portion of data, which has not yet been made available to the classifier for the needs of learning and updating its model after supplementing it with original labels. Two hundred fifty patterns were adopted as the size of a single chunk.

Each of the three feature extraction methods has been paired with each of the three stream processing strategies built on each of the three considered base classifiers, assuming 2, 10, 50, 100, 200, 500 or 1000 extracted attributes for the construction of the classification model, which resulted in 189 runs being the basis of the evaluation. Due to the balanced nature of the problem, the results are presented using the accuracy metric, being appropriate for this kind of data.

V. Experimental evaluation

The results of the conducted experiments were collected in Table I, divided by horizontal sections into individual stream processing strategies, with the numbers of extracted features and vertical sections for the used spatial reduction methods. Besides, in Figures 2, 3 and 4 respectively, the experiments' runs for each of the processing strategies were presented. The illustrations were reduced to runs only for 10, 100, and 1000 attributes to increase readability.

The results show that the FS method is the most effective when paired with SEA. At the same time, the MLP, in this case, is characterized by a gradual increase in the classification accuracy until 200 attributes, but over time reducing its performance. The behaviour of HT may be unusual, which despite initial growth, after exceeding 500 features, decreases in generalization capacity to the random classifier level. A similar observation is valid for PCA and CV. When the GNB classifier is used as the base for SEA, the method starts from a level slightly higher than the random classifier and achieves decent classification accuracy over time, but not at a level comparable to MLP.

The OB strategy seems to be far less suited for reduction by FS than SEA. The best results are achieved, again by the MLP classifier used as the base, using the PCA method. In this case, for a combination of PCA, OB and MLP with 1000 attributes generated, we achieve the highest average classification value of 81 per cent among the experiments carried out. An interesting difference between the SEA and OB approaches is that GNB rarely goes beyond the level of the random classifier for the latter and PCA reduction. On the other hand, in the case of CV and FS, GNB is characterized by a steady increase in classification ability, directly dependent on the number of problems features. HT properties do not differ from those developed by the SEA strategy.

The use of SM in the classification reveals the weakness of the PCA algorithm in tandem with GNB, further reducing the quality of such a solution. However, it can be seen that the same reduction with the MLP algorithm leads to the best results among all SM-base strategies. In this case, the CV extraction does not meet the expectations, in any of the tested instances leading to the best solution, which is evenly distributed between PCA and FS.

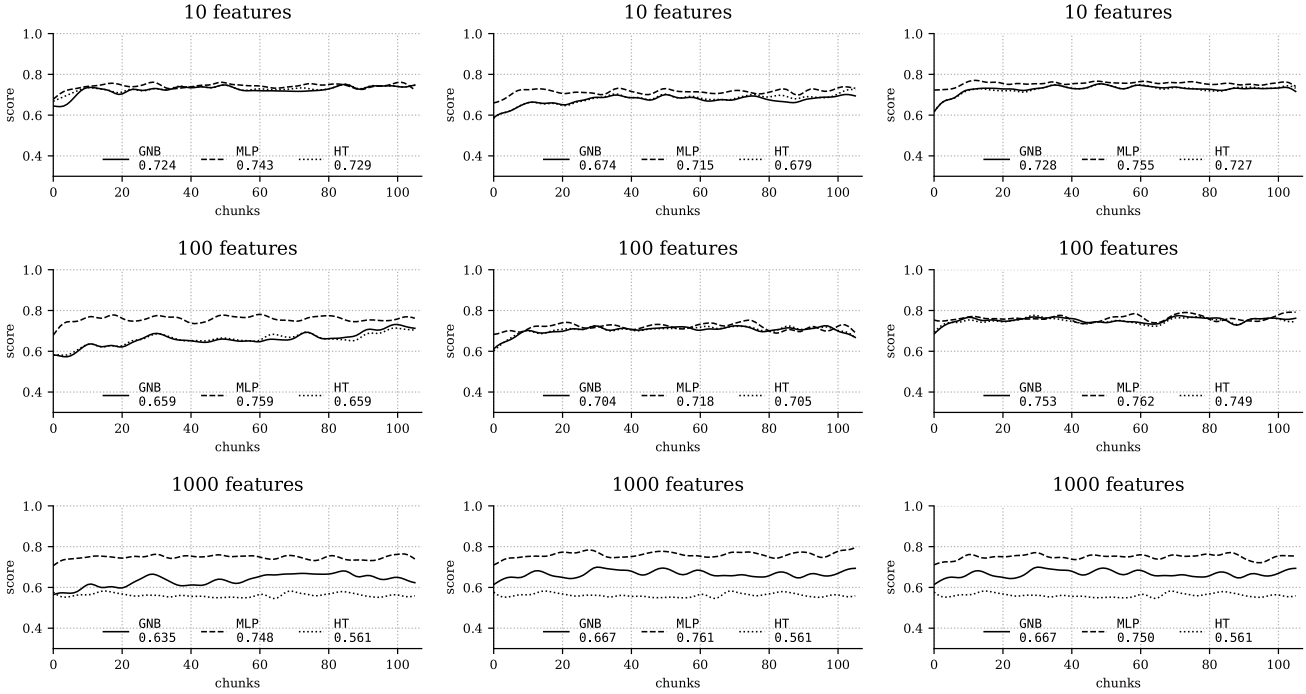
Summing up the analysis of the results obtained, it can be stated that the most effective of the considered classification algorithms is MLP. One may see a simple linear relationship between its generalization ability and the attributes number of the constructed model, almost regardless of the used extraction method and processing strategy. Obvious observation for all approaches is also quite the inverse relationship that occurs for HT, whose quality of classification degenerates with the increase in the dimensionality of the problem. GNB and PCA can be considered as the worst combination of streaming approaches, especially for OB and SM. MLP classifier with OB and PCA should be distinguished as the best combination of all analyzed strategies to deal with dimensionality reduction in fake news data stream processing.

VI. Conclusions

It is one of the first work which treats the problem of fake news detection as the stream data classification task and also takes into consideration that the characteristics described the incoming messages could change over time. Studies of this type have not been represented widely in the literature so far,

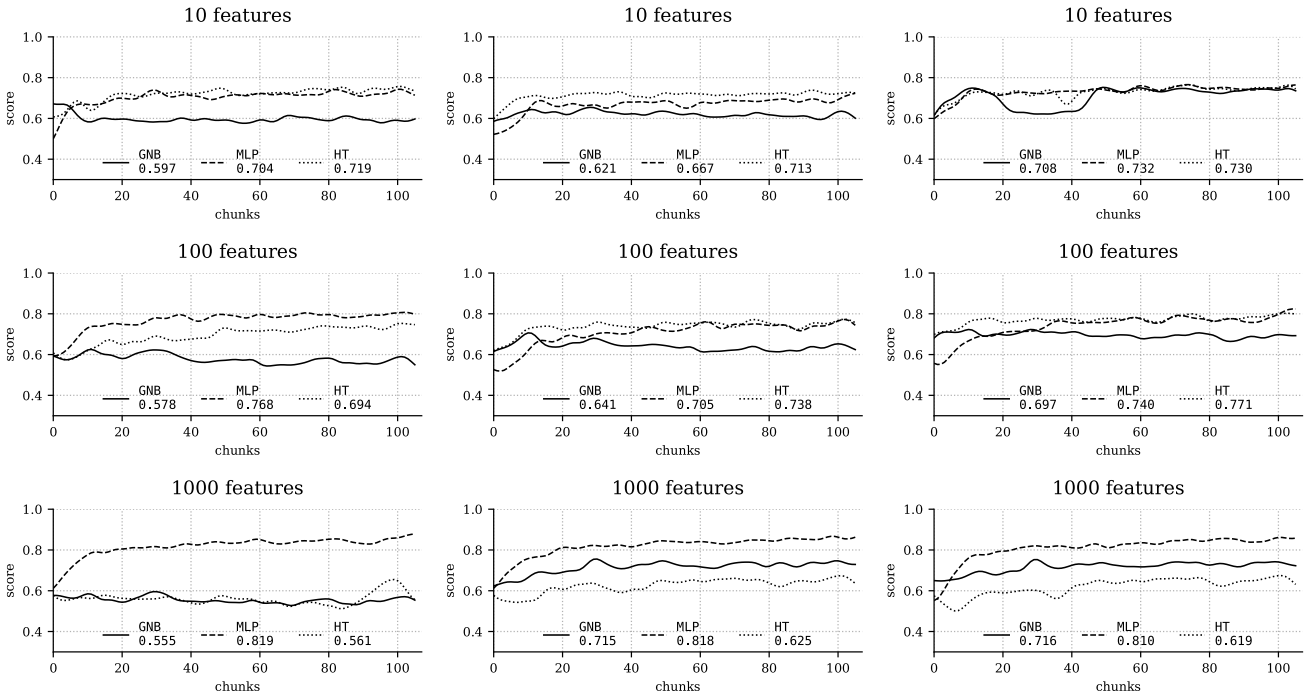
TABLE I: Results of experimental evaluation.

FEATURES	PCA			COUNT VECTORIZER			FEATURE SELECTION		
	GNB	MLP	HT	GNB	MLP	HT	GNB	MLP	HT
	SEA								
2	0.602	0.666	0.570	0.553	0.628	0.560	0.663	**0.679**	0.673
10	0.724	0.743	0.729	0.674	0.715	0.679	0.728	**0.755**	0.727
50	0.693	0.770	0.701	0.714	0.728	0.717	0.744	**0.778**	0.744
100	0.659	0.759	0.659	0.704	0.718	0.705	0.753	**0.762**	0.749
200	0.656	0.692	0.671	0.716	0.733	0.693	0.758	**0.764**	0.713
500	0.664	**0.747**	0.561	0.732	0.733	0.561	0.745	0.746	0.561
1000	0.635	0.748	0.561	0.667	**0.761**	0.561	0.667	0.750	0.561
	ONLINE BAGGING								
2	0.516	0.641	0.664	0.517	0.604	0.612	0.636	0.677	**0.688**
10	0.597	0.704	0.719	0.621	0.667	0.713	0.708	**0.732**	0.730
50	0.613	**0.768**	0.711	0.657	0.695	0.745	0.706	0.728	0.767
100	0.578	0.768	0.694	0.641	0.705	0.738	0.697	0.740	**0.771**
200	0.569	0.754	0.658	0.650	0.758	0.708	0.691	**0.764**	0.738
500	0.559	**0.804**	0.589	0.673	0.785	0.633	0.722	0.792	0.616
1000	0.555	**0.819**	0.561	0.715	0.818	0.625	0.716	0.810	0.619
	SINGLE MODEL								
2	0.512	0.633	0.654	0.518	0.595	0.611	0.634	0.673	**0.690**
10	0.595	0.700	0.702	0.622	0.662	0.707	0.710	**0.727**	0.694
50	0.618	0.768	0.672	0.644	0.697	0.730	0.707	0.727	**0.747**
100	0.579	**0.766**	0.663	0.645	0.705	0.722	0.696	0.742	0.745
200	0.562	0.752	0.667	0.651	0.753	0.714	0.686	**0.762**	0.724
500	0.556	**0.796**	0.618	0.672	0.777	0.661	0.719	0.788	0.630
1000	0.554	**0.808**	0.615	0.722	0.806	0.626	0.722	0.805	0.626



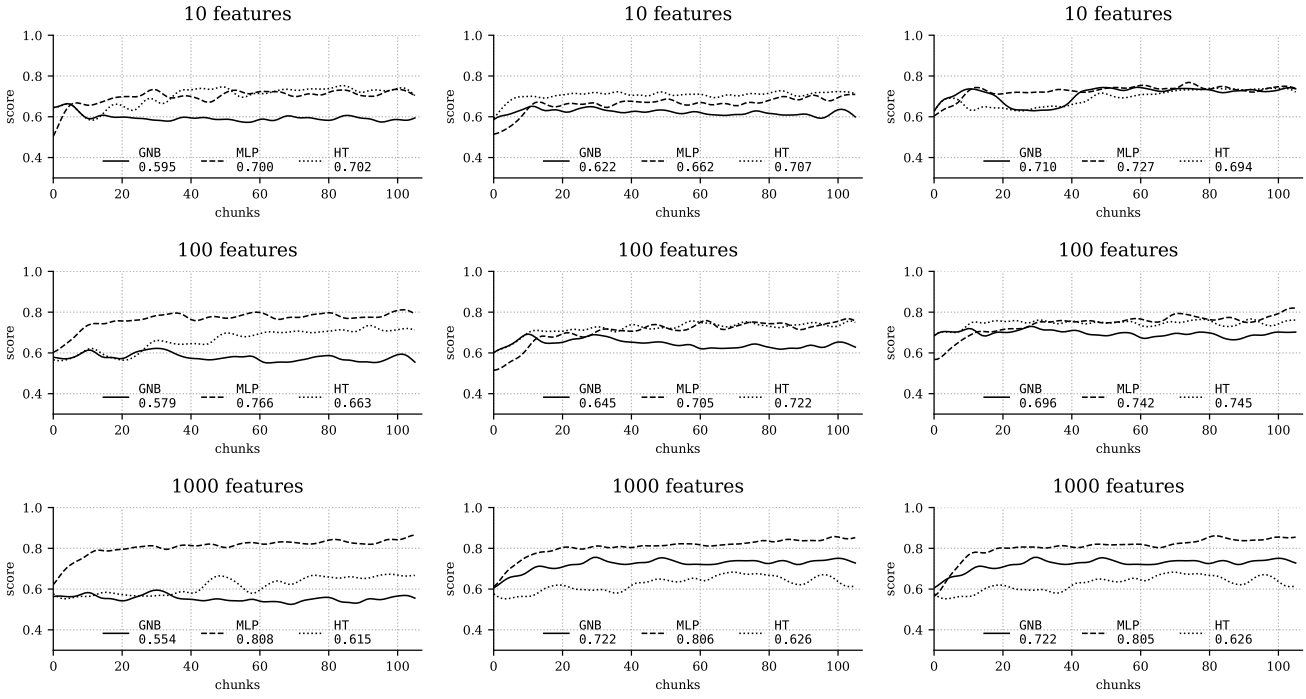
(a) PCA (b) COUNT VECTORIZER (c) FEATURE SELECTION

Fig. 2: Example runs of data stream processing for SEA processing approach.



(a) PCA (a) COUNT VECTORIZER (c) FEATURE SELECTION

Fig. 3: Example runs of data stream processing for ONLINE BAGGING processing approach.



(a) PCA (b) COUNT VECTORIZER (c) FEATURE SELECTION

Fig. 4: Example runs of data stream processing for SINGLE MODEL processing approach.

so it is a preliminary analysis of the effectiveness of typical methods of feature reduction and the construction of stream models for an entirely new problem.

Extensive experimental research on several algorithms from each of the considered aspects of processing confirmed the possibility of constructing systems of this type in an application for data with a stationary nature of the concept, suggesting extraction using the *Principal Components Analysis* algorithm in the construction of *Online Bagging* ensemble encapsulating the *Multi-layer Perceptron* base classifier.

The research presented in work will be continued by considering subsequent analyzes also data dynamics both in prior and posterior probabilities, taking into account different variants.

ACKNOWLEDGEMENT

This work is funded under SocialTruth project, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 825477.

REFERENCES

[1] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *SIGKDD Explor. Newsl.*, vol. 19, no. 1, pp. 22–36, Sep. 2017. [Online]. Available: http://doi.acm.org/10.1145/3137597.3137600

[2] X. Zhang and A. A. Ghorbani, "An overview of online fake news: Characterization, detection, and discussion," *Information Processing & Management*, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306457318306794

[3] M. Choraś, A. Giełczyk, K. Demestichas, D. Puchalski, and R. Kozik, "Pattern recognition solutions for fake news detection," in *IFIP International Conference on Computer Information Systems and Industrial Management*. Springer, 2018, pp. 130–139.

[4] N. Conroy, V. L. Rubin, and Y. Chen, "Automatic deception detection: Methods for finding fake news," *Proceedings of the Association for Information Science and Technology*, vol. 52, pp. 1–4, 01 2015.

[5] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *Proceedings of the 20th International Conference on World Wide Web*, ser. WWW '11. New York, NY, USA: ACM, 2011, pp. 675–684. [Online]. Available: http://doi.acm.org/10.1145/1963405.1963500

[6] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," *Commun. ACM*, vol. 59, no. 7, pp. 96–104, Jun. 2016. [Online]. Available: http://doi.acm.org/10.1145/2818717

[7] S. Afroz, M. Brennan, and R. Greenstadt, "Detecting hoaxes, frauds, and deception in writing style online," in *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, ser. SP '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 461–475. [Online]. Available: https://doi.org/10.1109/SP.2012.34

[8] K. Sharma, F. Qian, H. Jiang, N. Ruchansky, M. Zhang, and Y. Liu, "Combating fake news: A survey on identification and mitigation techniques," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 3, pp. 21:1–21:42, Apr. 2019. [Online]. Available: http://doi.acm.org/10.1145/3305260

[9] Z. Jin, J. Cao, Y. Zhang, J. Zhou, and Q. Tian, "Novel visual and statistical image features for microblogs news verification," *Trans. Multi.*, vol. 19, no. 3, pp. 598–608, Mar. 2017. [Online]. Available: https://doi.org/10.1109/TMM.2016.2617078

[10] B. D. Horne and S. Adali, "This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news," *CoRR*, vol. abs/1703.09398, 2017. [Online]. Available: http://arxiv.org/abs/1703.09398

[11] C. Chen, K. Wu, S. Venkatesh, and X. Zhang, "Battling the internet water army: Detection of hidden paid posters," *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)*, pp. 116–120, 2011.

[12] G. Gravanis, A. Vakali, K. Diamantaras, and P. Karadais, "Behind the cues: A benchmarking study for fake news detection," *Expert Systems with Applications*, vol. 128, pp. 201 – 213, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417419301988

[13] A. Bondielli and F. Marcelloni, "A survey on fake news and rumour detection techniques," *Information Sciences*, vol. 497, pp. 38 – 55, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025519304372

[14] C.-S. Atodiresei, A. Tănăselea, and A. Iftene, "Identifying fake news and fake users on twitter," *Procedia Computer Science*, vol. 126, pp. 451 – 461, 2018, knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 22nd International Conference, KES-2018, Belgrade, Serbia. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050918312559

[15] P. Ksieniewicz, M. Woźniak, B. Cyganek, A. Kasprzak, and K. Walkowiak, "Data stream classification using active learned neural networks," *Neurocomputing*, vol. 353, pp. 74–82, Aug. 2019. [Online]. Available: https://doi.org/10.1016/j.neucom.2018.05.130

[16] S. Wang, L. L. Minku, and X. Yao, "Resampling-based ensemble methods for online class imbalance learning," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 5, pp. 1356–1368, 2015.

[17] P. Ksieniewicz and M. Woźniak, "Imbalanced data classification based on feature selection techniques," in *Intelligent Data Engineering and Automated Learning – IDEAL 2018*. Springer International Publishing, 2018, pp. 296–303. [Online]. Available: https://doi.org/10.1007/978-3-030-03496-2_33

[18] N. Street and Y. Kim, "A streaming ensemble algorithm (sea) for large-scale classification," *Proceedings of the 7Th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 377–382, 01 2001.

[19] A. Cano and B. Krawczyk, "Kappa updated ensemble for drifting data stream mining," *Machine Learning*, vol. 109, no. 1, pp. 175–218, Oct. 2019. [Online]. Available: https://doi.org/10.1007/s10994-019-05840-z

[20] R. P. W. Duin, "The combining classifier: to train or not to train?" in *Object recognition supported by user interaction for service robots*, vol. 2, Aug 2002, pp. 765–770 vol.2.

[21] N. C. Oza, "Online bagging and boosting," in *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, Oct 2005, pp. 2340–2345 Vol. 3.

[22] P. Zyblewski, P. Ksieniewicz, and M. Woźniak, "Classifier selection for highly imbalanced data streams with minority driven ensemble," in *Artificial Intelligence and Soft Computing*. Springer International Publishing, 2019, pp. 626–635. [Online]. Available: https://doi.org/10.1007/978-3-030-20912-4_57

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[24] P. Ksieniewicz and P. Zyblewski, "stream-learn–open-source python library for difficult data stream batch analysis," *arXiv preprint arXiv:2001.11077*, 2020.

[25] J. Montiel, J. Read, A. Bifet, and T. Abdessalem, "Scikit-multiflow: A multi-output streaming framework," *Journal of Machine Learning Research*, vol. 19, no. 72, pp. 1–5, 2018. [Online]. Available: http://jmlr.org/papers/v19/18-251.html

[C6]
Pawel Ksieniewicz. “Combining Random Subspace Approach with smote Oversampling for Imbalanced Data Classification”. W: *Hybrid Artificial Intelligent Systems*. Red.
Hilde Pérez García i in. Cham: Springer International Publishing, 2019, s. 660–673.
ISBN: 978-3-030-29859-3. DOI:
10.1007/978-3-030-29859-3_56

Combining *Random Subspace* Approach with SMOTE Oversampling for Imbalanced Data Classification

Pawel Ksieniewicz(✉)

Wrocław University of Science and Technology, Wrocław, Poland
pawel.ksieniewicz@pwr.edu.pl

Abstract. Following work tries to utilize a hybrid approach of combining *Random Subspace* method and SMOTE oversampling to solve a problem of imbalanced data classification. Paper contains a proposition of the ensemble diversified using Random Subspace approach, trained with a set oversampled in the context of each reduced subset of features. Algorithm was evaluated on the basis of the computer experiments carried out on the benchmark datasets and three different base classifiers.

Keywords: Imbalanced classification · SMOTE · Random Subspace · Classifier ensembles

1 Introduction

A major part of the pattern recognition problems presents the task of classification, in which we train a model capable of assigning new, unknown objects to predefined groups on the basis of a knowledge extracted from a set of labeled patterns [5]. Most classical classification algorithms assume an equal percentage of each class and encounters a problem when the proportions between them are strongly disturbed, tending to favor prediction of the more common one. Data about this characteristic is called *imbalanced data* [22]. Most of the real problems, such as diagnosis of diseases, SPAM-detection or fraud recognition, require detection of events far from normal and therefore are not balanced, which makes necessary to modify the pattern recognition models for their needs [10].

In order to eliminate this problem and to construct a model capable of classifying imbalanced data, three approaches are most commonly used [13]. The first of these are methods of data pre-processing, in which we do not modify the learning process, but we introduce changes in the training set itself. The simplest examples are *random undersampling*, in which we take into training set the full minority class and a random subset of the same size from majority class and *random oversampling*, where we use the full majority class and randomly selected objects of the minority class of the same cardinality, regardless of the repetition of the patterns [6].

H. Pérez García et al. (Eds.): HAIS 2019, LNAI 11734, pp. 660–673, 2019.
https://doi.org/10.1007/978-3-030-29859-3_56

More complex solutions of this type are *oversampling* algorithms, which instead of repeating existing minority class patterns, generate new synthetic objects based on the information contained in their distribution. The most common of them are ADASYN [9] and SMOTE [3], developed into a multitude that also takes into account the distribution of the majority class of varieties such as *Borderline*-SMOTE [8], *Safe-Level*-SMOTE [2] or LN-SMOTE [18].

Another approach is to use *inbuilt mechanisms* into the classifier learning process itself. The most commonly used is the one-class classification, insensitive to the distribution of classes of the problem [14] and the cost-sensitive classification that takes into account the loss-function asymmetric in favor of the minority class [10].

The final group of considered approaches are hybrid solutions that combine preprocessing methods with classifier ensembles. The modifications of *Bagging* and *Boosting* are the most popular [20], but there are also methods based on combining a team of classifiers built on the basis of various methods of oversampling or random split undersampling [15].

An important factor that we must take into consideration during the experiments on imbalanced data is also the metric used to assess the quality of constructed models [12]. Typical accuracy, in a strongly imbalanced problem, gives us results being far from the truth, showing, for example, 90% accuracy when wrongly classifying the entire minority class occurring in one in ten samples. In binary classification problems, therefore, the most-used are taking into account the proportions of the measurement classes F-measure or geometric mean score. In multi-class problems, where the above measures can not be calculated, we use the most often the balanced accuracy score.

Following paper attempts to propose a new hybrid method, based on classifier ensembles built in accordance with the *Random Subspace* [26] principle common for multidimensional data and the SMOTE algorithm used for oversampling of objects in the subspaces of each of the member classifiers [24]. Previous studies have already used a combination of these methods, but *oversampling* is performed there before application of *Random Subspace*, which may not properly use the profits achieved by finding a subspace that allows effective determination of the decision boundary [11].

The main contributions of this work are:

- Proposition of the method of joint use of weighted classifier ensemble obtained by the *Random Subspace* method with the use of SMOTE oversampling.
- Implementation of the proposed method in varieties taking into account the separate use of each of its elements.
- Experimental evaluation of the impact of SMOTE, *Random Subspace* and weighing the ensemble fuser on the quality of imbalanced data classification.

2 Method Design

Random Subspace. The construction of the classifier ensemble gives us two basic difficulties [25]. The first is to provide a diverse pool of classifiers that allow to

perform independent, parallel prediction. We can achieve this by using different classification algorithms or various subsets of the training set. The proposed method uses the second approach, where each member of the committee is built on a random subspace of the training set.

SMOTE. Before training each member of the ensemble, minority oversampling is performed using the basic version of the SMOTE algorithm. Training takes place using the base classifier chosen by the experimenter, which must, however, be a probabilistic classifier, or at least have probabilistic interpretation.

Fuser. The second difficulty before the effective construction of the classifier ensemble is the construction of its *fuser* – the function responsible for making decisions on behalf of the ensemble based on the opinions of its members [16]. Among the concepts used, according to names popular in implementations, there are *hard fusers* – based on voting principles and *soft fusers* – based on support accumulation. The use of probabilistic base classifiers allows in this case the use of a method with accumulation of support, additionally enriched by weighing. The weights of member classifiers in the proposed method will be their quality measured with the *f-measure* metric determined for the training data. Utilized F-score is determined as ratio between duplicated product of precision and recall relative to its sum [21].

The full processing scheme of the method proposed in this paper is presented in Fig. 1. The available training set is divided into a random subsets of features using the *Random Subspace* method and minority class in all the subspaces is independently oversampled using the SMOTE method. The structure of classifiers constructed in this way allows for prediction by support accumulation weighted with *f-measure* obtained on a training set.

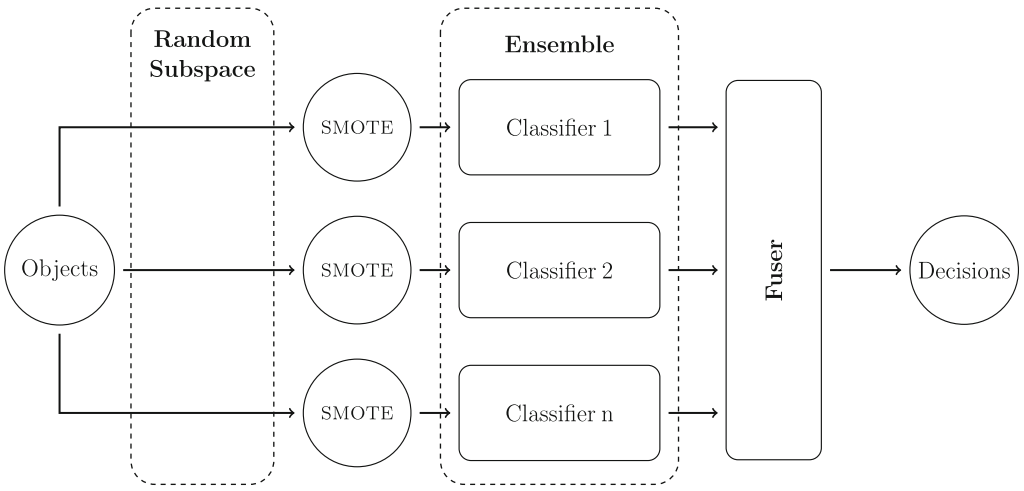


Fig. 1. Random subspace approach to build classifier ensemble

3 Experiments Set-Up

The experimental evaluation was carried out on the basis of 30 datasets with various imbalance ratio, available in the KEEL *data repository* [1]. The application of the *Random Subspace* method was based on arbitrarily set 30 subspaces using three features each [23]. As base classifiers, three standard methods with probabilistic interpretation were adopted:

1. Gaussian Naive Bayes
2. Logistic Regression
3. Support Vector Machine

The method was programmed in accordance with the programming interface of the *scikit-learn* environment [19], so it was possible to use the basic classifiers implemented in it. The used implementation of the SMOTE algorithm was the version available in the *imbalanced-learn* library [17].

The study identified and tested six approaches for each base classifier:

1. Base method.
2. SMOTE oversampling
3. RS – *Random Subspace*
4. RS+SMOTE – Combined *Random Subspace* and SMOTE
5. WRS – *Random Subspace* weighted by *f-score* obtained on a training set
6. WRS+SMOTE – Combined *Random Subspace* and SMOTE weighted by *f-score* obtained on a training set.

In the course of the experiment stratified 5-fold crossvalidation was used [4], measuring quality of all predictions using the *f-score* method. Tests analyzing statistical dependence were carried out using the Wilcoxon test [7]. The full code of the proposed solution as well as the experiments themselves was made available through the public Git repository[1].

4 Experimental Evaluation

The detailed results of the experiments, including the averaged f-measure for all folds of cross-validation along with the evaluation of the statistical dependence of the analyzed solutions for each of the data sets are presented in Tables 1, 2 and 3 respectively for all three base classifiers.

Table 4 contains the ranks determined by ranked Wilcoxon test for each of the analyzed methods and base classifiers. Based on them, assuming the levels of significance .9 and .95, a summary was prepared, counting the cases of advantage of each solution over the others and stored in Table 5.

As may be observed from the obtained results, which was also in line with expectations, regardless of the underlying classifier, using only SMOTE, without

[1] https://github.com/w4k2/wrssmote.

Table 1. Classification results on *Gaussian Naive Bayes*

Dataset	Base method	SMOTE	RS	RS+SMOTE	WRS	WRS+SMOTE
australian	0.719 —	0.718 —	0.755 —	0.795 1,2,3	**0.859** 1,2,3,4	**0.863** 1,2,3,4
glass-0-1-2-3-vs-4-5-6	**0.709** —	**0.718** —	**0.694** —	**0.744** —	**0.680** —	**0.737** —
glass-0-1-4-6-vs-2	**0.219** —	**0.231** —	**0.292** —	**0.270** —	**0.266** —	**0.226** —
glass-0-1-5-vs-2	**0.218** 3	**0.161** 3	0.016 —	**0.198** 3	**0.218** —	**0.176** 3
glass-0-1-6-vs-2	**0.199** —	**0.190** —	**0.192** —	**0.228** —	**0.155** —	**0.193** —
glass-0-1-6-vs-5	**0.760** —	**0.760** —	**0.636** —	**0.557** —	**0.717** —	**0.648** —
glass-0-4-vs-5	**0.960** —	**0.893** —	**0.727** —	**0.727** —	**0.893** —	**0.893** —
glass-0-6-vs-5	**0.893** —	**0.893** —	**0.733** —	**0.670** —	**0.796** —	**0.664** —
glass0	**0.642** —	**0.639** —	**0.633** —	**0.644** —	**0.644** —	**0.649** —
glass1	**0.604** —	**0.638** —	**0.504** —	**0.626** —	**0.603** —	**0.626** —
glass2	**0.189** —	**0.193** —	**0.177** —	**0.181** —	**0.202** —	**0.203** —
glass4	**0.237** —	**0.509** —	**0.200** —	**0.580** —	**0.200** —	**0.567** —
glass5	**0.768** —	**0.768** —	**0.591** —	**0.498** —	**0.659** —	**0.590** —
glass6	**0.772** —	**0.786** —	**0.815** —	**0.826** —	**0.800** —	**0.811** —
heart	**0.802** —	**0.808** —	**0.797** —	**0.800** —	**0.827** —	**0.808** —
hepatitis	0.719 —	**0.851** —	**0.649** —	0.588 —	**0.917** 1,4	**0.893** 4

Table 1. (*continued*)

Dataset	Base method	SMOTE	RS	RS+SMOTE	WRS	WRS+SMOTE
page-blocks-1-3-vs-4	**0.493** —	**0.540** —	**0.470** —	**0.498** —	**0.435** —	**0.471** —
pima	**0.621** 3	**0.665** 3,5	0.523 —	**0.657** 3,5	0.597 3	**0.664** 3,5
shuttle-c0-vs-c4	0.980 —	0.980 —	**0.975** —	**0.981** —	**0.996** —	1.000 1,2
vowel0	**0.709** 3	**0.592** 3	0.057 —	**0.568** 3	**0.426** 3	**0.569** 3
wisconsin	**0.943** —	**0.944** —	**0.959** —	**0.957** —	**0.959** —	**0.959** —
yeast-0-2-5-6-vs-3-7-8-9	**0.262** 3	**0.488** 3,5	0.023 —	**0.377** 3	0.113 —	**0.471** 3,5
yeast-0-2-5-7-9-vs-3-6-8	0.201 —	0.173 —	**0.619** 1,2,4	0.195 —	**0.706** 1,2,4	**0.650** 1,2,4
yeast-0-3-5-9-vs-7-8	**0.269** —	**0.216** 6	**0.244** —	**0.186** —	**0.151** —	**0.080** —
yeast-0-5-6-7-9-vs-4	0.175 —	0.180 —	**0.386** 1,2	0.193 —	**0.429** —	**0.445** 1,2,4
yeast-2-vs-4	0.297 —	0.270 —	**0.574** —	0.312 —	**0.743** 1,2,4	**0.671** 1,2,4
yeast-2-vs-8	0.254 —	0.175 —	**0.683** 1,2,4	0.109 —	**0.658** 2,4	**0.658** 2,4
yeast1	0.457 —	0.458 —	**0.557** 1,2,4,5	0.484 1,2	0.489 1,2	**0.557** 1,2,4,5
yeast3	0.236 —	0.252 —	**0.642** 1,2,4	0.243 —	**0.608** 1,2,4	**0.686** 1,2,4
yeast5	0.154 4	0.192 4	0.181 4	0.107 —	**0.582** 1,2,3,4	**0.485** 1,2,3,4

using Random Subspace, positively influences the quality of classification against the training of the classifier just with the original data set. At the same time, using only *Random Subspace* leads to very poor results, even worse than the base method.

Table 2. Classification results on *Logistic Regression*

Dataset	Base method	SMOTE	RS	RS+SMOTE	WRS	WRS+SMOTE
australian	**0.842** —	**0.846** —	**0.841** —	**0.859** —	**0.859** —	**0.853** —
glass-0-1-2-3-vs-4-5-6	**0.775** —	**0.807** —	**0.547** —	**0.747** —	**0.583** —	**0.720** —
glass-0-1-4-6-vs-2	**0.000** —	**0.258** 1,3,5	**0.000** —	**0.290** —	**0.000** —	**0.286** —
glass-0-1-5-vs-2	0.000 —	**0.184** 1,3,5	0.000 —	**0.172** 1,3,5	0.000 —	**0.181** 1,3,5
glass-0-1-6-vs-2	**0.000** —	**0.176** 1,3,5	**0.000** —	**0.196** 1,3,5	**0.000** —	**0.282** —
glass-0-1-6-vs-5	**0.149** —	**0.451** 3,5	0.000 —	**0.519** 3,5	0.000 —	**0.519** 3,5
glass-0-4-vs-5	**0.367** —	**0.720** 3,5	0.000 —	**0.829** 3,5	0.162 —	**0.829** 3,5
glass-0-6-vs-5	**0.347** 3,5	**0.584** 3,5	0.000 —	**0.638** 3,5	0.000 —	**0.657** 3,5
glass0	**0.516** 3	**0.678** 3,5	0.237 —	**0.675** 3,5	0.369 —	**0.678** 3,5
glass1	0.243 3,5	**0.568** 1,3,5	0.000 —	**0.553** 1,3,5	0.133 3	**0.538** 1,3,5
glass2	0.000 —	**0.167** 1,3,5	0.000 —	**0.202** 1,3,5	0.000 —	**0.195** 1,3,5
glass4	0.167 —	**0.578** 1,3	0.000 —	**0.591** 1,3	**0.200** —	**0.566** 1,3
glass5	**0.149** —	**0.510** 3,5	0.000 —	**0.465** 3,5	0.000 —	**0.428** 3,5
glass6	**0.759** —	**0.832** —	**0.570** —	**0.825** —	**0.742** —	**0.825** —

Table 2. (*continued*)

Dataset	Base method	SMOTE	RS	RS+SMOTE	WRS	WRS+SMOTE
heart	**0.829**	**0.817**	**0.804**	0.794	**0.829**	0.794
	4,6	—	—	—	—	—
hepatitis	**0.919**	**0.875**	**0.912**	**0.904**	**0.912**	**0.904**
	—	—	—	—	—	—
page-blocks-1-3-vs-4	**0.560**	**0.683**	**0.421**	**0.537**	**0.442**	**0.501**
	—	—	—	—	—	—
pima	0.618	**0.684**	0.432	**0.653**	0.560	**0.665**
	3,5	1,3,5	—	3,5	3	3,5
shuttle-c0-vs-c4	**0.996**	1.000	**0.996**	**0.996**	**0.996**	**0.996**
	—	—	—	—	—	—
vowel0	**0.579**	**0.652**	0.087	**0.637**	**0.550**	**0.628**
	3	3	—	3	3	3
wisconsin	**0.947**	**0.956**	**0.950**	**0.959**	**0.945**	**0.959**
	—	—	—	—	—	—
yeast-0-2-5-6-vs-3-7-8-9	0.019	**0.461**	0.000	**0.423**	0.000	**0.420**
	—	1,3,5	—	1,3,5	—	1,3,5
yeast-0-2-5-7-9-vs-3-6-8	0.236	**0.587**	0.000	**0.563**	0.000	**0.575**
	3,5	1,3,5	—	3,5	—	1,3,5
yeast-0-3-5-9-vs-7-8	0.067	**0.275**	0.000	**0.244**	0.067	**0.261**
	—	1,3,5	—	1,3,5	—	1,3,5
yeast-0-5-6-7-9-vs-4	0.000	**0.468**	0.000	**0.466**	0.000	**0.488**
	—	1,3,5	—	1,3,5	—	1,3,5
yeast-2-vs-4	0.170	**0.690**	0.000	**0.664**	0.103	**0.680**
	—	1,3,5	—	1,3,5	—	1,3,5
yeast-2-vs-8	0.080	**0.546**	0.000	**0.480**	0.080	**0.658**
	—	1,3,5	—	1,3,5	—	1,3,5
yeast1	0.329	**0.584**	0.062	**0.565**	0.241	**0.593**
	3	1,3,5	—	1,3,5	3	1,3,5
yeast3	0.109	**0.671**	0.000	**0.681**	0.056	**0.686**
	3	1,3,5	—	1,3,5	—	1,3,5
yeast5	0.000	**0.480**	0.000	**0.461**	0.000	**0.462**
	—	1,3,5	—	1,3,5	—	1,3,5

Table 3. Classification results on *Support Vector Machines*

Dataset	Base method	SMOTE	RS	RS+SMOTE	WRS	WRS+SMOTE
australian	0.000	0.247	0.693	**0.795**	0.675	**0.813**
	—	—	1,2	1,2,3,5	1,2	1,2,3,5
glass-0-1-2-3-vs-4-5-6	**0.741**	**0.864**	**0.674**	**0.835**	**0.666**	**0.842**
	—	—	—	—	—	—
glass-0-1-4-6-vs-2	0.000	**0.296**	0.000	**0.366**	0.000	**0.375**
	—	1,3,5	—	1,3,5	—	1,3,5
glass-0-1-5-vs-2	0.000	**0.264**	0.000	**0.377**	0.000	**0.389**
	—	1,3,5	—	1,3,5	—	1,3,5
glass-0-1-6-vs-2	0.000	**0.286**	0.000	**0.364**	0.000	**0.380**
	—	1,3,5	—	1,3,5	—	1,3,5
glass-0-1-6-vs-5	0.000	**0.544**	0.000	**0.492**	0.000	**0.466**
	—	1,3,5	—	1,3,5	—	1,3,5
glass-0-4-vs-5	**0.431**	**0.762**	0.031	**0.629**	**0.693**	**0.629**
	3	3	—	3	3	3
glass-0-6-vs-5	**0.467**	**0.800**	0.000	**0.700**	0.000	**0.700**
	3,5	3,5	—	3,5	—	3,5
glass0	0.460	**0.699**	0.198	**0.731**	0.455	**0.738**
	3	1,3,5	—	1,3,5	3	1,3,5
glass1	**0.578**	**0.603**	0.184	**0.596**	**0.618**	**0.588**
	3	3	—	3	3	3
glass2	0.000	**0.247**	0.000	**0.287**	0.000	**0.309**
	—	1,3,5	—	1,3,5	—	1,3,5
glass4	**0.671**	**0.772**	0.000	**0.762**	**0.593**	**0.781**
	3	3	—	3	3	3
glass5	0.000	**0.403**	0.000	**0.537**	0.000	**0.472**
	—	1,3,5	—	1,3,5	—	1,3,5
glass6	**0.824**	**0.809**	**0.827**	**0.847**	**0.827**	**0.847**
	—	—	—	—	—	—
heart	0.000	0.063	**0.764**	**0.802**	0.657	**0.714**
	—	1	1,2,5	1,2,5	1,2	1,2
hepatitis	**0.912**	**0.912**	**0.912**	**0.927**	**0.912**	**0.906**
	—	—	—	—	—	—

Table 3. (*continued*)

Dataset	Base method	SMOTE	RS	RS+SMOTE	WRS	WRS+SMOTE
page-blocks-1-3-vs-4	0.000	0.037	0.114	**0.478**	0.057	**0.398**
	—	—	—	1,2,3,5	—	1,2,3,5
pima	0.000	0.015	0.000	**0.400**	0.000	**0.368**
	—	—	—	1,2,3,5	—	1,2,3,5
shuttle-c0-vs-c4	0.190	0.606	**0.987**	**0.983**	**0.987**	**0.987**
	—	1	1,2	1,2	1,2	1,2
vowel0	**0.547**	**0.566**	0.208	**0.706**	**0.600**	**0.678**
	—	3	—	3	3	3
wisconsin	**0.933**	**0.936**	**0.961**	**0.968**	**0.961**	**0.968**
	—	—	—	—	—	—
yeast-0-2-5-6-vs-3-7-8-9	0.000	**0.493**	0.000	**0.420**	0.118	**0.402**
	—	1,3,5	—	1,3,5	—	1,3,5
yeast-0-2-5-7-9-vs-3-6-8	0.000	**0.652**	0.053	**0.559**	0.167	**0.579**
	—	1,3,5	—	1,3,5	—	1,3,5
yeast-0-3-5-9-vs-7-8	0.000	**0.198**	0.000	**0.239**	**0.114**	**0.241**
	—	1,3	—	1,3	—	1,3
yeast-0-5-6-7-9-vs-4	0.000	**0.472**	0.000	**0.495**	0.000	**0.497**
	—	1,3,5	—	1,3,5	—	1,3,5
yeast-2-vs-4	0.000	**0.697**	**0.463**	**0.664**	**0.589**	**0.682**
	—	1	1	1	1	1
yeast-2-vs-8	**0.613**	**0.658**	0.000	**0.487**	**0.658**	**0.658**
	3	3	—	3	3	3
yeast1	0.073	**0.573**	0.097	**0.574**	0.345	**0.584**
	—	1,3,5	—	1,3,5	1,3	1,3,5
yeast3	0.000	0.681	0.000	**0.721**	**0.759**	**0.701**
	—	1,3	—	1,3	1,2,3	1,3
yeast5	0.000	**0.457**	0.000	**0.522**	0.000	**0.523**
	—	1,3,5	—	1,3,5	—	1,3,5

Enhancing the Random Subspace method with weighing, which – thanks to the f-measure – is realized in the context of the imbalanced problem, makes it to be more effective than the base method, although it still does not affect the quality of classification as positively as SMOTE.

Table 4. Ranks computed by the Wilcoxon test

	Base method	SMOTE	RS	RS+SMOTE	WRS	WRS+SMOTE
GNB	-	191.0	258.0	245.0	196.0	132.0
SMOTE	274.0	-	273.0	279.0	194.0	137.0
RS	207.0	192.0	-	240.0	102.5	75.0
RS+SMOTE	220.0	186.0	195.0	-	153.0	100.0
WRS	269.0	241.0	362.5	312.0	-	203.5
WRS+SMOTE	333.0	298.0	360.0	335.0	231.5	-
LR	-	10.0	416.5	18.0	389.0	22.0
SMOTE	455.0	-	460.0	283.0	453.0	235.0
RS	18.5	5.0	-	4.0	52.0	4.0
RS+SMOTE	417.0	182.0	431.0	-	429.0	187.0
WRS	46.0	12.0	383.0	6.0	-	7.0
WRS+SMOTE	413.0	230.0	431.0	248.0	428.0	-
SVM	-	2.0	213.0	7.0	101.5	1.0
SMOTE	433.0	-	371.0	137.0	348.5	114.0
RS	222.0	64.0	-	1.0	114.0	5.0
RS+SMOTE	458.0	328.0	464.0	-	431.0	234.0
WRS	363.5	116.5	321.0	34.0	-	26.5
WRS+SMOTE	464.0	321.0	430.0	231.0	438.5	-

The extension of the basic form of *Random Subspace* with SMOTE leads to slightly better results than using only SMOTE, which suggests that the positive influence on the quality of classification by both methods may be independent and it may be complemented. This is confirmed by the use of the full proposal, based on the weighted Random Subspace from SMOTE, whose quality is definitely the best in the competition and except one case (dataset *heart* with the Linear Regression) there are no situations where it is not among the best approaches in the considered pool.

The *Random Subspace* method is particularly popular in the case of multidimensional data, being a solution to the significant problem of the curse of dimensionality. Lowering the number of features analyzed by each model, using the same cardinality of patterns, reduces the decision space, and thus compacts the samples in space. On the other hand, the role of SMOTE is to equalize the density of the occurrence of patterns in the problem by compacting the objects of the minority class. Using both methods together, controlling the influence of each subspace on the final prediction of the ensemble by weighing it with a

Table 5. Summary of the Wilcoxon test. •= the method in the row improves the method of the column. ○= the method in the column improves the method of the row. Upper diagonal of level significance $\alpha = 0.9$, Lower diagonal level of significance $\alpha = 0.95$

	Base method	SMOTE	RS	RS+SMOTE	WRS	WRS+SMOTE
GNB	-					○
SMOTE		-				○
RS			-		○	○
RS+SMOTE				-		○
WRS			•		-	
WRS+SMOTE	•		•	•		-
LR	-	○	•	○	•	○
SMOTE	•	-	•		•	
RS	○	○	-	○	○	○
RS+SMOTE	•		•	-	•	
WRS	○	○	•	○	-	○
WRS+SMOTE	•		•		•	-
SVM	-	○		○	○	○
SMOTE	•	-	•	○	•	○
RS		○	-	○	○	○
RS+SMOTE	•	•	•	-	•	
WRS	•	○	•	○	-	○
WRS+SMOTE	•	•	•		•	-
$\alpha = .9$	2	6	0	7	4	11
$\alpha = .95$	2	6	0	7	4	10

measure adequate to the imbalanced problem may lead to close to the optimal placement of training patterns in the classification space, and thus lead to a model with a high discriminatory ability, as shown by carried out experiments.

5 Summary

This paper proposes the use of a classifier ensemble diversified using the *Random Subspace* approach and trained on sets oversampled independently in each subspace using the SMOTE algorithm. The experiments performed for its needs are testing the quality of such solution with relation to both concepts present in it, using 30 imbalanced data sets and three probabilistic base classifiers.

As shown by the results of experiments, it is a promising approach, able to effectively use the advantages of both methods leading to a better solution than each of them individually. Research shows that the *Random Subspace* method, although in itself, does not allow to improve the prediction of imbalanced data, in the weighted option may positively affect the achieved quality of classification. Combining it with the creation of synthetic patterns in the subspace areas of the problem gives an effective solution with high usefulness in the processing of this type of problems.

Acknowledgements. This work was supported by the Polish National Science Center under the grant no. UMO- 2015/19/B/ST6/01597 and by the statutory fund of the Faculty of Electronics, Wroclaw University of Science and Technology.

References

1. Alcalá-Fdez, J., et al.: Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. J. Mult.-Valued Log. Soft Comput. **17**, 255–287 (2011)
2. Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C.: Safe-Level-SMOTE: safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In: Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, T.-B. (eds.) PAKDD 2009. LNCS (LNAI), vol. 5476, pp. 475–482. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01307-2_43
3. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
4. Diamantidis, N., Karlis, D., Giakoumakis, E.A.: Unsupervised stratification of cross-validation for accuracy estimation. Artif. Intell. **116**(1–2), 1–16 (2000)
5. Dietterich, T.G.: Ensemble methods in machine learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45014-9_1
6. García, S., Herrera, F.: Evolutionary undersampling for classification with imbalanced datasets: proposals and taxonomy. Evol. Comput. **17**(3), 275–306 (2009)
7. Gehan, E.A.: A generalized Wilcoxon test for comparing arbitrarily singly-censored samples. Biometrika **52**(1–2), 203–224 (1965)
8. Han, H., Wang, W.-Y., Mao, B.-H.: Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) ICIC 2005. LNCS, vol. 3644, pp. 878–887. Springer, Heidelberg (2005). https://doi.org/10.1007/11538059_91
9. He, H., Bai, Y., Garcia, E.A., Li, S.: ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence). IEEE, June 2008
10. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. **9**, 1263–1284 (2008)
11. Huang, H.Y., Lin, Y.J., Chen, Y.S., Lu, H.Y.: Imbalanced data classification using random subspace method and SMOTE. In: The 6th International Conference on Soft Computing and Intelligent Systems, and the 13th International Symposium on Advanced Intelligence Systems. IEEE, November 2012

12. Jeni, L.A., Cohn, J.F., De La Torre, F.: Facing imbalanced data-recommendations for the use of performance metrics. In: 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, pp. 245–251. IEEE (2013)
13. Krawczyk, B.: Learning from imbalanced data: open challenges and future directions. Prog. Artif. Intell. **5**(4), 221–232 (2016)
14. Krawczyk, B., Woźniak, M., Herrera, F.: On the usefulness of one-class classifier ensembles for decomposition of multi-class problems. Pattern Recogn. **48**(12), 3969–3982 (2015)
15. Ksieniewicz, P.: Undersampled majority class ensemble for highly imbalanced binary classification. In: Torgo, L., Matwin, S., Japkowicz, N., Krawczyk, B., Moniz, N., Branco, P. (eds.) Proceedings of the Second International Workshop on Learning with Imbalanced Domains: Theory and Applications. Proceedings of Machine Learning Research, PMLR, ECML-PKDD, Dublin, Ireland, vol. 94, pp. 82–94, 10 September 2018
16. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley, Hoboken (2004)
17. Lemaître, G., Nogueira, F., Aridas, C.K.: Imbalanced-learn: a Python toolbox to tackle the curse of imbalanced datasets in machine learning. J. Mach. Learn. Res. **18**(1), 559–563 (2017)
18. Maciejewski, T., Stefanowski, J.: Local neighbourhood extension of SMOTE for mining imbalanced data. In: 2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM). IEEE, April 2011
19. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**(Oct), 2825–2830 (2011)
20. Quinlan, J.R., et al.: Bagging, boosting, and C4. 5. In: AAAI/IAAI, vol. 1, pp. 725–730 (1996)
21. Sasaki, Y., et al.: The truth of the F-measure. Teach Tutor Mater **1**(5), 1–5 (2007)
22. Sun, Y., Wong, A.K., Kamel, M.S.: Classification of imbalanced data: a review. Int. J. Pattern Recogn. Artif. Intell. **23**(04), 687–719 (2009)
23. Topolski, M.: Multidimensional MCA correspondence model supporting intelligent transport management. Arch. Transp. Syst. Telemat. **11**, 52–56 (2018)
24. Topolski, M.: Algorithm of multidimensional analysis of main features of PCA with blurry observation of facility features detection of carcinoma cells multiple myeloma. In: Burduk, R., Kurzynski, M., Wozniak, M. (eds.) CORES 2019. AISC, vol. 977, pp. 286–294. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-19738-4_29
25. Wozniak, M.: Hybrid Classifiers: Methods of Data, Knowledge, and Classifier Combination. Studies in Computational Intelligence, vol. 519. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40997-4
26. Yu, G., Zhang, G., Domeniconi, C., Yu, Z., You, J.: Semi-supervised classification based on random subspace dimensionality reduction. Pattern Recogn. **45**(3), 1119–1135 (2012)

[C7]
Paweł Ksieniewicz i in. “Data stream classification using active learned neural networks”. W: *Neurocomputing* 353 (2019), s. 74–82. DOI: 10.1016/j.neucom.2018.05.130

Neurocomputing 353 (2019) 74–82

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Data stream classification using active learned neural networks

Paweł Ksieniewicz[a,*], Michał Woźniak[a], Bogusław Cyganek[b], Andrzej Kasprzak[a], Krzysztof Walkowiak[a]

[a] Department of Systems and Computer Networks, Wrocław University of Science and Technology, Wyb. Wyspiańskiego 27, Wrocław 50–370, Poland
[b] AGH University of Science and Technology Al. Mickiewicza 30, Kraków 30–059, Poland

ARTICLE INFO

Article history:
Received 12 January 2018
Revised 8 April 2018
Accepted 26 May 2018
Available online 13 March 2019

Keywords:
Pattern classification
Data stream
Active learning
Concept drift
Forgetting

ABSTRACT

Due to variety of modern real-life tasks, where analyzed data is often not a static set, the data stream mining gained a substantial focus of machine learning community. Main property of such systems is the large amount of data arriving in a sequential manner, which creates an endless stream of objects. Taking into consideration the limited resources as memory and computational power, it is widely accepted that each instance can be processed up once and it is not remembered, making reevaluation impossible. In the following work, we will focus on the data stream classification task where parameters of a classification model may vary over time, so the model should be able to adapt to the changes. It requires a forgetting mechanism, ensuring that outdated samples will not impact a model. The most popular approaches base on so-called *windowing*, requiring storage of a batch of objects and when new examples arrive, the least relevant ones are forgotten. Objects in a new window are used to retrain the model, which is cumbersome especially for *online* learners and contradicts the principle of processing each object at most once. Therefore, this work employs inbuilt forgetting mechanism of neural networks. Additionally, to reduce a need of expensive (sometimes even impossible) object labeling, we are focusing on *active learning*, which asks for labels only for *interesting* examples, crucial for appropriate model upgrading. Characteristics of proposed methods were evaluated on the basis of the computer experiments, performed over diverse pool of data streams. Their results confirmed the convenience of proposed strategy.

1. Introduction and related works

Classification is one of the most frequent decision task which aims at assigning an observed object to one of the predefined categories. There are plethora of solutions [1], nonetheless, there are still a few issues that need to be discussed for contemporary pattern recognition systems. We have to realize that modern machine learning algorithms should build the predictive models on the basis of huge, rapidly changing databases, i.e., we have to propose efficient procedures which can produce accurate classifiers on the basis of so-called data streams. This issue is still a focus of diligent studies, because many classifiers cannot upgrade their models using recent observations, and they usually do not take into consideration that the statistical dependencies between the attributes, describing incoming objects, and their correct classification may change over time.

1.1. Concept drift

Aforementioned phenomenon is called a *concept drift* [2] and its appearances have become a challenge for many practical tasks, such as medical diagnosis (surgery prediction) [3], computer systems security (SPAM filtering, IPS design) [4,5], marketing (client profiling) [6], or banking (fraud detection) [7] to enumerate only a few.

There are several taxonomies of *concept drift*, based on its quickness (sudden shift and smooth ones, as incremental changes) or impact on the *posterior* probabilities [2,8] (*real concept drift* or *virtual concept drift*). Considering classification task, the *real drift* is crucial, because, in opposite to *virtual*, it can significantly change the shape of the decision boundaries.

Changes may be discovered by monitoring the unlabeled data and observing novelties related to the presence of *outliers* or by monitoring classification accuracy [8,9]. However, to properly detect the *real concept drift* we require the labels, because detectors based on unlabeled examples do not guarantee to sense the *real* drift, while the *virtual* one may be detected precisely [10].

* Corresponding author.
E-mail addresses: pawel.ksieniewicz@pwr.edu.pl (P. Ksieniewicz), michal.wozniak@pwr.edu.pl (M. Woźniak), cyganek@agh.edu.pl (B. Cyganek), andrzej.kasprzak@pwr.edu.pl (A. Kasprzak), krzysztof.walkowiak@pwr.edu.pl (K. Walkowiak).

https://doi.org/10.1016/j.neucom.2018.05.130
0925-2312/

1.2. Data stream classification

Development of the methods efficiently tackling phenomenon of *concept drift*, have become an important research challenge for machine learning community. Basically, we may consider the following approaches to deal with it:

- *Frequently retraining a classification model, when data comes.* It is costly and practically unviable. Moreover, in the case, when a drift does not appear, rebuilding is needless, but if a model shift occurs rapidly, the time laps of a new model may be unacceptable.
- *Detecting* concept drift*by monitoring incoming data.* Classifier is retrained on the basis of newly collected objects if the changes of accuracy is significant *enough*.
- *Constant classifier updating*. It uses incremental learning methods that allow to add new training observations during a classifier exploiting or by implementing forgetting mechanisms as dataset weighting or windowing [11].

Gama et al. [11] analyze adaptive algorithms that can react to changes in data streams. Authors enumerate two main concepts of memory and forgetting. Firstly, let's focus on the *online* learners [12], which have to meet the following requirements:

- The classifier learning algorithm has limited memory and processing time at its disposal.
- Every incoming learning object may be processed at most once over a training and it is not memorized, because of aforementioned limitations.
- The training can be paused at any time, but the accuracy of *online* trained classifier should be the same or higher than the quality of a model trained on the whole batch of data collected by then.

The *online* learners are seen as naturally adaptive methods, because they continuously update the predictive model on the basis of incoming observations and they behave pretty well especially for slow changing data streams. There are several propositions of such algorithms, which should be mentioned, as WINNOW [13] or VFDT [14]. Nevertheless, the main drawback of *online* learners is slow adaptation to sudden concept drift, therefore the several works are trying to combat with this problem by control the model updating on the basis of new incoming observations [15]. We have also mention the works where authors propose how to explicitly react to changes in data streams as STAGGER [16], DWM [17], or GT2FC [18] to enumerate only a few.

One may also mentions algorithms that incorporate the *forgetting mechanism* in the form of windowing or data weighting. This approach assumes that examples come in recent time are the most significant, because they are consistent with the present context. Nevertheless, their relevance decreases in the process of time. Therefore, taking into consideration only the latest incoming objects seems to be reasonable, because it helps to collect a dataset representing the present context. We may enumerate the following strategies:

- Selecting the learning examples by means of a sliding data window that cuts off older observations [2].
- Weighting learning instances according to their relevance [19].
- Applying bootstraping-based algorithms as *boosting*, that focus on incorrect classifier objects [20].

One may observe that two main forgetting mechanisms are usually employed. For typical windowing *abrupt forgetting* is frequently used, i.e., the outdated (old) observations are not used to update the model and they are also removed from memory, while *gradual forgetting* means that no observation is discarded from memory, but the observations are associated with weights related to the period of their staying in memory [11]. This approach requires implementation of a decay function, which ensures that old observations do not impact strongly to the recent model. There are several examples of decay functions, as linear one proposed by Koychev [21] or exponential decay function proposed by Klinkerberg [22]. Nevertheless, this approach has one significant drawbacks, because it requires that all observations have to be stored in the memory. This assumption does not fulfill the requirement that we have the limited memory at our disposal. Therefore, if the weight of a given object is less than a given threshold then the object is removed from the memory [23]. One has also mention the interesting researches on *Reservoir Sampling*, where a representative sample are obtained for the analyzed data stream [24].

On the other hand, the main problem of using a sliding window approach is to properly set the window size. The shorter window allows to concentrate on the emerging context, though data may not be representative for a longer lasting context. On the other hand, the bigger window may include instances representing different contexts [25]. Thus, we may find the propositions where the fixed window size is used, i.e., the fixed number of recent incoming examples are stored, or the number of the memorized examples may vary over time. Usually, the windows size depends on drift detector output, i.e., if drift is detected then the window size is decreased. It is worth mentioning FLORA2 [2], which is recognized as the first method, where adaptive windowing had been used, but its descendants as FLORA3 and FLORA4 should be instanced, because they are able to deal with recurring concept drift and noisy data, again several works propose to adjust the window size dynamically, as ADWIN2 [26], or [27] where multiple window approach is discussed.

In this work, we decided to benefit from the phenomenon called *catastrophic forgetting* [28], which is unnecessary for the multi-task learning, because neural networks exposed on the data relevant to new task have the tendency to abruptly forget the knowledge of the previously learned ones. This forgetting occurs specifically when the network is trained sequentially on multiple tasks, because the weights in the network that are important for the previous tasks are changed to meet the new objectives [29].

This adverse behavior for multi-task learning is highly desirable in classifier updating for the drifted data stream, because artificial neural networks should naturally forget outdated concept by rebuilding their internal knowledge representation to properly solve incoming problem.

1.3. Labeling

In this work we will mostly focus on the labeling cost. Majority of the data stream classifiers produce models on the basis of supervised learning algorithms. For some practical tasks we are able to obtain true labels with reasonable cost and time (weather prediction), but for the most of practical tasks, labeling requires human effort or access to the kind of an oracle (e.g., for medical diagnosis – human expert should always verify the diagnosis), thus it is usually very expensive. Let us notice that labels are usually assigned by human experts and therefore they can not label all the new examples if they come too fast, e.g., for SPAM filtering – user should confirm the decision if incoming mail is legitimate or not, i.e., the continuous access to the human expert should be granted. Additionally, sometimes the proper classification is available with a long delay (e.g., for the true label for credit approval is available ca. 2 years after the decision).

Therefore methods of classifier design which could produce the recognition system on the basis of a partially labeled set of examples (especially if learner could point out the interesting example to be labeled) are still very desirable goal [30].

Žliobaitė et al. [31] discuss the theoretical framework for predictive model learning using active learning approach and discuss tree labeling strategies. Kurlej and Wozniak [32] propose the active learning approach for minimal distance classifier applied to drifted streams, where decision about labeling depends on the distance between a given example to the decision boundary. In their later works [25,33] they analyze selected characteristics of such an approach.

Nguyen et al. [34] develop an incremental algorithm CSL-*stream*, that performs clustering and classification at the same time. Mohamad [35] propose similar data stream classifier, which combines uncertainty and density-based querying criteria. Korycki and Krawczyk [36] apply active learning strategy to classify data streams and combine it with self-labeling approach.

The windowing is very accurate for the models which does not require learning with *lazy learners* [37], but for the classifiers based on models, the data window shift requires to rebuild the model on the basis of data in a recent window. Our work will focus on a hybrid active learning approach which is combination of *online* learning approach and sliding windows method with forgetting. Because of the fact, that implementation of the forgetting mechanism requires additional memory to remember the data in window and additional computational effort to rebuild the model when the window is shifted, we have been looking for a classification model with inbuilt forgetting mechanism. It will make our method highly suitable to real-life data stream mining with a limited budget.

1.4. Contributions

In a nutshell, the contributions of this work are as follows:

- Proposition of the active learning strategy which makes a decision for each classification model on the basis of support function.
- Employing *catastrophic forgetting* phenomenon as forgetting mechanism for neural network classifiers.
- Experimental evaluation of the proposed approach on the basis of diverse benchmark datasets and a detailed comparison with the semi-supervised and fully supervised strategies.

2. Proposition of active learning neural network classifier for data stream

Firstly, let us formalize the classification model. Usually, a classifier Ψ makes a decision using so-called support functions which return support for each class [1]

$$\mathcal{F} = \{F_1, F_2, \ldots, F_M\} \tag{1}$$

To make a decision, Ψ usually uses the maximum rule

$$\Psi(x) = \max_{k \in \mathcal{M}} (F_k(x)), \tag{2}$$

where x is the feature vector, k stands for the label from finite set of possible classes $\mathcal{M} = \{1, 2, \ldots, M\}$.

We assume that if a decision about an object is supported by the high value of the support, then the label is not so *"interesting"*, because it probably will not have a significant impact on the classification model improvement. If support functions have probabilistic interpretation, e.g., they are *posterior* probabilities (in the case of so-called 0-1 loss function) [1], then high value of the support function associated with a chosen class means that the probability of misclassification is very low. Otherwise, if the difference among the support function values is low, i.e., the decision is very uncertain.

Let us propose RSFD (*Relative Support Function Difference*) function, which measures average difference among the highest support and support for the rest of the classes for a given observation x

$$RSFD(x) = \frac{\sum_{i=1}^{M} \left[\max_{k \in \mathcal{M}} (F_k(x)) - F_i(x) \right]}{M - 1} \tag{3}$$

The graphical interpretation of RSFD is presented in Fig. 1.

This approach is similar to the *classification with reject option* [38,39], where the trade-off between the error and rejection rate is considered. Basically, in this approach, the decision is made only in a case if the maximum support is high enough (or difference between maximum support and supports for the rest of classes is significant), otherwise the classifier rejects the decision, instead of a giving label, answering *"I do not know"*.

The proposed framework works as the block classifier [40], because it collects the data in the form of chunk, but for each chunk the *online* learner is used. To avoid the influence of object order on a decision about labeling, samples in each chunk are randomized before processing.

The decision about the object labeling depends on two parameters:

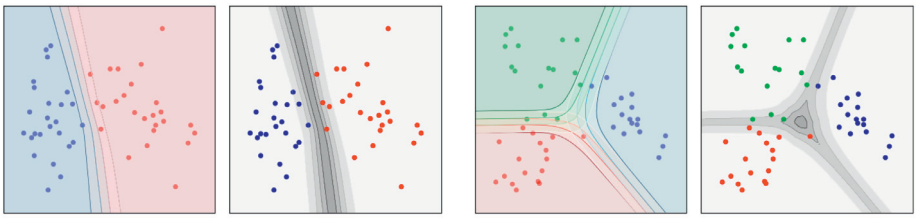
- *threshold* – responsible for choosing the *"interesting"* examples, i.e., if relative support function difference is lower than a given threshold the object seems to be interesting and the learning algorithm is asking for its label.
- *given_budget* – the label will be assigned only in the case if its budget related to a given chunk allows to pay for it. For each chunk only limited percentage of the objects could be labeled.

The idea of the algorithm is presented in Algorithm 1.

Algorithm 1 Active learning classifier for data stream.

Require: input data stream,
n - data chunk size,
M - number of classes,
$incremental\_training\_procedure()$ - classifier training procedure,
$label()$ - function which returns label of a given example,
$classifier()$ - classification model,
$F_1(), F_2(), \ldots, F_M()$ - support functions using by $classifier()$,
m - number of examples required to initialize $classifier()$,
$given\_budget$ - max. percent of labeled example in a chunk,
$treshold$

```
i ← 0
for j = 0 to m do
    ask for the label of the jth example x_j
    classifier ← incremental_training_procedure(x_j, label(x_j))
end for
repeat
    budget ← floor(n * given_budget)
    collect new data chunk DS_i = {x_i^1, x_i^2, ..., x_i^n}
    set random order of collected examples in DS_i
    for j = 0 to n do
        if RFSD(x) < treshold then
            if budget > 0 then
                ask for the label of the jth example x_j
                classifier() ← incremental_training_procedure
                                (x_j, label(x_j))
                budget ← budget − 1
            end if
        end if
    end for
    i ← i + 1
until end of the input data stream
```



(a) binary problem (b) multiclass problem

Fig. 1. An example of the supports for each of classes (left) and RSFD (*Relative Support Function Difference*) (right) for 2 and 3 class problems.

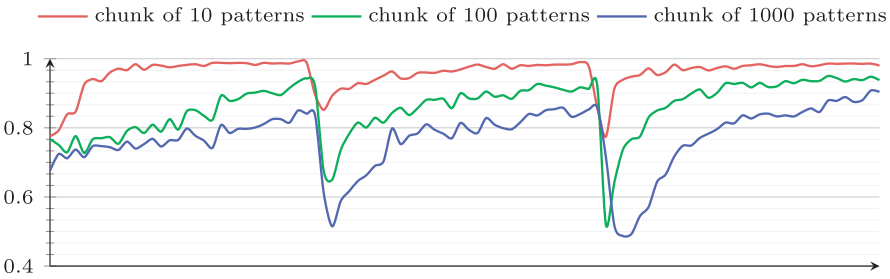


Fig. 2. Influence of chunk size on a learning curve (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.).

3. Inbuilt forgetting mechanism of artificial neural networks

Because we would like to employ inbuilt forgetting mechanism of artificial neural networks, therefore let us shortly present their main properties when they are used as the data stream classifiers. For each simulations presented in Sections 3 and 4, the Multilayer Perceptron (MLP) has been used with one hidden layer. In this section the chosen results of the simulation research are presented, but the additional results, as well as the used software may be found in article repository.[1] All the examples presented below have been prepared with data streams generated using *Radial Basis Function* for different concept drift types.

Fig. 2 shows learning curves of three identical neural networks (MLP with 100 neurons in a hidden layer) classifying a stream (10k instances with two sudden drifts) using different size of data chunk.

As we can observe, a smaller chunk (red line) means a more dynamic learning curve (faster accuracy growth). This is most likely due to the increased frequency of repeating the learning procedure, even with smaller portions of information, leading faster to establishment of a generalization power. Simultaneously, a larger chunk (blue line) leads to a larger decrease of accuracy caused by a sudden drift. At the other side, the smaller chunk, with extreme of incremental learning when we run a learning procedure over every single object, also leads to a higher computational complexity.

Fig. 3 shows the reaction of different neural networks (10, 50 and 500 neurons) to different types of concept drift (from sudden drift to stages of incremental drift).

As we can see, in case of a sudden drift, a size of MLP hidden layer influences a learning curve in a similar way as a size of data chunk, but here the fewer number of artificial neurons decreases the learning abilities. On the other hand, more smooth drifts cause a lower drop of accuracy for, which means that in a fully incremental drift, for a structure of 50 and 500 neurons, we no longer observe a negative reaction to the occurrence of a drift. Natural mechanism of neural network forgetting allows (for a sufficiently large structure) to adapt smoothly to the changing concept.

In both previous examples, we observe that the change in learning parameters (chunk size) or network size (the number of neurons in the hidden layer) slows the rate of learning. In each case, the learning curves tend to go up in different dynamics. Fig. 4 presents that thanks to a sufficiently large data stream, various network structures achieve maximum discriminatory power and then encounter an sudden drift.

As expected, the return to a full accuracy (restoration time) of classification takes more time for smaller structures. Interesting, however, is the difference in the accuracy achieved in the first moments after the concept drift. Enlarging the structure leads to a greater decrease from the maximum accuracy when the drift occurs. It can therefore be concluded that the more complex structures, despite the greater ability to rebuild knowledge, are also more severely affected by the effects of sudden drifts. That is why for a practical task, it is necessary to find such a structure of MLP which will be a kind of trade-off between the restoration time and the classification accuracy drop after a sudden concept drift appearance.

4. Experiments

4.1. Goals

The main goal of conducted experimental evaluation was to verify the impact of *given budget* and *threshold* values on overall classification accuracy, measured for each processed chunk, with the aim to calibrate parameters in a way that (a) reduces usage of labels (b) without classification accuracy deterioration.

4.2. Setup

4.2.1. Software environment

Most of the research devoted to *data streams* is currently conducted using the MOA environment [41], implemented in the *Java* programming language. It includes a collection of base classifiers, necessary experimental methods, measurement tools and, above all, data stream generators. Knowing a growing tendency to employ *scikit-learn* library [42], it was decided to use it in implementation of the method described in following paper. It is not

[1] https://github.com/w4k2/active_learning.

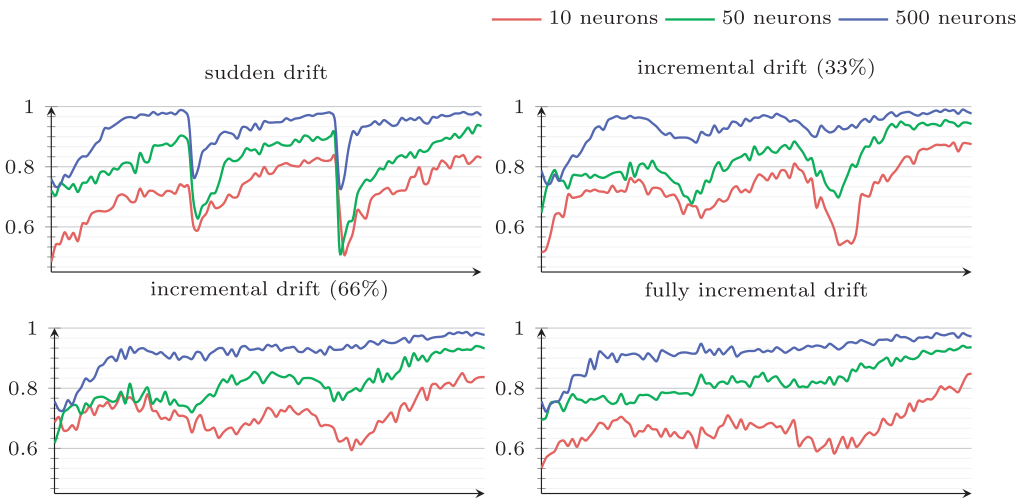


Fig. 3. Reaction of different artificial neural network structures to different types of drift.

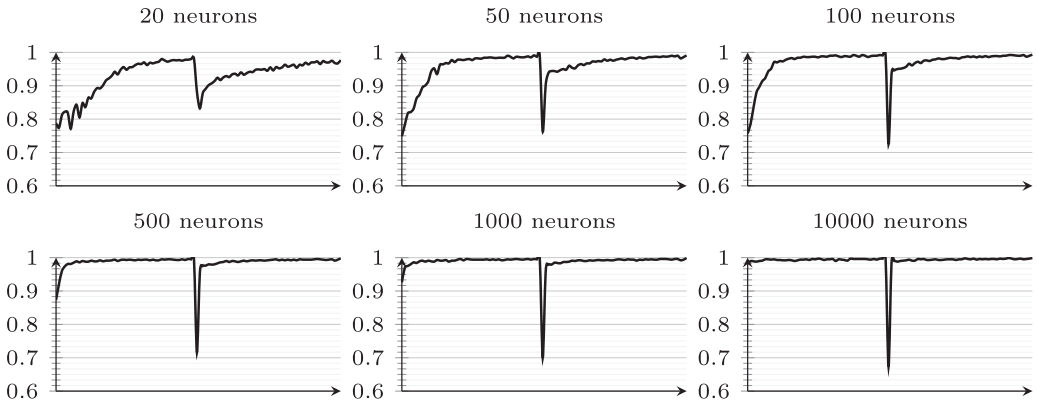


Fig. 4. Reaction of different artificial neural network structures to a sudden drift.

widely used with data streams, however, the potentiality of such processing was confirmed by a short article with a *proof-of-concept* available in the on-line *scikit-learn* documentation.[2]

It was necessary to create a new experimental flow for processing streams, enhanced with a class used to control a learning process. Current state of *scikit-learn* library allows to process data streams using classifiers having implemented the *partial fit* method, nevertheless as we mentioned above, we would like to employ inbuilt forgetting mechanism used by neural networks, therefore we used the implementation of MLP – Multi-layer Perceptron classifier, optimizing the log-loss function using LBFGS (*Limited memory Broyden-Fletcher-Goldfarb-Shanno*) algorithm [43].

The implementation of the active approach described in this paper as well as the workflow of learning from data streams using the *scikit-learn* library are part of the *stream-learn* module being developed by our research team.[3] Full code of experiments and presented examples, together with extended research results, can be found in the article repository.[4]

4.3. Benchmark data streams

The pool of analyzed data consist of 12 streams:

- Three real streams:
  - *covtypeNorm* dataset [44] includes the observations which may be used to classify the cover type of a forest on the basis of cartographic variables. It contains 54 attributes and 7 class labels. The appearance of the concept drift is a result of the changes in geographical condition.
  - *clecNormNew* (electricity) dataset [45] includes the data which may be used to predict the rise or fall of the electricity price in New South Wales, Australia. It contains 6 attributes and 2 classes. Concept drift is caused by the changes in consumption habits, events, and seasons.
  - *poker-lsn* (poker hand) dataset [44] includes the data which may be used to poker hands. It contains 10 attributes and 10 classes. Concept drift is caused by card changing at hand.
- Nine computer generated streams:
  - Two streams with concept drift (*RBFBlips, RBFGradualRecurring*) and one without it (*RBFNoDrift*) generated with *Radial Basis Function*,
  - A stream with a sudden drift (*LED*) and without it (*LEDNoDrift*) generated with LED generator,
  - Two streams with sudden drifts with different dynamics (*SEASudden, SEASuddenFaster*) generated with *Streaming Ensemble Algorithm* [46],
  - Two streams with gradual drifts with different dynamics (*HyperplaneFaster, HyperplaneSlow*) made by *Hyperplane* generator.

[2] http://scikit-learn.org/stable/modules/scaling_strategies.html.
[3] https://github.com/w4k2/stream-learn.
[4] https://github.com/w4k2/active_learning.

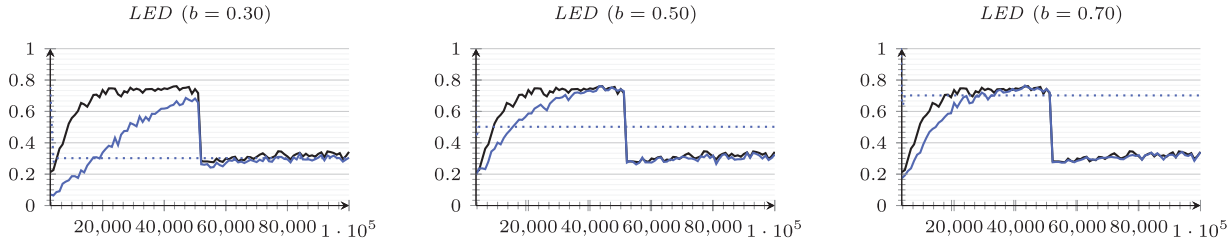


Fig. 5. Budget and learning curve (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.).

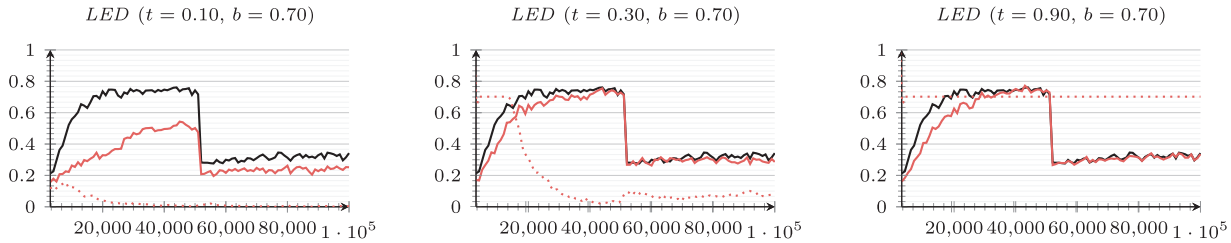


Fig. 6. Threshold and learning curve.

All the synthetic streams were generated by the MOA software.

4.3.1. Error evaluation

Every classifier uses a recent portion of data to train, but its evaluation (i.e., error estimation) is done on the following (unseen) data chunk. This type of performance evaluation is known as *test and train* or *block evaluation method* [41].

All the experiments were conducted with MLP classifier with 100 neurons in hidden layer, using chunks with size of 500 samples, evaluating the learning procedure after each percent of processed patterns.

4.4. Results

Experiments were conducted with 12 streams with three strategies:

- Measuring accuracy of incrementally training a model with all samples available in data stream.
- Measuring accuracy of incrementally trained model with randomized chunk subset, according to a given budget with five values in range 10—90%.
- Measuring accuracy of incrementally trained model with active learning approach described in Section 2 with five values in range 10—90% for both given budget and threshold.

Because conducted experiments produced many learning curves, only the part of them is included in the paper, but the detailed experimental results may be found in the article repository.[5]

Fig. 5 shows, with a blue line, a learning curve for three different budget values (30%, 50%, and 70%), while the black line is a curve for the classifier trained on the basis of all labeled examples (budget = 100%).

Decreasing a given budget leads, similarly to increasing the chunk or reducing a structure, to reduce the dynamics of learning. Therefore, it can not be a sufficient practice to reduce the cost of labeling. To maintain the accuracy of learning on a smaller training set, it is necessary to select the appropriately reduced set of samples, which we try to achieve by the proposed active learning method.

[5] https://github.com/w4k2/active_learning.

Table 1
A summary of the results obtained for the best combinations.

Data stream	t	b	Accuracies		difference	usage
			full	active		
RBFGradualRecurring	.9	9	0.677	0.653	−0.025	88.8%
RBFBlips	.9	9	0.773	0.725	−0.048	79.2%
HyperplaneFaster	.5	9	0.834	0.826	−0.008	52.0%
LEDNoDrift	.5	9	0.491	0.477	−0.013	42.8%
HyperplaneSlow	.1	7	0.837	0.868	0.030	39.9%
LED	.5	9	0.490	0.482	−0.008	39.5%
SEASuddenFaster	.3	7	0.833	0.828	−0.005	39.2%
RBFNoDrift	.1	9	0.807	0.817	0.010	34.5%
covtypeNorm	.9	9	0.734	0.719	−0.015	29.2%
SEASudden	.1	3	0.829	0.842	0.013	22.4%
poker-lsn	.9	1	0.563	0.558	−0.005	10.3%
elecNormNew	.1	9	0.623	0.577	−0.046	2.3%

For the same stream, Fig. 6 shows, with a continuous red line, a learning curve for three different threshold values (.1,.3, and.9) with a fixed budget of 70%. The black line is a curve for learning with all samples labeled, while the red dotted line is the percentage of samples used to train the model.

As we can see, too restrictive value of the threshold ($t = .1$) reduces the dynamics of the learning curve similarly to a low budget. The case of overly lax threshold ($t = .9$) allows to obtain the correct learning curve, but it does not reduce the samples necessary for the labeling, so it is no different from using only the budget parameter. However, proper calibration of the threshold ($t = .3$) allows to obtain the optimal learning curve, gradually reducing the need for sample labeling while obtaining knowledge by the classifier.

Fig. 7 shows the learning curves for selected, best parameters of a given budget and threshold for all tested data streams (continuous red lines). The black line is a curve for learning with all samples labeled. The red dotted line is the percentage of samples which were labeled during the model training.

Table 1 presents a summary of the results obtained for the best combinations. It contains, for each data set, selected parameters, average accuracy for the model learned on the full and reduced stream, the difference between the accuracies and the percentage use of the stream.

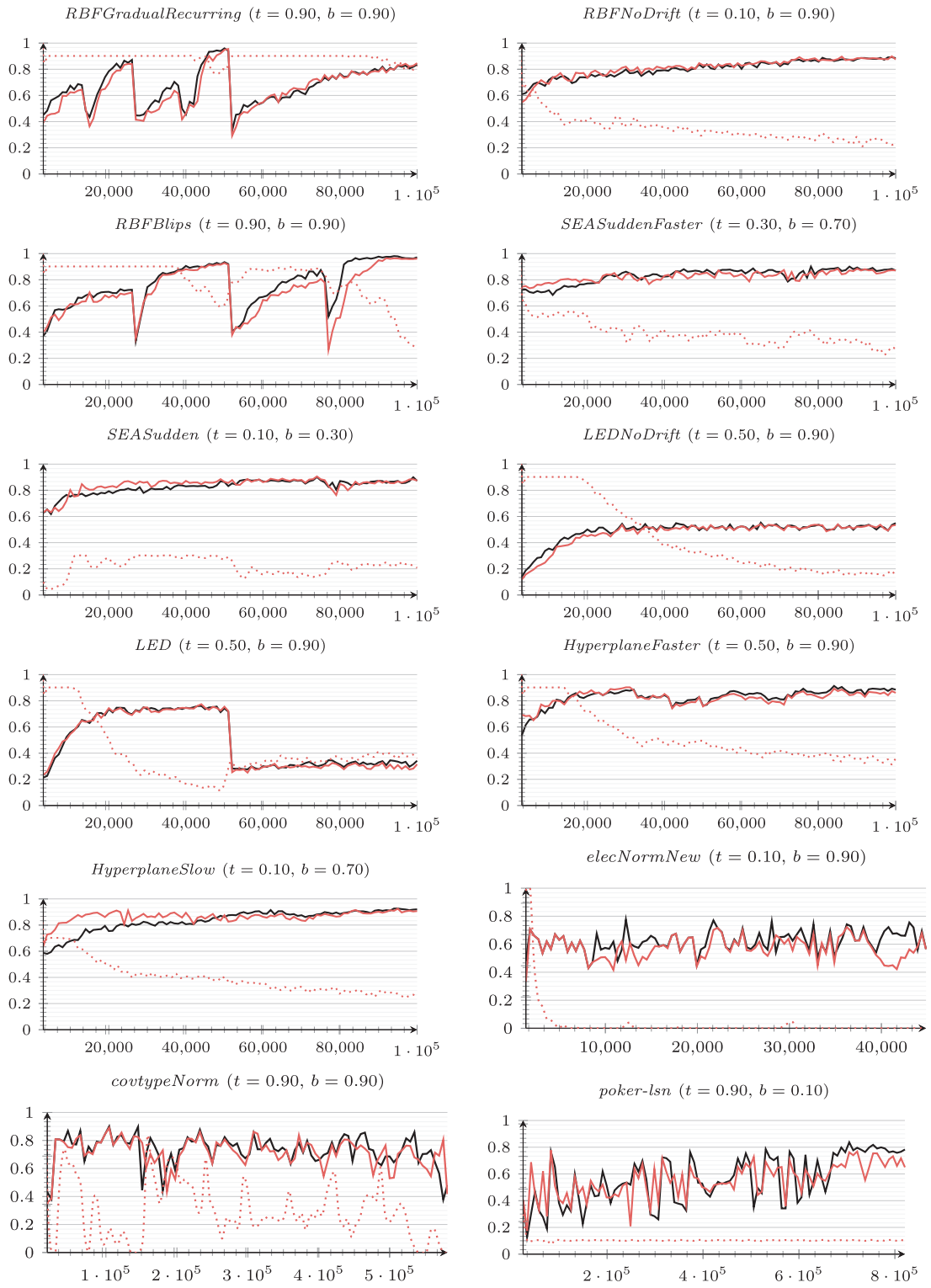


Fig. 7. Learning curves and label usages for selected, best parameters of a given budget and threshold for all tested data streams.

4.5. Analysis of the results

Most of the observations discussed around the Figs. 5 and 6 are confirmed by curves available in the Fig. 7. We need to reject the *elecNormNew* dataset, where even training with the fully labeled dataset led to a model with results similar to a random classifier. It is worth mentioning that Zliobaite [47] observes the problem of testing data stream classifiers on autocorrelated data and that getting high accuracy, especially on the *elecNormNew* dataset does not necessarily mean that the adaptation mechanism work well.

In every dataset, after this exclusion, a tendency to consequently reducing the chunk usage is observed. In the case streams with many concept drifts (*RBFGradualRecurring* and *RBFBlips*) the reduction of label use is relatively small, but it is caused by a slow accumulation of knowledge of the selected classifier, where 100 neurons in the hidden layer were not sufficient to achieve the

maximum accuracy of classification before the next drift. We may also observe the intensive growth of the label demand, when a concept drift appears.

For data streams generated by *Streaming Ensemble Algorithm*, the reduction of the accuracy is not observed. In the case of two sets of data (*SEASudden* and *HyperplaneSlow*), the appropriate combination of a given budget and threshold allows not only to preserve the dynamics of learning, but also to accelerate it.

Even in the case of data streams where the selected neural network structure was insufficient to achieve full discriminative power before occurrence of the next drift, it was possible to reduce the number of samples necessary for labeling by 10–20%. In the case of the remaining sets, a reduction of 50–90% was achieved, without a negative impact on the average quality of the classification.

4.6. Lessons learned

Let us summarize the research findings and observations that could be drawn from the experimental analysis:

- Active learning approach may lead to a solution where a trained model keeps the classification accuracy of full-stream learning, with a slight reduction of used labels.
- There is no rule of correct setting the parameters related to a given budget and threshold, so it should be calibrated for a particular task.
- Active learning approach is able to detect a concept drift and react on it without negative impact on classification accuracy.
- Chunk usage decreases slowly in time, according to stabilization of a model on current concept.
- Acquiring the necessary knowledge by a model reduces the need for new samples.
- A *knowledge saturation* (degree of obtaining the maximum discriminative power) of a model can be successfully measured by RFSD.
- Chunk reduction has a positive effect by increasing the learning frequency but extends the learning time itself.
- The size of a neural network which is responsible for the concept memorization should be set carefully, for a given decision task, because on the one hand the smaller structure of the network the faster is its reaction to the concept drift, what is especially important in the case of sudden drift. Such fast model adaptation protect against the huge classification accuracy drop. On the other hand appropriate increasing of neural network size allows the immunization of the model to accuracy drop when the slow, incremental concept drift goes ahead.
- Limitation of training set by just a given budget with random subset of samples causes a proportional reduction in the learning dynamics.

5. Conclusions

The novel active learning strategy for neural network classifiers has been proposed, where to implement the forgetting mechanism, we employed the *catastrophic forgetting* phenomenon. The results of the experimental evaluation of the proposed algorithm carried out on the basis of diverse benchmark datasets and a detailed comparison with the semi-supervised and fully supervised strategies prove the usefulness of the proposed methods and show that it can better allocate the labeling budget.

In the near future we are going to:

- Develop the methods which will allow to control the speed of forgetting, especially adaptive forgetting is the desirable characteristic, i.e., possibility of changing the forgetting speed on the basis of concept drift appearance frequency and/or its severity.
- Evaluating the proposed algorithm behavior for more type of concept drift and maybe employ concept drift detector to establish the chunk size, threshold and budget dynamically.
- Applying the proposed approach to the classifier ensemble.

Acknowledgment

This work is supported by the Polish National Science Center under the Grant no. UMO-2015/19/B/ST6/01597 as well the statutory funds of the Department of Systems and Computer Networks, Faculty of Electronics, Wrocław University of Science and Technology.

References

[1] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, 2 ed., Wiley, New York, 2001.
[2] G. Widmer, M. Kubat, Learning in the presence of concept drift and hidden contexts, Mach. Learn. 23 (1) (1996) 69–101.
[3] A.A. Beyene, T. Welemariam, M. Persson, N. Lavesson, Improved concept drift handling in surgery prediction and other applications, Knowl. Inf. Syst. 44 (1) (2015) 177–196, doi:10.1007/s10115-014-0756-9.
[4] T. Lane, C.E. Brodley, Approaches to online learning and concept drift for user identification in computer security, in: R. Agrawal, P.E. Stolorz, G. Piatetsky-Shapiro (Eds.), Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98), New York City, New York, USA, AAAI Press, 1998, pp. 259–263.
[5] J.R. Méndez, F. Fdez-Riverola, E.L. Iglesias, F. Díaz, J.M. Corchado, Tracking Concept Drift at Feature Selection Stage in SpamHunting: An Anti-spam Instance-Based Reasoning System, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 504–518.
[6] M. Black, R. Hickey, Classification of customer call data in the presence of concept drift and noise, Soft-Ware Comput. Imperfect World (2002) 221–254.
[7] A.D. Pozzolo, G. Boracchi, O. Caelen, C. Alippi, G. Bontempi, Credit card fraud detection and concept-drift adaptation with delayed supervised information., in: Proceedings of the IJCNN, IEEE, 2015, pp. 1–8.
[8] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM Comput. Surveys (2013). In Press.
[9] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, 2004.
[10] P. Sobolewski, M. Woźniak, Concept drift detection and model selection with simulated recurrence and ensembles of statistical detectors, J. Univ. Comput. Sci. 19 (4) (2013) 462–483.
[11] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM Comput. Surv. 46 (4) (2014) 44:1–44:37.
[12] P. Domingos, G. Hulten, A general framework for mining massive data streams., J. Comput. Graph. Stat. 12 (2003) 945–949.
[13] N. Littlestone, Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm, Mach. Learn. 2 (4) (1988) 285–318, doi:10.1023/A:1022869011914.
[14] P. Domingos, G. Hulten, Mining high-speed data streams, in: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD '00, ACM, New York, NY, USA, 2000, pp. 71–80.
[15] G.A. Carpenter, S. Grossberg, D.B. Rosen, Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system, Neural Netw. 4 (6) (1991) 759–771, doi:10.1016/0893-6080(91)90056-B.
[16] J.C. Schlimmer, R.H. Granger Jr., Incremental learning from noisy data, Mach. Learn. 1 (3) (1986) 317–354.
[17] J. Kolter, M. Maloof, Dynamic weighted majority: a new ensemble method for tracking concept drift, in: Proceedings of the Third IEEE International Conference on Data Mining ICDM, 2003, pp. 123–130.
[18] A. Bouchachia, C. Vanaret, GT2FC: an online growing interval type-2 self-learning fuzzy classifier, IEEE Trans. Fuzzy Syst. 22 (4) (2014) 999–1018, doi:10.1109/TFUZZ.2013.2279554.
[19] B. Krawczyk, M. Woźniak, Incremental learning and forgetting in one-class classifiers for data streams, in: R. Burduk, K. Jackowski, M. Kurzynski, M. Wozniak, A. Zolnierek (Eds.), Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013, Advances in Intelligent Systems and Computing, 226, Springer International Publishing, 2013, pp. 319–328.
[20] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, R. Gavaldà, New ensemble methods for evolving data streams, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD '09, ACM, New York, NY, USA, 2009, pp. 139–148.
[21] I. Koychev, Gradual forgetting for adaptation to concept drift, in: Proceedings of the ECAI Workshop on Current Issues in Spatio-Temporal Reasoning, Berlin, Germany, 2000, pp. 101–106.
[22] R. Klinkenberg, Learning drifting concepts: example selection vs. example weighting, Intell. Data Anal. 8 (3) (2004) 281–300.
[23] M. Woźniak, A. Kasprzak, P. Cal, Application of combined classifiers to data stream classification, in: Proceedings of the 10th International Conference on Flexible Query Answering Systems FQAS 2013, in: LNCS, Springer-Verlag, Berlin, Heidelberg, 2013, p. in press.

[24] J.S. Vitter, Random sampling with a reservoir, ACM Trans. Math. Softw. 11 (1) (1985) 37–57, doi:10.1145/3147.3165.
[25] B. Kurlej, M. Woźniak, Impact of window size in active learning of evolving data streams, in: Proceedings of the 45th International Conference on Modelling and Simulation of Systems MOSIS, 2011, pp. 56–62.
[26] A. Bifet, R. Gavaldà, Learning from time-changing data with adaptive windowing, in: Proceedings of the Seventh SIAM International Conference on Data Mining, Minneapolis, Minnesota, USA, 2007, pp. 443–448.
[27] M.M. Lazarescu, S. Venkatesh, H.H. Bui, Using multiple windows to track concept drift, Intell. Data Anal. 8 (1) (2004) 29–59.
[28] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A.A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, R. Hadsell, Overcoming catastrophic forgetting in neural networks, Proc. Natl. Acad. Sci. 114 (13) (2017) 3521–3526, doi:10.1073/pnas.1611835114.
[29] D. Kumaran, D. Hassabis, J.L. McClelland, What Learning Systems do Intelligent Agents Need? Complementary Learning Systems Theory Updated, Trends Cogn. Sci. 20 (7) (2016) 512–534, doi:10.1016/j.tics.2016.05.004.
[30] R. Greiner, A.J. Grove, D. Roth, Learning cost-sensitive active classifiers, Artif. Intell. 139 (2) (2002) 137–174.
[31] I. Žliobaitė, A. Bifet, B. Pfahringer, G. Holmes, Active learning with drifting streaming data, IEEE Trans. Neural Netw. Learn. Syst. 25 (1) (2014) 27–39.
[32] B. Kurlej, M. Woźniak, Active learning approach to concept drift problem, Log. J. IGPL 20 (3) (2012) 550–559.
[33] B. Kurlej, M. Woźniak, Learning curve in concept drift while using active learning paradigm, in: A. Bouchachia (Ed.), Adaptive and Intelligent Systems, Lecture Notes in Computer Science, 6943, Springer Berlin Heidelberg, 2011, pp. 98–106.
[34] H.-L. Nguyen, W.-K. Ng, Y.-K. Woon, Concurrent Semi-supervised Learning with Active Learning of Data Streams, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 113–136.
[35] S. Mohamad, A. Bouchachia, M. Sayed-Mouchaweh, A bi-criteria active learning algorithm for dynamic data streams, IEEE Trans. Neural Netw. Learn. Syst. 29 (1) (2018) 74–86, doi:10.1109/TNNLS.2016.2614393.
[36] Ł. Korycki, B. Krawczyk, Combining Active Learning and Self-Labeling for Data Stream Mining, Springer International Publishing, Cham, pp. 481–490.
[37] D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, Mach. Learn. 6 (1) (1991) 37–66.
[38] C. Chow, On optimum error and reject trade-off, IEEE Trans. Inf. Theory 16 (1970) 41–46.
[39] G. Fumera, F. Roli, G. Giacinto, Multiple Reject Thresholds for Improving Classification Reliability, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 863–871.
[40] B. Krawczyk, L.L. Minku, J. Gama, J. Stefanowski, M. Woźniak, Ensemble learning for data stream analysis: A survey, Inf. Fus. 37 (2017) 132–156.
[41] A. Bifet, G. Holmes, R. Kirkby, B. Pfahringer, Moa: Massive online analysis, J. Mach. Learn. Res. 11 (2010) 1601–1604.
[42] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[43] A. Mokhtari, A. Ribeiro, Global convergence of online limited memory BFGS, J. Mach. Learn. Res. 16 (2015) 3151–3181.
[44] A. Frank, A. Asuncion, in: UCI machine learning repository, 2010. http://archive.ics.uci.edu/ml.
[45] M. Harries, SPLICE-2 Comparative Evaluation: Electricity Pricing, Technical Report, The University of South Wales, 1999.
[46] W.N. Street, Y. Kim, A streaming ensemble algorithm (SEA) for large-scale classification, in: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD '01, ACM, New York, NY, USA, 2001, pp. 377–382, doi:10.1145/502512.502568.
[47] I. Zliobaite, How good is the electricity benchmark for evaluating concept drift adaptation, CoRR abs/1301.3524 (2013).

Paweł Ksieniewicz is an assistant professor of computer science at the Department of Systems and Computer Networks, Wroclaw University of Science and Technology, Poland. He received M.Sc. and Ph.D. degrees in computer science from the Wroclaw University of Science and Technology in 2013 and 2017, respectively. His research focuses on pattern classification and machine learning methods and computer vision.

Michał Woźniak is a professor of computer science at the Department of Systems and Computer Networks, Wroclaw University of Science and Technology, Poland. He received M.Sc. degree in biomedical engineering from the Wroclaw University of Technology in 1992, and Ph.D. and D.Sc. (habilitation) degrees in computer science in 1996 and 2007, respectively, from the same university. In 2015 he was nominated as the professor by President of Poland. His research focuses on compound classification methods, hybrid artificial intelligence and medical informatics. Prof. Wozniak has published over 260 papers and three books. His recent one Hybrid classfiers: Method of Data, Knowledge, and Data Hybridization was published by Springer in 2014. He has been involved in research projects related to the abovementioned topics and has been a consultant of several commercial projects for well-known Polish companies and public administration. Prof. Wozniak is a senior member of the IEEE.

Bogusław Cyganek received his M.Sc. degree in electronics in 1993, and then M.Sc. in computer science in 1996, from the AGH University of Science and Technology, Krakow, Poland. He obtained his Ph.D. degree cum laude in 2001 and D.Sc. degree in 2011. During the recent years, Dr. Boguslaw Cyganek cooperated with many scientific and industrial partners such as Glasgow University Scotland UK, DLR Germany, and Surrey University UK. Currently he is a researcher and lecturer at the Department of Electronics, AGH University of Science and Technology, Poland. His research interests include computer vision, pattern recognition, as well as development of programmable devices and embedded systems. He is an author or a co-author of over a hundred of conference and journal papers, as well as books with the latest Object Detection and Recognition in Digital Images: Theory and Practice published by Wiley in 2013. Dr. Cyganek is a member of the IEEE, IAPR and SPIE.

Andrzej Kasprzak received his Ph.D. and D.Sc. (habilitation) degrees in computer science from Wroclaw University of Science and Technology, Poland, in 1979 and 1989, respectively. In 2001 he was nominated as the professor by President of Poland. His research focuses on designing computer network strucures and intelligent computing. Currently, he serves as the chair of the Department of Systems and Computer Networks, Wroclaw University of Technology. He has published more than 200 scientific papers in international conferences and journals.

Krzysztof Walkowiak received his Ph.D. and D.Sc. (habilitation) degrees in computer science from Wroclaw University of Science and Technology, Poland, in 2000 and 2008, respectively. In 2017 he was nominated as the professor by President of Poland. Currently, he serves as an associate professor at the Department of Systems and Computer Networks, Wroclaw University of Technology. He received the Best Paper Award from the International Workshop on Design of Reliable Communication Networks 2009. He has published more than 160 scientific papers in international conferences and journals.

[C8]
Pawel Ksieniewicz. “Undersampled Majority Class Ensemble for highly imbalanced binary classification”. W: *Proceedings of the Second International Workshop on Learning with Imbalanced Domains: Theory and Applications*. Red. Luís Torgo i in. T. 94. Proceedings of Machine Learning Research. PMLR, paź. 2018, s. 82–94. URL: https://proceedings.mlr.press/v94/ksieniewicz18a.html

Proceedings of Machine Learning Research 94:82–94, 2018 LIDTA 2018

Undersampled Majority Class Ensemble for highly imbalanced binary classification

Paweł Ksieniewicz PAWEL.KSIENIEWICZ@PWR.EDU.PL
Department of Systems and Computer Networks
Faculty of Electronics
Wrocław University of Science and Technology

Editors: Luís Torgo, Stan Matwin, Nathalie Japkowicz, Bartosz Krawczyk, Nuno Moniz, and Paula Branco

Abstract

Following work tries to utilize an ensemble approach to solve a problem of highly imbalanced data classification. Paper contains a proposition of UMCE – a multiple classifier system, based on *k-fold division* of the *majority class* to create a pool of classifiers breaking one *imbalanced problem* into many balanced ones while ensuring the presence of all available samples in the training procedure. Algorithm, with five proposed fusers and a pruning method based on the statistical dependencies of the classifiers response on the testing set, was evaluated on the basis of the computer experiments carried out on the benchmark datasets and two different base classifiers.

Keywords: classification, classifier ensemble, undersampling, imbalanced data

1. Introduction

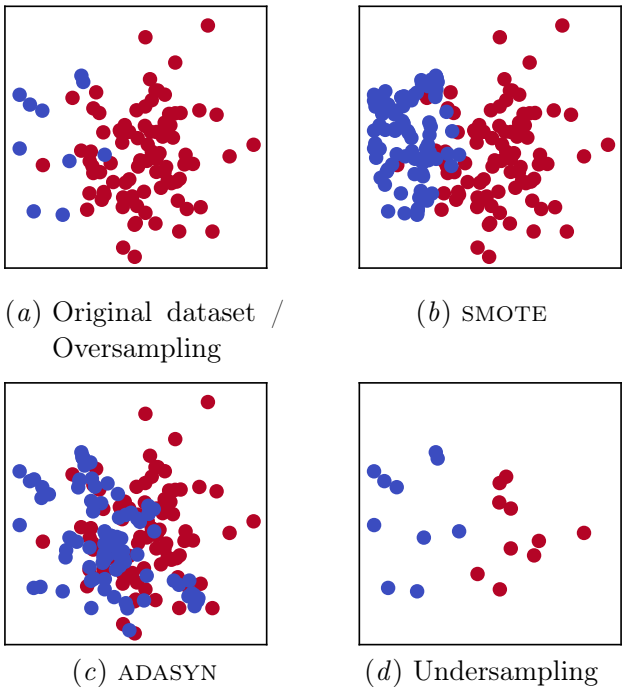
Most of existing classification models benefit from the assumption that there are no significant disparities between the classes of the considered problem. Nevertheless, in the real world, there are many situations in which the number of objects from one of the classes (called the *majority class*) significantly exceeds the number of objects of the remaining classes (*minority classes*), which often leads to decisions biased towards the *majority class*. However, when considering cases such as spam filtering, medical tests or fraud detection, we may come to the conclusion that the cost of making an incorrect decision against a minority class is much greater than in other cases. The above-mentioned problem is called in the literature the *imbalanced data classification* (Wang et al., 2017; Sun et al., 2009).

Following work focuses on the binary classification of the highly imbalanced problems, with an IR (*imbalanced ratio*) greater than 9, which is an important issue not only in the context of the construction of appropriate models, but even in a proper quality measurement (Elazmeh et al., 2006). One of the important problems is also the fact that the number of patterns in the *minority class* may be so small that it will not allow to achieve the appropriate discriminatory power of the model, which may lead to its *overfitting* (Chen and Wasikowski, 2008). Most of these problems are the subject of extensive research (Bunkhumpornpat et al., 2009; Chawla et al., 2002).

One of the possible approaches to solve such problems are *inbuild mechanisms*, trying to adapt existing classification models to balance the accuracy between classes. Popular solution of this kind is the learning approach without counter-examples, using *one-class*

classification (Japkowicz et al., 1995; Krawczyk et al., 2014), where the aim is to get to know the decision boundaries within *minority classes*. The solution may also be the *cost sensitive solutions*, assuming the asymmetric *loss function* (Lopez et al., 2012; He and Garcia, 2009).

Another approach, more connected with the scope of following paper, is the group of *data preprocessing methods*, which focuses on reducing the number of *majority class* objects (*undersampling*) or generating patterns of *minority class* (*oversampling*) to balance a dataset. Graphical overview of methods from this group is presented in Figure 1.



(*a*) Original dataset / Oversampling

(*b*) SMOTE

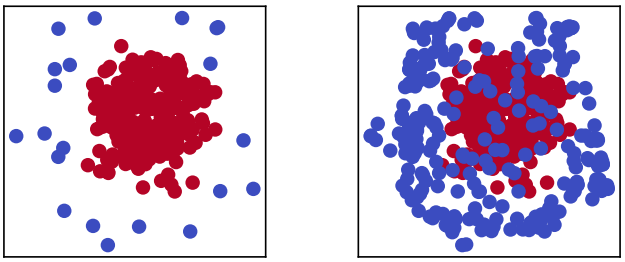
(*c*) ADASYN

(*d*) Undersampling

Figure 1: Examples of data preprocessing methods.

These algorithms are addressing the task of balancing the number of objects within the problem classes. In the case of basic *oversampling*, new objects are created as random copies of those already existing in the training set[1]. Currently, the most common kind of *oversampling* is SMOTE (Chawla et al., 2011), shown in Figure 1(*b*), creating new, synthetic objects based on k averaged examples nearest to a random points from the space occupied by a minority class. An active version of SMOTE is the ADASYN algorithm (He et al., 2008), shown in Figure 1(*c*), which takes into account the difficulty of synthetic samples. This approach allows to solve the problem of repeating samples in the training set, but can also lead to *overfitting*, which is presented in Figure 2.

1. Since the characteristics of the new patterns will be identical to those already present in the dataset, we can consider Figure 1(*a*), an illustration of the original dataset, also as the presentation of pattern distribution after oversampling.



(*a*) Original dataset (*b*) SMOTE

Figure 2: Example of wrong SMOTE oversampling.

In the case of *undersampling*, shown in Figure 1(*d*), in which we draw as many objects from the majority class as are present in the minority class, there is no risk of erroneous mixing of the classes distribution.

The last group of methods to be mentioned here are *hybrid approaches*, combining *over-* and *undersampling* algorithms with *ensemble classifiers* (Galar et al., 2012). The *Bagging* and *Boosting* variants, such as *AdaBoost.NC* (Wang et al., 2010) or *SMOTEBoost* (Chawla et al., 2003), have become particularly popular in this area.

The main contributions of this work are:

- a method of establishing a homogenous *ensemble* using a *k-fold undersampling* of *majority class*,
- proposition of five *fusers* to generate *ensemble* decision,
- a *pruning* method adjusting the decision rule to the *testing set*,
- implementation and experimental evaluation of proposed method.

2. *Undersampled Majority Class Ensemble*

2.1. Establishing ensemble

Complex oversampling methods, such as SMOTE or ADASYN, despite the large possibilities in most of the problems in imbalanced domain, are not applicable to extreme situations where the *minority class* is represented by only a few samples, which makes it impossible to designate the nearest neighbors to create a new synthetic object. This could lead to the use of *undersampling* in such problems, but it is characterized, due to high randomness, by a strong instability in a situation of high IR (*imbalance ratio*), which does not allow for the development of a reliable solution.

A popular answer to the above-mentioned problem are the *ensemble* methods of *Bagging* or *Boosting*, characterized by random sampling with replacement of the training set, breaking a large problem, into a set of smaller ones. This work proposes a basic method, which also breaks the imbalanced task, but with ensuring the use of all the patterns available in the data set, but without a risk of overlapping. Its description may be found in Algorithm 1.

Algorithm 1: Training classifier ensemble from multiple balanced training datasets separated from one imbalanced dataset of binary problem
Given a dataset DS:

1. Divide DS into subsets of minority- $MinC$ and majority-class $MajC$
2. Calculate imbalanced ratio IR as the proportion of the number of patterns in $MinC$ and $MajC$
3. Establish k by rounding IR to nearest integer
4. Perform a *shuffled k-fold division* of $MajC$ to produce a set of subsets $MajC_1, MajC_2, \ldots, MajC_k$
5. For every i in range to k
    6. Join $MajC_i$ with $MinC$ to prepare a training set TS_i,
    7. Train classifier Ψ_i on TS_i and add it into ensemble

After dividing the dataset with imbalanced binary problem into separated minority ($MinC$) and majority class ($MajC$), we are calculating the IR (*imbalanced ratio*) between given classes. Rounding IR to the nearest integer value k allows us to find the optimal division coefficient of the majority class samples in the context of maximizing the balance between the $MinC$ and any $MajC_i$ subsets while ensuring that all $MajC$ patterns are used in learning process with no overlapping between the individual $MajC_i$'s. Each of k classifiers Ψ_i is trained on union of $MinC$ and $MajC_i$ sets.

Extending pool with oversampling As an extension of the method of classifier ensemble construction, it is also proposed to expand its pool by a model learned on an additional data set, which is a full set of data subjected to *oversampling*. It is worth testing if the knowledge gained from this method may be a valuable contribution to the ensemble decision. Due to impossibility of using SMOTE or ADASYN for oversampling the minority class with only few instances, only its basic variant will be employed.

2.2. Fuser design

In addition to ensuring the diversity of the classifiers pool, which we achieve by a homogenous committee built on disjoint subsets of the majority class supplemented by minority patterns, the key aspect of the hybrid classification system is the appropriate design of its *fuser* – the element responsible for making decisions based on the answers of the base classifiers.

There are two groups of solutions here. The first are based on component *decisions* of the committee, most often employing the *majority voting* to produce a final decision. The decision rules proposed in this work are, however, part of the second group, where the *fuser* is carried out by *averaging* (or *accumulating*) the *support vectors* received from the members of a pool. It should be remembered that in such methods, it is necessary to use a *probabilistic classification model*, which also requires *quantitative* and not *qualitative*

data, so we need to reject classification algorithms such as *Support Vector Machines*, whose probabilistic interpretation becomes reliable only in cases of large training sets.

Five accumulative fusers were proposed to analyze:

1. **REG** — regular accumulation of support.

   A basic method without weighing the members of a committee.

2. **WEI** — accumulation weighted after members of a committee.

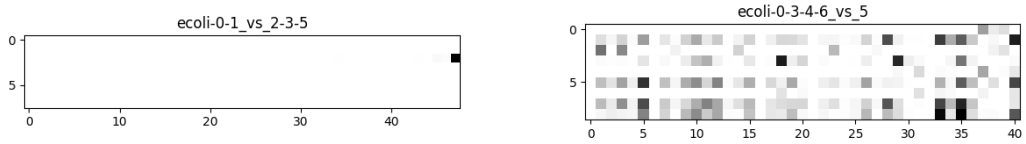
   The weight of the classifier in the pool is its quality achieved for the training set. We can not use here the measure of *accuracy*, which does not fit with the task of the imbalanced classification, so a *balanced accuracy* was chosen (Brodersen et al., 2010).

3. **NOR** — same as **WEI**, but with normalization of weights,

   To reward classifiers with a higher *discriminative power*, weights are subjected to normalization by a *MinMaxScaler*.

4. **CON** — accumulation weighted by tested patterns.

   In order to reward classifiers with greater *"certainty"* for given object, the decision for each pattern is weighted by the absolute difference between class support, for the needs of research called the *contrast*. Individual classifiers in the pool do not have to be better or worse for each of the tested patterns. This is illustrated in Figure 3, where we can see two cases of ensembles. There are tested patterns on the X axis and classifiers in the pool on the Y axis. A white square means the *contrast* of 1, and therefore a *sure* decision, and the black square the *contrast* of 0, which describes the pattern that is exactly on the decision boundary.



(*a*) Example of a *"sure"* ensemble

(*b*) Example of *"unsure"* ensemble

Figure 3: Illustration of the *contrast* in committees built on two different datasets.

5. **NCI** — accumulation weighted by a product of normalized weights and a *contrast*.

The proposed method of constructing the committee makes its size directly dependent on the IR, which, given the highly unbalanced data (for example with IR greater than 40), leads to the construction of an extensive hybrid model. Therefore, the method of prunning it to a smaller size was also considered.

2.3. Ensemble pruning

Typical methods of *ensemble pruning* follow the phase of training the committee, for example, by eliminating the classifiers that achieve the lowest quality on the *training* or separated *validation set*. This paper proposes a method of *response pruning* based on the assumption that during the testing phase we analyze not just a single test pattern, but the entire *testing set*.

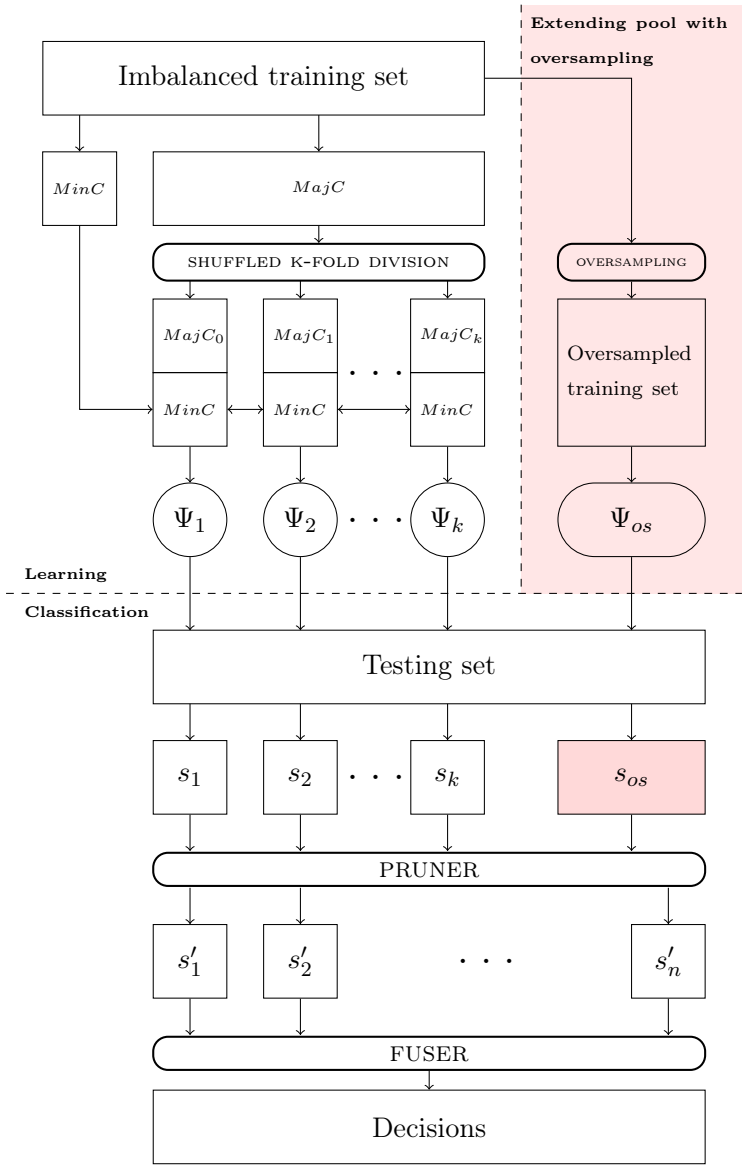


Figure 4: Diagram of *Undersampled Majority Class Ensemble* structure

Ensemble, receiving a *testing set*, generates *support vectors* (s_i) for each classified object, so, with a binary problem, we can treat received support for one of the problem classes as values from the *random variables* to analyze their mutual statistical dependence.

In the proposed method, using the signed-rank test, we are *clustering* the pool of k (or $k+1$ on the *oversampling* variation of a method) classifiers to n groups (where $n \leq k$), to average the support and weight classes within groups to create a new set of supports from s'_1 to s'_n, passed later on to *fuser*. It is important to denote, that in the considered case of pruning, we ignore the possible situation in which the answer Ψ_1 is dependent on Ψ_2, the answer Ψ_2 is dependent on Ψ_3, but Ψ_1 is not dependent on Ψ_3. This is an interesting issue that will be addressed in future research, but to clarify the proposal, a simplified approach has been used.

The scheme of the full decision model of the proposed method is shown in Figure 4.

3. Experiment design

For the experimental evaluation of the proposed method, a collection of datasets made available with KEEL (Alcalá-Fdez et al., 2011) was used, focusing on a section containing highly unbalanced data, with IR greater than 9 (Fernández et al., 2009). From among the available datasets, 40 were selected, presenting only binary problems with quantitative attributes. A review of selected datasets, including information on their number of features, the number of patterns in each class and the imbalance ratio is presented in Table 1.

DS	Features	Samples			IR
		ALL	MAJ	MIN	
ecoli-0-1-3-7-vs-2-6	7	281	274	7	39.14
ecoli4	7	336	316	20	15.80
glass-0-1-6-vs-2	9	192	175	17	10.29
glass-0-1-6-vs-5	9	184	175	9	19.44
glass2	9	214	197	17	11.59
glass4	9	214	201	13	15.46
glass5	9	214	205	9	22.78
page-blocks-1-3-vs-4	10	472	444	28	15.86
shuttle-c0-vs-c4	9	1829	1706	123	13.87
shuttle-c2-vs-c4	9	129	123	6	20.50
vowel0	13	988	898	90	9.98
yeast-0-5-6-7-9-vs-4	8	528	477	51	9.35
yeast-1-2-8-9-vs-7	8	947	917	30	30.57
yeast-1-4-5-8-vs-7	8	693	663	30	22.10
yeast-1-vs-7	7	459	429	30	14.30
yeast-2-vs-4	8	514	463	51	9.08
yeast-2-vs-8	8	482	462	20	23.10
yeast4	8	1484	1433	51	28.10
yeast5	8	1484	1440	44	32.73
yeast6	8	1484	1449	35	41.40
ecoli-0-1-4-6-vs-5	6	280	260	20	13.00
ecoli-0-1-4-7-vs-2-3-5-6	7	336	307	29	10.59
ecoli-0-1-4-7-vs-5-6	6	332	307	25	12.28
ecoli-0-1-vs-2-3-5	7	244	220	24	9.17
ecoli-0-1-vs-5	6	240	220	20	11.00
ecoli-0-2-3-4-vs-5	7	202	182	20	9.10
ecoli-0-2-6-7-vs-3-5	7	224	202	22	9.18
ecoli-0-3-4-6-vs-5	7	205	185	20	9.25
ecoli-0-3-4-7-vs-5-6	7	257	232	25	9.28
ecoli-0-3-4-vs-5	7	200	180	20	9.00
ecoli-0-4-6-vs-5	6	203	183	20	9.15
ecoli-0-6-7-vs-3-5	7	222	200	22	9.09
ecoli-0-6-7-vs-5	6	220	200	20	10.00
glass-0-1-4-6-vs-2	9	205	188	17	11.06
glass-0-1-5-vs-2	9	172	155	17	9.12
glass-0-4-vs-5	9	92	83	9	9.22
glass-0-6-vs-5	9	108	99	9	11.00
yeast-0-2-5-6-vs-3-7-8-9	8	1004	905	99	9.14
yeast-0-2-5-7-9-vs-3-6-8	8	1004	905	99	9.14
yeast-0-3-5-9-vs-7-8	8	506	456	50	9.12

Table 1: Summary of imbalanced datasets chosen for evaluation

As may be observed in the summary, the experiments are based on datasets with relatively small spatiality (up to 13 dimensions), with imbalance ratio from 9 to even 40. The

datasets provided by KEEL, to ensure easy comparison between results presented in various research, are already pre-divided into five parts, which forces the use of *k-fold cross-validation* with 5 folds in experiments (Alpaydin, 2009).

In the task of imbalanced data classification, due to its strong bias towards majority class, the *accuracy* measure is not a proper tool. For a reliable result, a measure of *balanced accuracy* is given as test results.

Both the implementation of the proposed method and the experimental environment have been constructed using the *scikit-learn* library (Pedregosa et al., 2011) in version *0.20.dev0*[2]. Among the available classification models, the MLP (*Multilayer Perceptron*) and SVC (*Support Vector Machine*) were rejected. First one was not able to build a correct model due to the lack of convergence on the small datasets (minority class of data chosen for experiments is often represented by only two patterns in cross-validated folds) and second one, whose probabilistic interpretation is measurable only with sufficiently large data sets, did not allow credible construction of a fuser. As base classifiers, the following algorithms were used:

- *Gaussian Naive Bayes* (GNB) (Chan et al., 1982),
- *Decision Tree Classifier* (DTC) — with *Gini* criterion (Loh, 2011).

To provide a comparative result for the method presented in the following paper, each base classifier was also tested for (*i*) the raw, imbalanced dataset and its (*ii*) under- and (*iii*) oversampled versions. Undersampling, due to high instability of results, was repeated five times on each fold. Used statistical analysis tool was a paired dependency between the classifier, which achieved the highest result and each of the others, calculated using the signed-rank *Wilcoxon* test (Wilcoxon, 1945).

The full implementation of the proposed method, content of the following paper and the script allowing to reconstruct the presented research may be found in the *git* repository[3].

4. Experimental evaluation

The results of the conducted research, for individual base classifiers, are presented in Tables 2 and 3. They were divided to present in individual sections a *balanced accuracy* achieved by particular variations of the method proposed in the following paper. In the first division stage, we show the impact of inclusion of the classifier built on the *oversampled* dataset, in the second, the use of the proposed *pruning* method, and in the third – employed *fuser*. It gave the number of 20 algorithm variations.

The presented results were supplemented by a balanced accuracy achieved by the classifier built on a full, *imbalanced dataset* (**Full**), a set after *undersampling* (**US**) and an *oversampling* (**OS**). The table cells marked in green indicate the best result for a dataset or the result statistically dependent on it, calculated in accordance with previously described assumptions of the experiments.

As we can see in Table 2, which presents the quality of classification using the GNB algorithm, there were only two datasets, where the lone best solution was to train the

2. At the time of conducting research, only the development version of the package already has the implementation of *balanced accuracy* measure.
3. https://github.com/w4k2/umce

Without oversampled set										With oversampled set													
Without pruning					With pruning					Without pruning					With pruning								
REG	WEI	CON	NOR	NC	REG	WEI	CON	NOR	NC	REG	WEI	CON	NOR	NC	REG	WEI	CON	NOR	NC	OS	US	Full	Dataset
.843	.843	.843	.834	.839	.845	.845	.845	.834	.837	.830	.828	.828	.817	.817	.845	.845	.845	.834	.834	.807	.835	.825	*ecoli-0-1-3-7-vs-2-6*
.685	.735	.785	.807	.807	.717	.767	.792	.875	.896	.779	.823	.841	.910	.896	.740	.792	.842	.898	.918	.860	.765	.878	*ecoli4*
.613	.613	.616	.582	.582	.580	.580	.574	.616	.588	.585	.582	.585	.580	.580	.577	.577	.574	.613	.585	.577	.574	.580	*glass-0-1-6-vs-2*
.989	.989	.989	.989	.989	.989	.989	.989	.989	.989	.989	.989	.989	.989	.989	.989	.989	.989	.989	.989	.941	.967	.941	*glass-0-1-6-vs-5*
.641	.641	.644	.641	.641	.641	.641	.641	.641	.641	.619	.616	.619	.641	.641	.641	.641	.641	.641	.641	.610	.629	.591	*glass2*
.731	.731	.781	.779	.781	.776	.776	.776	.774	.771	.731	.731	.781	.788	.789	.779	.776	.779	.776	.776	.731	.728	.587	*glass4*
.938	.938	.938	.938	.938	.938	.938	.938	.938	.938	.938	.938	.938	.938	.938	.938	.938	.938	.938	.938	.938	.943	.938	*glass5*
.818	.818	.818	.817	.818	.831	.831	.831	.821	.821	.821	.821	.819	.817	.818	.832	.832	.831	.823	.823	.789	.816	.763	*page-blocks-1-3-vs-4*
.995	.995	.995	.994	.994	.994	.994	.994	.994	.994	.995	.995	.995	.991	.991	.994	.994	.994	.991	.991	.990	.994	.991	*shuttle-c0-vs-c4*
.959	.959	.955	.963	.959	.984	.988	.988	.992	.992	.959	.959	.959	.980	.976	.988	.988	.988	.992	.992	.986	.950	.996	*shuttle-c2-vs-c4*
.909	.909	.909	.909	.909	.909	.909	.909	.909	.909	.909	.909	.909	.911	.911	.909	.909	.910	.910	.909	.906	.906	.917	*vowel0*
.650	.680	.650	.751	.727	.733	.747	.737	.761	.767	.673	.704	.674	.759	.733	.734	.751	.748	.761	.767	.498	.620	.504	*yeast-0-5-6-7-9-vs-4*
.614	.601	.614	.624	.619	.570	.589	.588	.636	.606	.570	.583	.578	.613	.614	.561	.581	.581	.636	.620	.540	.588	.544	*yeast-1-2-8-9-vs-7*
.610	.598	.604	.628	.606	.553	.569	.561	.572	.558	.586	.601	.594	.627	.605	.581	.558	.552	.588	.556	.541	.570	.547	*yeast-1-4-5-8-vs-7*
.714	.714	.717	.721	.718	.689	.696	.683	.713	.705	.694	.701	.653	.721	.718	.696	.682	.677	.713	.714	.588	.699	.604	*yeast-1-vs-7*
.785	.795	.785	.824	.815	.807	.807	.815	.838	.841	.818	.823	.817	.824	.813	.800	.824	.840	.838	.841	.533	.733	.561	*yeast-2-vs-4*
.773	.773	.773	.773	.773	.773	.773	.773	.773	.773	.796	.798	.797	.773	.773	.773	.773	.773	.773	.773	.614	.775	.657	*yeast-2-vs-8*
.618	.643	.616	.792	.781	.699	.785	.768	.819	.809	.744	.752	.722	.797	.770	.722	.790	.769	.818	.808	.526	.645	.551	*yeast4*
.936	.936	.936	.936	.936	.931	.935	.937	.950	.951	.895	.911	.906	.936	.935	.928	.933	.934	.949	.950	.782	.918	.831	*yeast5*
.699	.708	.704	.828	.767	.832	.860	.845	.886	.878	.675	.681	.695	.820	.736	.821	.858	.841	.886	.878	.628	.779	.650	*yeast6*
.894	.900	.898	.898	.902	.687	.763	.763	.910	.910	.896	.896	.896	.898	.890	.712	.863	.863	.912	.912	.883	.672	.877	*ecoli-0-1-4-6-vs-5*
.630	.630	.630	.630	.630	.630	.630	.630	.662	.663	.630	.630	.630	.663	.663	.630	.630	.630	.662	.663	.663	.638	.630	*ecoli-0-1-4-7-vs-2-3-5-6*
.617	.617	.637	.710	.732	.617	.617	.617	.838	.811	.757	.813	.835	.852	.839	.617	.617	.617	.846	.846	.860	.638	.735	*ecoli-0-1-4-7-vs-5-6*
.558	.558	.558	.598	.558	.558	.558	.558	.698	.698	.618	.618	.638	.675	.638	.558	.578	.578	.718	.718	.639	.570	.638	*ecoli-0-1-vs-2-3-5*
.666	.691	.691	.691	.691	.616	.641	.641	.716	.741	.716	.741	.816	.814	.814	.641	.666	.664	.789	.814	.800	.662	.782	*ecoli-0-1-vs-5*
.673	.678	.675	.686	.681	.653	.681	.706	.825	.848	.728	.728	.728	.722	.722	.672	.728	.728	.870	.870	.638	.674	.754	*ecoli-0-2-3-4-vs-5*
.585	.605	.605	.593	.593	.563	.608	.588	.643	.646	.605	.603	.605	.591	.595	.588	.608	.610	.648	.648	.588	.588	.563	*ecoli-0-2-6-7-vs-3-5*
.870	.870	.870	.901	.895	.898	.898	.898	.895	.895	.890	.887	.884	.882	.874	.898	.898	.895	.898	.898	.704	.736	.784	*ecoli-0-3-4-6-vs-5*
.633	.633	.633	.656	.659	.613	.613	.613	.801	.786	.675	.709	.707	.696	.696	.613	.633	.633	.793	.789	.728	.695	.775	*ecoli-0-3-4-7-vs-5-6*
.808	.833	.833	.867	.872	.714	.739	.736	.900	.900	.819	.839	.817	.844	.822	.733	.786	.783	.903	.903	.738	.647	.817	*ecoli-0-3-4-vs-5*
.773	.776	.773	.801	.801	.731	.778	.781	.903	.903	.792	.795	.828	.901	.901	.778	.778	.801	.903	.901	.894	.694	.854	*ecoli-0-4-6-vs-5*
.603	.603	.603	.585	.605	.548	.508	.508	.605	.603	.603	.603	.603	.603	.603	.548	.508	.508	.603	.603	.548	.567	.508	*ecoli-0-6-7-vs-3-5*
.747	.757	.738	.747	.750	.615	.710	.715	.865	.838	.812	.832	.863	.853	.853	.637	.760	.760	.857	.863	.851	.664	.780	*ecoli-0-6-7-vs-5*
.656	.623	.656	.630	.630	.620	.620	.620	.620	.620	.592	.625	.592	.630	.597	.586	.620	.620	.620	.620	.599	.615	.577	*glass-0-1-4-6-vs-2*
.600	.587	.594	.605	.616	.519	.533	.539	.626	.642	.600	.619	.612	.548	.548	.536	.523	.523	.574	.580	.518	.515	.519	*glass-0-1-5-vs-2*
.994	.994	.994	.994	.994	.994	.994	.994	.994	.994	.994	.994	.994	.994	.994	.994	.994	.994	.994	.994	.994	.984	.994	*glass-0-4-vs-5*
.995	.995	.995	.995	.995	.995	.995	.995	.995	.995	.995	.995	.995	.995	.995	.995	.995	.995	.995	.995	.950	.983	.945	*glass-0-6-vs-5*
.581	.576	.576	.576	.576	.576	.576	.576	.788	.784	.689	.707	.711	.775	.776	.576	.576	.576	.790	.786	.782	.596	.670	*yeast-0-2-5-6-vs-3-7-8-9*
.895	.895	.895	.895	.895	.896	.898	.897	.896	.895	.803	.837	.817	.895	.899	.902	.900	.896	.902	.899	.525	.741	.577	*yeast-0-2-5-7-9-vs-3-6-8*
.606	.606	.606	.616	.616	.607	.607	.607	.615	.607	.650	.635	.637	.616	.625	.623	.612	.607	.615	.607	.537	.633	.557	*yeast-0-3-5-9-vs-7-8*

Table 2: Balanced accuracy scores obtained using GNB as a base classifier

Without oversampled set										With oversampled set										OS	US	Full	Dataset
Without pruning					With pruning					Without pruning					With pruning								
REG	WEI	CON	NOR	NC	REG	WEI	CON	NOR	NC	REG	WEI	CON	NOR	NC	REG	WEI	CON	NOR	NC				
.705	.805	.703	.811	.811	.809	.809	.809	.823	.823	.714	.716	.718	.747	.653	.809	.816	.816	.827	.827	.639	.772	.739	*ecoli-0-1-3-7-vs-2-6*
.865	.865	.866	.903	.903	.878	.878	.878	.878	.878	.849	.849	.851	.835	.835	.878	.878	.878	.859	.859	.817	.858	.861	*ecoli4*
.785	.785	.782	.713	.760	.794	.788	.788	.846	.846	.777	.716	.722	.654	.599	.783	.805	.805	.761	.761	.575	.656	.552	*glass-0-1-6-vs-2*
.931	.931	.931	.934	.937	.934	.934	.934	.940	.940	.934	.934	.934	.954	.954	.934	.934	.934	.946	.946	.869	.898	.886	*glass-0-1-6-vs-5*
.779	.779	.785	.792	.792	.833	.828	.828	.766	.766	.802	.805	.807	.670	.677	.782	.782	.782	.703	.703	.630	.654	.609	*glass2*
.880	.880	.873	.895	.885	.903	.903	.903	.908	.908	.895	.900	.900	.915	.918	.903	.903	.903	.913	.913	.815	.829	.799	*glass4*
.934	.934	.937	.949	.949	.934	.934	.934	.951	.951	.946	.949	.949	.956	.963	.939	.939	.939	.956	.956	.933	.873	.898	*glass5*
.987	.987	.988	.991	.990	.990	.990	.990	.990	.990	.990	.990	.989	.994	.994	.990	.990	.990	.991	.991	.998	.959	.996	*page-blocks-1-3-vs-4*
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	.000	1	*shuttle-c0-vs-c4*
1	1	1	1	1	1	1	1	1	1	1	.950	.950	.950	.950	1	1	1	1	1	.995	.965	.950	*shuttle-c2-vs-c4*
.964	.964	.964	.965	.965	.966	.964	.964	.964	.964	.958	.958	.958	.956	.956	.966	.966	.966	.968	.968	.925	.942	.943	*vowel0*
.770	.769	.769	.784	.782	.784	.782	.782	.787	.787	.772	.774	.777	.728	.715	.776	.771	.771	.753	.753	.660	.746	.648	*yeast-0-5-6-7-9-vs-4*
.685	.687	.665	.653	.651	.721	.721	.721	.714	.714	.685	.679	.641	.602	.598	.740	.737	.737	.726	.726	.623	.631	.631	*yeast-1-2-8-9-vs-7*
.596	.598	.599	.601	.601	.619	.605	.605	.610	.610	.591	.599	.576	.540	.549	.618	.618	.618	.608	.608	.541	.569	.533	*yeast-1-4-5-8-vs-7*
.697	.698	.697	.691	.706	.701	.690	.690	.680	.680	.726	.731	.735	.710	.696	.717	.717	.717	.718	.718	.603	.692	.651	*yeast-1-vs-7*
.951	.951	.951	.958	.958	.943	.943	.943	.957	.957	.937	.937	.937	.901	.902	.937	.937	.937	.918	.918	.831	.904	.843	*yeast-2-vs-4*
.771	.772	.795	.775	.770	.817	.817	.817	.748	.748	.801	.808	.817	.716	.716	.760	.755	.755	.770	.770	.692	.726	.690	*yeast-2-vs-8*
.853	.853	.842	.854	.872	.854	.852	.852	.858	.858	.840	.833	.818	.710	.684	.845	.845	.845	.862	.862	.621	.805	.664	*yeast4*
.961	.961	.959	.960	.960	.965	.965	.965	.966	.966	.966	.966	.955	.962	.963	.967	.966	.966	.968	.968	.850	.932	.843	*yeast5*
.858	.858	.858	.865	.862	.847	.847	.847	.852	.852	.872	.863	.855	.765	.768	.852	.850	.850	.855	.855	.751	.813	.746	*yeast6*
.902	.902	.900	.904	.904	.883	.883	.883	.890	.890	.860	.860	.858	.842	.819	.867	.863	.863	.867	.867	.800	.813	.775	*ecoli-0-1-4-6-vs-5*
.847	.847	.848	.853	.865	.845	.842	.842	.833	.833	.849	.848	.851	.874	.875	.845	.843	.843	.854	.854	.795	.805	.814	*ecoli-0-1-4-7-vs-2-3-5-6*
.883	.881	.860	.856	.856	.869	.883	.883	.886	.886	.869	.869	.869	.859	.866	.869	.869	.869	.862	.862	.853	.801	.865	*ecoli-0-1-4-7-vs-5-6*
.820	.820	.842	.822	.822	.837	.837	.837	.797	.797	.804	.804	.804	.771	.773	.804	.804	.804	.786	.786	.745	.808	.762	*ecoli-0-1-vs-2-3-5*
.816	.816	.816	.816	.816	.825	.825	.825	.836	.836	.830	.830	.832	.843	.845	.839	.834	.834	.841	.841	.810	.826	.861	*ecoli-0-1-vs-5*
.834	.834	.834	.814	.811	.817	.817	.817	.847	.847	.856	.856	.856	.839	.842	.828	.828	.828	.836	.836	.828	.847	.808	*ecoli-0-2-3-4-vs-5*
.815	.815	.815	.820	.820	.820	.818	.818	.823	.823	.835	.835	.835	.803	.803	.820	.820	.820	.840	.840	.800	.797	.795	*ecoli-0-2-6-7-vs-3-5*
.907	.907	.904	.929	.932	.901	.901	.901	.907	.907	.915	.915	.915	.878	.881	.923	.915	.915	.893	.893	.810	.846	.781	*ecoli-0-3-4-6-vs-5*
.886	.886	.886	.884	.884	.884	.884	.884	.890	.890	.881	.881	.881	.894	.896	.885	.883	.883	.892	.892	.823	.812	.818	*ecoli-0-3-4-7-vs-5-6*
.928	.928	.928	.928	.928	.931	.931	.931	.908	.908	.917	.917	.917	.897	.897	.917	.917	.917	.897	.897	.854	.864	.856	*ecoli-0-3-4-vs-5*
.901	.901	.881	.862	.859	.906	.906	.906	.890	.890	.895	.895	.901	.861	.861	.901	.901	.901	.884	.884	.825	.858	.834	*ecoli-0-4-6-vs-5*
.828	.828	.828	.835	.832	.838	.838	.838	.845	.845	.850	.850	.853	.853	.853	.857	.857	.857	.863	.863	.847	.805	.828	*ecoli-0-6-7-vs-3-5*
.865	.865	.865	.878	.878	.875	.873	.873	.885	.885	.872	.872	.872	.870	.875	.875	.875	.875	.895	.895	.826	.815	.770	*ecoli-0-6-7-vs-5*
.810	.807	.804	.810	.810	.829	.829	.829	.747	.747	.814	.816	.808	.686	.692	.778	.778	.778	.759	.759	.659	.674	.615	*glass-0-1-4-6-vs-2*
.777	.777	.777	.783	.783	.743	.786	.786	.782	.782	.738	.738	.738	.574	.534	.738	.735	.735	.653	.653	.566	.638	.570	*glass-0-1-5-vs-2*
.982	.982	.982	.982	.982	.982	.982	.982	.982	.982	.982	.982	.982	.982	.982	.982	.982	.982	.982	.982	.994	.946	.994	*glass-0-4-vs-5*
.929	.929	.939	.934	.944	.934	.934	.934	.965	.965	.949	.949	.954	.970	.970	.965	.960	.960	.965	.965	.960	.880	.940	*glass-0-6-vs-5*
.759	.759	.763	.758	.762	.767	.767	.767	.777	.777	.779	.780	.782	.749	.725	.782	.780	.780	.782	.782	.705	.717	.742	*yeast-0-2-5-6-vs-3-7-8-9*
.901	.900	.899	.894	.894	.893	.893	.893	.894	.894	.907	.907	.907	.890	.893	.907	.907	.907	.902	.902	.861	.862	.850	*yeast-0-2-5-7-9-vs-3-6-8*
.694	.694	.694	.717	.701	.714	.714	.714	.725	.725	.707	.714	.700	.598	.593	.717	.710	.710	.658	.658	.586	.642	.670	*yeast-0-3-5-9-vs-7-8*

Table 3: Balanced accuracy scores obtained using DTC as a base classifier

model on a full, imbalanced dataset, and one where the best solution were simple *over-* or *undersampling*. In the Table 3, showing the results for the DTC classifier, we are dealing with a similar situation in which, however, *undersampling* never turns out to be the best in the tested pool of solutions.

A clearer interpretation of the results may take place after the analysis of the Table 4, showing a summary of the results achieved by individual variations of the proposed method, presenting the number of datasets for which a given variation took part in the construction of the best solution.

Classifier	**Full**	**US**	**OS**	**OSE**		**Pru.**		**Fuser**				
				NO	YES	NO	YES	REG	WEI	CON	NOR	NCI
GNB	3	1	1	10	**12**	6	**12**	6	5	6	11	**12**
DTC	3	0	2	7	**8**	7	**8**	7	6	6	**8**	**8**

Table 4: Final summary of proposed method variations.
(OSE – extending pool by oversampled dataset, Pru. – usage of pruning)

As we may observe, both the extension of the classifier pool by the model built on the oversampled dataset as well as the proposed pruning method has a positive impact on the quality of the final solution. Among the fusers, the best performers are NOR – normalizing the calculated weights for the members of the committee and NCI - complementing NOR by the accumulated support with a stronger impact of the certainty of the decision. Even just the basic ensemble construction, in its simplest form without improvements and using the decision rule without weighting, allows to achieve better results than learning on a full dataset or basic under- or oversampling.

5. Conclusions

This paper presents UMCE (*Undersampled Majority Class Ensemble*) – a hybrid method for solving the problem of binary classification of datasets with a high *imbalance ratio*, based on *k-fold division* of the *majority class* samples to create an *ensemble* of classifiers breaking one *imbalanced problem* into many balanced problems. The basic division method has been supplemented with a variant extending the pool with the *oversampled* dataset and the *post-pruning* method based on the analysis of the statistical dependencies of the classifiers response on the testing set. For the *ensemble* it were also proposed five different *fusers*.

Computer experiments have shown, that this approach led to create a method solving targeted problem and able to outperform other possible basic solutions, proving that it may be employed for real-life appliance.

Acknowledgments

This work was supported by the Polish National Science Center under the grant no. UMO-2015/19/B/ST6/01597 and by the statutory fund of the Faculty of Electronics, Wroclaw University of Science and Technology.

References

Jesús Alcalá-Fdez, Alberto Fernández, Julián Luengo, Joaquín Derrac, Salvador García, Luciano Sánchez, and Francisco Herrera. Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic & Soft Computing*, 17, 2011.

Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2009.

Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. The balanced accuracy and its posterior distribution. In *Pattern recognition (ICPR), 2010 20th international conference on*, pages 3121–3124. IEEE, 2010.

C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap. Safe-Level-SMOTE: safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In *Advances in Knowledge Discovery and Data Mining, 13th Pacific-Asia Conference 2009, Bangkok, Thailand, April 27-30, 2009, Proceedings*, pages 475–482, 2009.

T. F. Chan, G. H. Golub, and R. J. LeVeque. Updating formulae and a pairwise algorithm for computing sample variances. In H. Caussinus, P. Ettinger, and R. Tomassone, editors, *COMPSTAT 1982 5th Symposium held at Toulouse 1982*, pages 30–41, Heidelberg, 1982. Physica-Verlag HD. ISBN 978-3-642-51461-6.

N V Chawla, K W Bowyer, L O Hall, and W P Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *arXiv.org*, June 2011.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

Nitesh V. Chawla, Aleksandar Lazarevic, Lawrence O. Hall, and Kevin W. Bowyer. *SMOTEBoost: Improving Prediction of the Minority Class in Boosting*, pages 107–119. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. ISBN 978-3-540-39804-2. doi: 10.1007/978-3-540-39804-2_12. URL https://doi.org/10.1007/978-3-540-39804-2_12.

Xue-wen Chen and Michael Wasikowski. Fast: A ROC-based feature selection metric for small samples and imbalanced data classification problems. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 124–132, 2008.

William Elazmeh, Nathalie Japkowicz, and Stan Matwin. Evaluating misclassifications in imbalanced data. In *Proceedings of the 17th European Conference on Machine Learning*, ECML'06, pages 126–137, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-45375-X, 978-3-540-45375-8. doi: 10.1007/11871842_16. URL http://dx.doi.org/10.1007/11871842_16.

Alberto Fernández, María José del Jesus, and Francisco Herrera. Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets. *International Journal of Approximate Reasoning*, 50(3):561–577, 2009.

M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42 (4):463–484, July 2012. ISSN 1094-6977. doi: 10.1109/TSMCC.2011.2161285.

H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

H. He, Y. Bai, E. A. Garcia, and S. Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *Proceedings of the International Joint Conference on Neural Networks, 2008, part of the IEEE World Congress on Computational Intelligence, 2008, Hong Kong, China, June 1-6, 2008*, pages 1322–1328, 2008.

Nathalie Japkowicz, Catherine Myers, and Mark Gluck. A novelty detection approach to classification. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 518–523, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1-55860-363-8, 978-1-558-60363-9. URL http://dl.acm.org/citation.cfm?id=1625855.1625923.

Bartosz Krawczyk, Michal Wozniak, and Boguslaw Cyganek. Clustering-based ensembles for one-class classification. *Information Sciences*, 264:182–195, 2014.

Wei-Yin Loh. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23, 2011.

V. Lopez, A. Fernandez, J. G. Moreno-Torres, and F. Herrera. Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. open problems on intrinsic data characteristics. *Expert Systems with Applications*, 39(7):6585–6608, 2012.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Y. Sun, A. K. C. Wong, and M. S. Kamel. Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(4):687–719, 2009.

S. Wang, H. Chen, and X. Yao. Negative correlation learning for classification ensembles. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, July 2010. doi: 10.1109/IJCNN.2010.5596702.

Shuo Wang, Leandro L. Minku, and Xin Yao. A systematic study of online class imbalance learning with concept drift. *CoRR*, abs/1703.06683, 2017. URL http://arxiv.org/abs/1703.06683.

Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6): 80–83, 1945.

[C9]
Paweł Ksieniewicz i Michał Woźniak. “Imbalanced Data Classification Based on Feature Selection Techniques”. W: *Intelligent Data Engineering and Automated Learning – IDEAL 2018*. Red. Hujun Yin i in. Cham: Springer International Publishing, 2018, s. 296–303. DOI: 10.1007/978-3-030-03496-2_33

Imbalanced Data Classification Based on Feature Selection Techniques

Paweł Ksieniewicz(✉) and Michał Woźniak

Department of Systems and Computer Networks,
Wrocław University of Science and Technology, Wrocław, Poland
{pawel.ksieniewicz,michal.wozniak}@pwr.edu.pl

Abstract. The difficulty of the many classification tasks lies in the analyzed data nature, as disproportionate number of examples from different class in a learning set. Ignoring this characteristics causes that canonical classifiers display strongly biased performance on imbalanced datasets. In this work a novel classifier ensemble forming technique for imbalanced datasets is presented. On the one hand it takes into consideration selected features used for training individual classifiers, on the other hand it ensures an appropriate diversity of a classifier ensemble. The proposed method was tested on the basis of the computer experiments carried out on the several benchmark datasets. Their results seem to confirm the usefulness of the proposed concept.

Keywords: Machine learning · Classification
Imbalanced data · Feature selection · Random search

1 Introduction

Most of classifier training methods assume that the numbers of objects from each classes are roughly equal in a learning set. However, in many real-life decision tasks this assumption is not fulfilled. We may deal with examples from classes being abundant and easy to collect and with the classes where number of examples is small and hard to access [1,11]. Therefore, there is a need of constructing effective predictive systems which can take into consideration imbalanced data distributions [3]. Let us present shortly the main groups of algorithms in imbalanced data classification.

Data Preprocessing. Such methods modify the learning set, before a classifier is being trained [12]. They should manipulate learning examples to obtain a balanced dataset. One may achieve this by either removing samples from the majority classes (*undersampling*), or adding new object from the minority ones (*oversampling*). One have also mention techniques of dimensionality reduction as feature selection which may be also applied to this task [4].

Algorithm-Level Methods. They modify the classifier learning procedure to take into consideration imbalanced data distributions. Usually, they use non-symmetric loss-function [7] to assign higher cost to the error committed on

H. Yin et al. (Eds.): IDEAL 2018, LNCS 11315, pp. 296–303, 2018.
https://doi.org/10.1007/978-3-030-03496-2_33

minority class objects. Another approaches employ one-class classifier learning techniques, where a given class is learned only and the objects which do not belong to it are treated as outliers.

Hybrid Solutions. They are trying to exploit the strengths of the previously discussed methods and to combine them with other techniques. Usually ensemble learning is used [13], which is able to train set of diverse individual predictors, which may take into consideration the data imbalance and propose such combination rule which can make a high quality decision on complex data.

In this paper, we introduce a novel hybrid technique that employs feature selection techniques to train a pool of individual classifiers used by a classifier ensemble. To avoid the overfitting a regularization techniques are used which on the one hand ensures that the pool of classifier is diverse, i.e., subsets of features should be different for each classifier and on the other hand the number of features used by all individuals should be as small as possible. To train the pool we use simple techniques based on random search, but experimental study carried out on a number of benchmarks prove that the proposed method is able to return satisfactory performance.

2 Proposed Algorithm

Selecting appropriate set of the feature is an important data preprocessing step in classifier learning [8] and Chawla et al. [4] underlined its crucial role when classification model is train on the basis of imbalanced data. Traditional feature selection techniques usually use criterion based on the accuracy with a factor responsible for regularization, i.e., discourages learning too complex model to avoid the overfitting. While the methods dedicated for imbalanced data [9,14] usually employs metrics related with binary problem, as *g-mean* [6].

To present the proposed solution, firstly let us formulate the classification problem.

2.1 Problem Formulation

Classifier Ψ makes a decision by assigning an observed object into one of predefined classes derived from the set of possible labels $\mathcal{M} = \{1, 2, ..., M\}$ [7]. Each object is described by the set of attributes (features) gathered in the feature vector x belonging to d dimensional feature space $\mathcal{X}$

$$x = [x^1, x^2, \ldots, x^d]^T \in \mathcal{X} \subseteq \mathcal{R}^d, \tag{1}$$

The aim of a feature selection algorithm is to choose k valuable features only to avoid so-called *course of dimensionality* [5].

$$x = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \ldots \\ x^{(d)} \end{bmatrix} \rightarrow \overline{x} = \begin{bmatrix} \overline{x}^{(1)} \\ \overline{x}^{(2)} \\ \ldots \\ \overline{x}^{(k)} \end{bmatrix}, \quad k < d \tag{2}$$

Let us also define a pool of individual classifiers

$$\Pi = \{\Psi_1, \Psi_2, \ldots, \Psi_K\} \tag{3}$$

where Ψ_k denotes the k-th elementary classifier. To ensure diversity of the pool we will train the individuals on the basis of the different set of features. Let's propose the following representation of the classifier pool Π as word of bits:

$$\Pi = \left[\left[b_1^1, b_1^2, ..., b_1^d\right] \left[b_2^1, b_2^2, ..., b_2^d\right] ... \left[b_K^1, b_K^2, ..., b_K^d\right]\right] \tag{4}$$

where b_i^j denotes if the jth feature is used by the ith classifier.

2.2 Criterion

In our algorithm we use the following optimization criterion based on *Balanced Accuracy*

$$Q(\Pi) = BAC(\Pi) - \alpha * \frac{no - features(\Pi)}{d} + \beta * \frac{av - Hamming(\Pi)}{d} \tag{5}$$

where $BAC(\Pi)$ denotes balanced accuracy of the classifier ensemble based on the ensemble represented by Π. The first regularization factor $no-features(\Pi)$ is responsible for the number of the selected features, i.e., number of features used by all individual in the ensemble, while the second regularization factor $av-Hamming(\Pi)$ is the average Hamming distance between the words represented individuals in Π. It is a kind of a diversity measure [13], which encourages to select different features by different individuals. α and β are the parameters of the algorithm, which should be set experimentally.

2.3 Algorithm Description

Firstly, the algorithm randomly generates the population of ensembles

$$Population = \{\Pi_1, \Pi_2, ..., \Pi_S\} \tag{6}$$

A size of the *Population* is an input parameter. Its value S is set arbitrary, but we have to take into consideration, that on the one hand the larger S guarantees the more comprehensive optimization, but on the other hand, the larger S requires the higher computational effort.

Individuals in the population are evaluated by criterion Eq. 5 calculated on the basis of Algorithm 1 using samples stored in the training set. We decide to select the best evaluated ensemble only.

The *combination rule* of chosen ensemble is carried out by *averaging* the *support vectors* received from the members of a pool. It is important, that for such approach, it is necessary to use a *probabilistic classification model*. Three combination rules are proposed for further analysis:

1. **R**—basic accumulation of support without weighing the committee members.

Algorithm 1. Criterion count

Input: pool of individual classifiers Π, training set $\mathcal{TS}$
Parameters: α, $beta$
Output: value of criterion 5 for Π

$counter \leftarrow 0$
$nobits \leftarrow 0$
$word \leftarrow [00..0]$
for $i \leftarrow 1$ **to** $K-1$ **do**
  for $j \leftarrow i$ **to** K **do**
    $counter \leftarrow counter + 1$
    $nobits \leftarrow$ nobits + number of bits of $\left[b_i^1, b_i^2, ..., b_i^d\right] XOR \left[b_j^1, b_j^2, ..., b_j^d\right]$
  end for
  $word \leftarrow wordOR \left[b_i^1, b_i^2, ..., b_i^d\right]$
end for
$word \leftarrow wordOR \left[b_K^1, b_K^2, ..., b_K^d\right]$
$no - features \leftarrow$ number of bits in $word * \frac{1}{d}$
$av - Hamming \leftarrow \frac{nobits}{counter*d}$
$BAC \leftarrow$ balanced accuracy of Π calculated on $\mathcal{TS}$
$criterion \leftarrow BAC - \alpha * no - used - features + \beta * av - Hamming - dist$
return $criterion$

2. **W**— weighted aggregation, where weights are proportional to balanced accuracy values achieved by individual classifiers.
3. **N**—weighted aggregation, where weights are proportional to balanced accuracy values achieved by individual classifiers and additionally weights are subjected to *MinMax* scaling.

3 Experimental Study

Experimental investigations, backed up with statistical analysis of the results, were conducted to evaluate the practical usefulness of the proposed strategy. In the remainder of this section we describe set-up of the study, present obtained results and discuss achieved outcomes.

3.1 Set-Up

For the experimental evaluation of the proposed method, a series of benchmark datasets available on the KEEL repository [2] were used. Selection was made to ensure wide scope of 35 binary problems with *Imbalance Ratio* IR varying from 1 to around 40. The overview of chosen datasets, informing about their IR and number of features, was included in Table 1.

To allow a reliable comparison of literature methods, datasets from KEEL repository are pre-divided into folds. It led to employ *k-fold cross-validation* with 5 folds in the experimental procedure. Due to strong bias of regular classification metrics towards majority class, to ensure reliable results, all scores are presented

as *balanced accuracy*, according to its implementation from the development version (0.20.dev0) of the *scikit-learn* library [10].

Implementation of the experimental procedure, as well as the implementation of the method itself, has been prepared according to the *scikit-learn* library API, using *Gaussian Naive Bayes* as a base classifier. Besides the variations of a method, to provide a comparative result, each problem was also evaluated on a full-featured representation of a dataset. To analyze a paired dependency between the classifiers outputs, the signed-rank *Wilcoxon* test was employed.

The implementation of the method proposed in following paper, as well as the script allowing to reconstruct conducted research can be found in repository[1].

3.2 Results

First step of experimental evaluation was optimization procedure to obtain the best α and β values in the context of *balanced accuracy*. It has been conducted with a *Grid Search* approach, analyzing 7 values evenly dividing the range from 0 to 1. Example visualizations of results for three datasets are presented on Fig. 1. Presentation for all datasets is available at website[2].

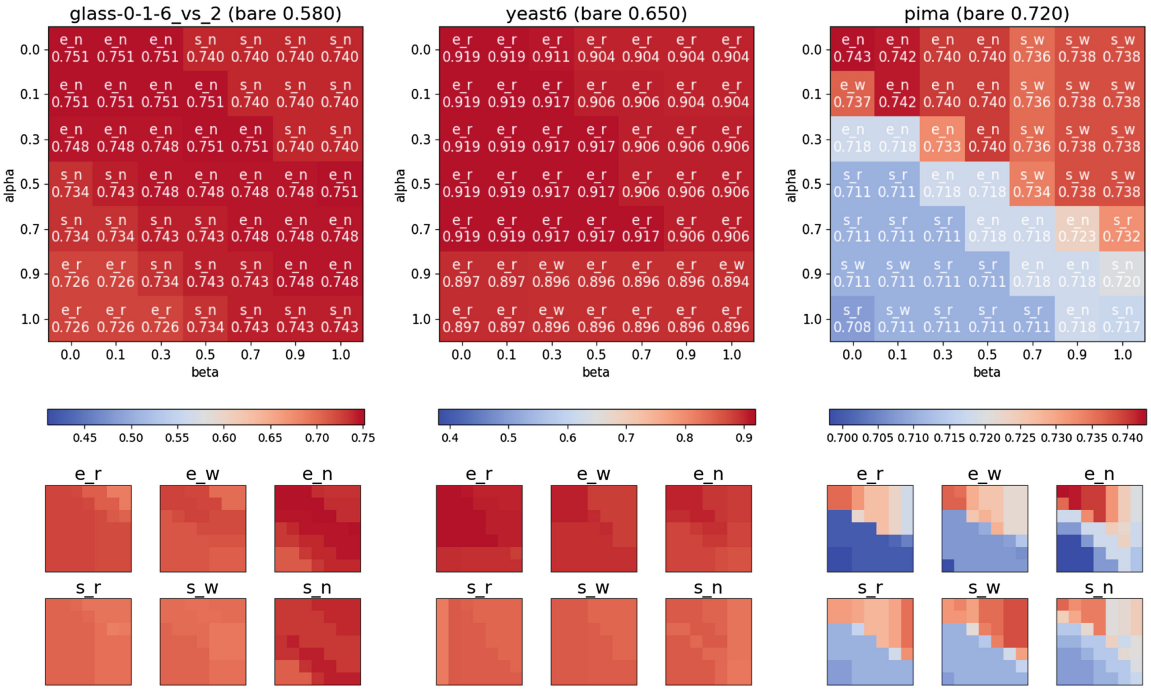


Fig. 1. Examples of α and β influence on classification quality for best (top) and every (bottom) approach. Blue indicates result worse than full-featured classification, red – a better result. (Color figure online)

[1] https://github.com/w4k2/ideal2018.
[2] http://w4k2.github.io/ideal2018.

Table 1. Balanced accuracy scores obtained with the optimized hyperparameters α and β on datasets selected to experimental evaluation. **Full** stands for the results of the classifier using all features, **Ensemble** stands for results of classifier ensemble using different combination rules described in Sect. 2.3, and **Best in ensemble** stands for balance accuracy of the best individual in the ensemble.

Dataset	IR	F.	Params.		Balanced Accuracy Scores						
					Full	Ensemble			Best in ensemble		
			α	β		E_R	E_W	E_N	S_R	S_W	S_N
australian	1	14	.0	.0	0.777	0.852	0.855	0.877	0.878	0.876	0.861
heart	1	13	.0	.0	0.838	0.870	0.878	0.874	0.847	0.847	0.868
glass0	2	9	.3	.5	0.700	0.746	0.746	0.749	0.750	0.750	0.763
glass1	2	9	.1	.0	0.671	0.721	0.719	0.710	0.738	0.723	0.732
pima	2	8	.0	.0	0.720	0.736	0.737	0.743	0.731	0.732	0.736
wisconsin	2	9	.0	.0	0.969	0.976	0.976	0.976	0.967	0.967	0.974
yeast1	2	8	.0	.0	0.519	0.695	0.699	0.680	0.654	0.659	0.660
glass0123vs456	3	9	.0	.0	0.869	0.891	0.898	0.910	0.900	0.910	0.910
hepatitis	5	19	.0	.0	0.687	0.880	0.903	0.877	0.872	0.872	0.881
glass6	6	9	.0	.0	0.891	0.939	0.942	0.945	0.945	0.959	0.959
yeast3	8	8	.0	.0	0.605	0.904	0.895	0.915	0.841	0.840	0.813
glass015vs2	9	9	.0	.0	0.519	0.711	0.728	0.765	0.691	0.696	0.710
glass04vs5	9	9	.0	.0	0.994	0.994	0.994	0.994	0.994	0.994	0.994
yeast0256vs3789	9	8	.0	.3	0.670	0.689	0.689	0.737	0.753	0.748	0.771
yeast02579vs368	9	8	.0	.0	0.577	0.912	0.911	0.900	0.878	0.878	0.894
yeast0359vs78	9	8	.0	.0	0.557	0.668	0.662	0.621	0.607	0.600	0.607
yeast05679vs4	9	8	.0	.0	0.504	0.780	0.763	0.720	0.706	0.702	0.710
yeast2vs4	9	8	.3	.0	0.561	0.897	0.887	0.892	0.838	0.904	0.885
glass016vs2	10	9	.0	.0	0.580	0.726	0.726	0.751	0.705	0.700	0.731
vowel0	10	13	.5	.3	0.917	0.898	0.914	0.911	0.924	0.929	0.933
glass0146vs2	11	9	.1	.0	0.577	0.747	0.761	0.739	0.724	0.746	0.773
glass06vs5	11	9	.0	.0	0.945	0.995	0.995	0.995	0.960	0.960	0.995
glass2	12	9	.0	.0	0.591	0.767	0.775	0.747	0.718	0.721	0.721
shuttlec0vsc4	14	9	.0	.0	0.991	1.000	1.000	1.000	1.000	1.000	1.000
glass4	15	9	.1	.0	0.587	0.609	0.609	0.718	0.768	0.766	0.753
pageblocks13vs4	16	10	.0	.0	0.763	0.786	0.866	0.949	0.867	0.879	0.928
glass016vs5	19	9	.0	.0	0.941	0.991	0.991	0.989	0.989	0.989	0.989
shuttlec2vsc4	20	9	.0	.0	0.996	1.000	1.000	1.000	0.996	0.996	1.000
yeast1458vs7	22	8	.0	.0	0.547	0.592	0.588	0.574	0.556	0.556	0.569
glass5	23	9	.0	.0	0.938	0.988	0.988	0.988	0.988	0.988	0.988
yeast2vs8	23	8	.0	.0	0.657	0.799	0.810	0.774	0.774	0.774	0.774
yeast4	28	8	.0	.0	0.551	0.817	0.783	0.797	0.679	0.670	0.651
yeast1289vs7	31	8	.0	.0	0.544	0.683	0.701	0.706	0.629	0.628	0.606
yeast5	33	8	.0	.0	0.831	0.963	0.964	0.973	0.954	0.947	0.945
yeast6	41	8	.0	.0	0.650	0.919	0.905	0.903	0.821	0.857	0.858

The results of the evaluation after optimization procedure are presented in Table 1, which has been divided to present a *balanced accuracy* obtained on different variations of the method, using whole ensemble or just its best member,

according to the different fusers (R – regular, W – weighted and N – normalized weights). Such division led to the number of 6 analyzed approaches.

Scores for the method were supplemented by the quality of a single model trained on a whole possible feature space. The green color in table indicates the statistical dependency to the best result and underline – the highest *balanced accuracy* obtained on a given dataset.

The presented results clearly showed that feature selection plays important role for imbalanced data classification. Our proposition usually outperforms the results obtained by the classifier using the whole set of features. It also behaves better (22 out of 35 datasets) than the best classifier in the pool. It has been probably caused by very naive optimization method (random search) used in this work.

4 Conclusions and Future Directions

The novel hybrid classification method for imbalanced data classification was presented. It employs ensemble learning to increase performance (*balanced accuracy*) of the combined classifier. To ensure the appropriate level of diversity, each individual is trained on the basis of selected features. Nevertheless, in contrast with well-known methods based on randomly chosen features (as *Random Subspaces*), the choice of the features is the results of the optimization procedure. The optimization criterion takes into consideration not only the performance of the classifier, but to protect against *overfitting* it encourages to build the ensemble of diverse individuals which do not use too many features. Additionally, we observed that the proposed method can significantly outperform the classifier based on whole set of features.

As the future works we are going to use more sophisticated optimization procedure based on genetic approach, but we also realize that it will negatively impact computational complexity, therefore we will focus on the method which can be run in distributed computing systems as GPU or SPARK. Additionally, we plan to definitely extend the scope of experiments to compare our methods with other methods as *Random Subspaces* or *Decision Forrest*.

Acknowledgments. This work was supported by the Polish National Science Center under the grant no. UMO-2015/19/B/ST6/01597 as well as Statutory Found of the Faculty of Electronics, Wroclaw University of Science and Technology.

References

1. Ahmed, F., Samorani, M., Bellinger, C., Zaïane, O.R.: Advantage of integration in big data: feature generation in multi-relational databases for imbalanced learning. In: 2016 IEEE International Conference on Big Data, BigData 2016, Washington DC, USA, 5–8 December 2016, pp. 532–539 (2016)
2. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. J. Multiple-Valued Logic Soft Comput. **17** (2011)

3. Branco, P., Torgo, L., Ribeiro, R.P.: A survey of predictive modeling on imbalanced domains. ACM Comput. Surv. **49**(2), 1–50 (2016)
4. Chawla, N.V., Japkowicz, N., Kotcz, A.: Editorial: special issue on learning from imbalanced data sets. SIGKDD Explor. Newsl. **6**(1), 1–6 (2004)
5. Domingos, P.: A few useful things to know about machine learning. Commun. ACM **55**(10), 78–87 (2012)
6. Du, L.M., Xu, Y., Zhu, H.: Feature selection for multi-class imbalanced data sets based on genetic algorithm. Ann. Data Sci. **2**(3), 293–300 (2015)
7. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley, New York (2001)
8. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (2003)
9. Maldonado, S., Weber, R., Famili, F.: Feature selection for high-dimensional class-imbalanced data sets using support vector machines. Inf. Sci. **286**, 228–246 (2014)
10. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
11. Porwik, P., Doroz, R., Orczyk, T.: Signatures verification based on PNN classifier optimised by PSO algorithm. Pattern Recogn. **60**, 998–1014 (2016)
12. Triguero, I., Galar, M., Merino, D., Maillo, J., Bustince, H., Herrera, F.: Evolutionary undersampling for extremely imbalanced big data classification under apache spark. In: IEEE Congress on Evolutionary Computation, CEC 2016, Vancouver, BC, Canada, 24–29 July 2016, pp. 640–647 (2016)
13. Wozniak, M., Graña, M., Corchado, E.: A survey of multiple classifier systems as hybrid systems. Inf. Fusion **16**, 3–17 (2014)
14. Yin, L., Ge, Y., Xiao, K., Wang, X., Quan, X.: Feature selection for high-dimensional imbalanced data. Neurocomputing **105**, 3–11 (2013)

[C10]
Paweł Ksieniewicz, Bartosz Krawczyk i Michał Woźniak. "Ensemble of Extreme Learning Machines with trained classifier combination and statistical features for hyperspectral data". W: *Neurocomputing* 271 (2018), s. 28–37. DOI: 10.1016/j.neucom.2016.04.076

Neurocomputing 271 (2018) 28–37

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Ensemble of Extreme Learning Machines with trained classifier combination and statistical features for hyperspectral data

Paweł Ksieniewicz[a], Bartosz Krawczyk[b,*], Michał Woźniak[a]

[a] Department of Systems and Computer Networks, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50–370 Wrocław, Poland
[b] Department of Computer Science, Virginia Commonwealth University, Richmond, VA 23284, USA

ARTICLE INFO

Article history:
Received 27 December 2015
Accepted 25 April 2016
Available online 6 July 2017

Keywords:
Ensemble learning
Extreme Learning Machines
Hyperspectral imaging
Computer vision
Feature extraction
Dimensionality reduction
Image classification

ABSTRACT

Remote sensing and hyperspectral data analysis are areas offering wide range of valuable practical applications. However, they generate massive and complex data that is very difficult to be analyzed by a human being. Therefore, methods for efficient data representation and data mining are of high interest to these fields. In this paper, we introduce a novel pipeline for feature extraction and classification of hyperspectral images. To obtain a compressed representation we propose to extract a set of statistical-based properties from these images. This allows for embedding feature space into fourteen channels, obtaining a significant dimensionality reduction. These features are used as an input for the ensemble learning based on randomized neural networks. We introduce a novel method for forming ensembles of Extreme Learning Machines based on randomized feature subspaces and a trained combiner. It is based on continuous outputs and uses a perceptron-based learning scheme to calculate weights assigned to each classifier and class independently. Extensive experiments carried on a number of benchmarks images prove that using proposed feature extraction and extreme learning ensemble leads to a significant gain in classification accuracy.

1. Introduction

Because we are living in big data century, therefore the efficient analytical tools which can analyze the huge volume of multidimensional data are still focus of intense research. One of the example of such a data is hyperspectral imaging, which is widely used in agriculture, mineralogy etc. One can say that *if a picture is worth 1000 words, a hyperspectral image is worth almost 1000 pictures.*[1] Hyperspectral cameras are able to capture hundreds of monochrome images correlated with a particular spectrum, nevertheless they still need to be analyzed manually, what is highly time consuming and requires very expensive manual labeling. Therefore methods which can use partially labelled data are desirable tools for hyperspectral image classification [6]. One of the very promising direction is active learning paradigm [27,35], which employs an iterative data labeling and classifier training strategy with as small as possible set of training examples. A complementary approach proposes an efficient data representation of hyperspectral images which could be used by a classification system. As we deal with multi-class and high-dimensional problem, we require a highly effective pattern classification system to be able to analyze such data.

Classifier ensembles are nowadays recognized as the on of the most promising direction in pattern classification [42]. This approach exploit the conclusions from so-called Wolpert's *no free lunch* theorem, that there is not a single classifier, which is the best one for all decision tasks, but each model has its own, specific domain of competence [41] where it may outperform other competing algorithms. Let's formulate the main presumptions of using such a classification model [13]

- Classifier ensembles behave well both in the case when a learning set is very small and when we have a huge amount of learning examples at our disposal. In the first case, classifier ensemble can exploit methods based on bootstrapping [30], while for the second case it allows to train individuals on partitions of dataset.
- Classifier ensemble may outperform the best individual classifier [10] and under some conditions (e.g., majority voting by a group of individual classifiers committed error independently) this improvement has been proven analytically [26].
- Many classifier training methods, as decision tree [34], are heuristic search algorithms which usually suffer from local

* Corresponding author.
E-mail addresses: pawel.ksieniewicz@pwr.edu.pl (P. Ksieniewicz), bkrawczyk@vcu.edu (B. Krawczyk), michal.wozniak@pwr.edu.pl (M. Woźniak).

[1] J.P.Ferguson, An Introduction to Hyperspectral Imaging, Photonics and Analytical Marketing Ltd.

http://dx.doi.org/10.1016/j.neucom.2016.04.076
0925-2312/

optima. Therefore, the ensemble learning approach approach is equivalent to a multi-start local random search which increases the probability of finding an optimal model.
- Classifier ensemble may be easy implemented in efficient computing environments such as parallel and multithreaded computer architectures [40].

In this work, we propose a novel ensemble dedicated to analysis of hyperspectral data. Its base classifiers are being built on the basis of decomposed color channels. This assures their initial diversity, as every color channel carries different information. We further augment this idea by using a trained fuser, based on perceptron learning. This allows us to assign higher weights to more competent classifiers. As not all of the channels carry equally useful features, we boost the influence of the most relevant ones on the final decision of the ensemble.

As the basis of our ensemble we propose to use Extreme Learning Machines (ELMs), a popular branch of randomized neural networks. Due to their efficacy and low training complexity they have been reported to display high usefulness for the hyperspectral data analysis task [28,31]. However, methods for constructing efficient ELMs ensembles still require development [4].

The main contributions of the paper are as follows:

1. A novel proposition of the statistical-based feature extraction from hyperspectral images.
2. An efficient ELMs ensemble architecture based on trained combiner.
3. Application of the proposed features and ensemble structure together with Random Subspaces method to the problem of hyperspectral image classification.
4. Experimental evaluation of the proposed approach.

In Section 2, we shortly introduce into hyperspectral image analysis, then in Section 3 presents the proposition how to extract the valuable features from hyperspectral data. Section 4 describes the classification methods based on ensemble approach. The experimental evaluation is presented in Section 5. At the end, in Section 6 shows conclusions and possible usages of the proposed approach.

2. Hyperspectral image analysis

Natural perception of electromagnetic waves is limited to only four features of information spectrum. Each of them is a single chrome channel, which composed together by a human brain brings its owner an chemical illusion called color vision. The color can be interpreted as short vector, most often builded by three values. Its most popular representation is based on human perception on daylight, described by Svaetichin in 1956 [37] RGB model. Place of S *Mand Lcone cells* is taken there for channels of particular light impressions.

Hyperspectral image is a collection of high-resolution monochromatic pictures covering large spacial region for broad range of wavelengths. Structurally it is a three-dimensional matrix of reflectance. First two dimensions are standard lengths of a flat projection. The third is a spectral depth. Main idea of hyperspectral imaging is minimization of range covered by every band with maximization of band number. The current industrial standard, AVIRIS spectrometer, captures images with 224 channels in range $0.4 - 2.5\,\mu$m.

A slice taken from hyperspectral cube provides us information of reflectance of the area for a given spectral band. Taking a vector alongside the spectral band axis provides us spectral signature, which carries information about reflectance of one particular pixel for every covered spectral band. Example slice and signature are presented in Fig. 1.

Signatures are used to detect type of material represented by pixel on an image. It is possible to distinguish type of ground, vegetation, used building material, rock strata or many other.

Method of separation of an hyperspectral image into channels is based on human perception of colorful images. Its main base is to replace a *reading* from *photoreceptors* with statistical measurement, doing e.g., elementary statistical operations on signature vector. Monochromatic image from this kind of metric can turn into channel used to construct colorful picture or, after posterization, set of labels. It also implements a method of separation of homogenous areas on image, used also to filter noisy ranges of spectrum.

After the image color decomposition, we need to apply machine learning algorithms in order to conduct segmentation or classification. Among a plethora of classification methods, ensembles has gained a significant interest of researchers over the last decade [21]. Combining multiple classifiers can lead to a significant improvement of the accuracy in comparison to single learner. There are many different methods for forming efficient ensembles [42], but they all share several fundamental ideas. In order for the ensemble to work, we need to have more than one classifiers at our disposal. They can be trained on the given dataset, or supplied by heterogeneous sources. A special attention should be paid to the properties of used classifiers. For an ensemble to work properly, it must consist of classifiers that at the same time display a high individual accuracy and are mutually complementary with each other. As, in most cases, not all of the available classifiers satisfy this condition, one needs to discard the irrelevant models. This step has a crucial impact on the quality of the formed committee and is known as classifier selection or ensemble pruning [11]. Another important part of ensemble design is the combination rule. It will fuse the individual outputs of base classifiers into a single committee decision. This task can be tackled in two different ways: with untrained or trained fuser. Untrained fusers (such as voting) [39] are simple and straightforward to use, but can be subject of performance limitations. Trained fusers adapt their behavior to the analyzed data, but require some time to establish their rules and a dedicated training set [25].

Most common method of generating false-color pictures from hyperspectral data is maping three bands from a wide signature into RGB channels. For case of spectral depth reduction, the most popular standard is PCA (*Principal Components Analysis*) [1]. Three, richest in information, principal components from hyperspectral cube are mapped to various color models chanels (RGB, HSL, HSV) [38].

Some works suggest to balance S/N(*Signal-to-noise ratio*) to enhance contrast of an image [15] and reduce noise impact.

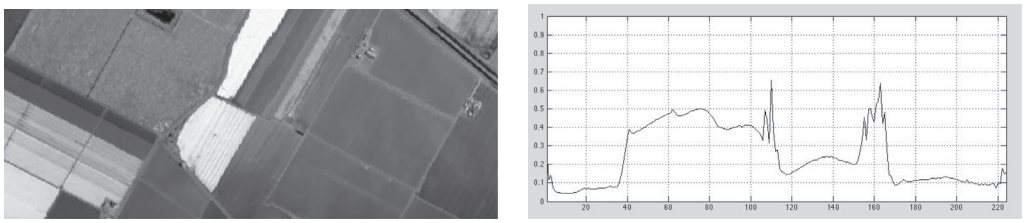
3. Proposed statistical features for hyperspectral images

We propose a novel method for simultaneous feature extraction and low-dimensional embedding of hyperspectral images. It is based on the idea of creating a new representation, evolving from human color perception, preserving as much information as possible, with simple, time efficient computations.

We are interpreting the matrix of cone cells reacting on same wavelengths as a transformation, projecting three-dimensional input onto two-dimensional result. Hyperspectral imaging is there a discrete form of this three-dimensional input, which provides enough data to acquire other transformation functions. So created artificial cone cells matrixes will generate our statistical features.

New proposition is a significant extension of our previous proposal [24], introducing procedure of spatial blurring, normalization and histogram equalization. Also the extended set of statistical features is proposed.

The procedure run as follows. At the start, the class edges are recognized and calculated. Next, the collection of side information



(a) Slice of single spectral band (b) Example spectral signature

Fig. 1. Hyperspectral image elements.

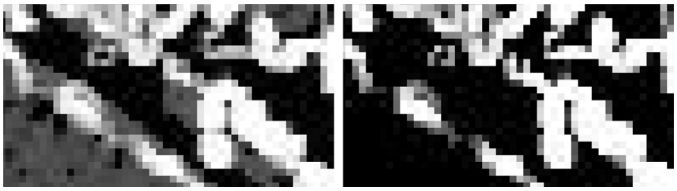


Fig. 2. Mask of region borders before (left) and after filtering (right).

produced during first step, lets us to generate a filter for noisy bands of image. Next the features of filtered image are computed and, at the end, they are prepared for classification.

3.1. Noise detection

A value denivelation in finite neighborhood of every pixel can be used to detect borders between non-texture areas of picture [12]. A side effect of this method is the measurement of entropy ($\bar{\mathrm{H}}$), calculated from amount of all values (ρ) divided by calculation of pixels per layer (ppl).

$$\bar{\mathrm{H}} = \frac{\sum \rho}{ppl} \tag{1}$$

While every hyperspectral cube contains wavelengths with high noise ratio, adequate threshold to drain most of them would be a mean value of entropy. To separate hills of entropy changes we are using information about its dynamics. An vector of dynamics was made in a way analogous to edge detection, by calculating discrepancy between actual ($\bar{H}$) and next value ($\bar{H}'$) on the vector of entropy.

$$\mathrm{D}\bar{\mathrm{H}} = |\bar{H} - \bar{H}'| \tag{2}$$

Mean dynamics filter was generated in an analogous way as the one for entropy. Concluding filter was the *blend* of mean entropy and mean dynamics filters. Fig. 2 presents difference between unfiltered and filtered mask of region borders.

3.2. Feature computation

Filtering out the noise makes possible an effective usage of simple statistical operations like maximum or minimum, and improve the precision of average, mean, mode or median value. We have proposed a set of fourteen features. Fig. 3 provides HSV visualization three example features. Complete collection of statistical features is presented in Table 1.

3.3. Preparing features for classification

As we can see in Fig. 3, while some statistical features are giving us clear information, enough to distinguish classes in data, some of them seems completely useless. To extract, boost and stabilize data coming from them, we have added three more steps of processing.

To stabilize information, we used the anisotropic diffusion [33]. Normalization paired with histogram equalization brought us more contrast and extraction of sparse values. Fig. 4 shows the same set of metrics, after these three steps.

Result of this three-staged process is presented in Fig. 4.

4. Ensemble of Extreme Learning Machines

In this section, we will present a brief description of the ELMs approach and present details of the proposed ensemble approach.

4.1. Extreme Learning Machines

Extreme Learning Machines [14] are a family of algorithms designed for fast, random-based training of single-layer feedforward neural networks. In last decades there were significant developments reported on methods designed for training accurate neural classifiers [22]. However, most of these approaches suffered from the extended computational time required for effective execution and a large number of parameters to be set. ELMs are one of recently emerging trends in neural-based classification that aims at alleviating the training complexities of its predecessor methods by using random weights assigned to hidden layer in a neural network. One must note here that despite the emerging popularity of ELMs-based approaches this concept can be traced further down in the literature to the proposals of Randomized Neural Networks [36] and Random Vector Functional Link [32].

Let us describe now the basic concept of ELMs. We assume that we have n labeled objects described by d features and a set of M labels. A single-layer feedforward neural network with N hidden neurons can be described by the following equation:

$$\mathbf{y} = \sum_{i=1}^{N} \mathbf{B}_i f\left(\mathbf{w}_i \cdot \mathbf{x} + b_i\right), \tag{3}$$

where $f\,()$ is the activation function, $\mathbf{x}$ is the analyzed object, $\mathbf{w}_i$ are the input weights for ith hidden neuron, b_i is the bias of ith hidden neuron and $\mathbf{B}_i$ are weights assigned to outputs.

This equation with respect to all n points can be written in matrix form:

$$\mathbf{Y} = \mathbf{HB}, \tag{4}$$

where $\mathbf{H}$ is the matrix consisting of outputs of hidden layer for each input object:

$$\mathbf{H} = \begin{pmatrix} f(\mathbf{w}_1 \cdot \mathbf{x_1} + b_1) & f(\mathbf{w}_2 \cdot \mathbf{x_1} + b_2) & \cdots & f(\mathbf{w}_N \cdot \mathbf{x_1} + b_N) \\ f(\mathbf{w}_1 \cdot \mathbf{x_2} + b_1) & f(\mathbf{w}_2 \cdot \mathbf{x_2} + b_2) & \cdots & f(\mathbf{w}_N \cdot \mathbf{x_2} + b_N) \\ \vdots & \vdots & \ddots & \vdots \\ f(\mathbf{w}_1 \cdot \mathbf{x_n} + b_1) & f(\mathbf{w}_2 \cdot \mathbf{x_n} + b_2) & \cdots & f(\mathbf{w}_N \cdot \mathbf{x_n} + b_N) \end{pmatrix}, \tag{5}$$

and $\mathbf{B} = (\mathbf{B}_1, \mathbf{B}_2, \ldots \mathbf{B}_N)^T$ and $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots \mathbf{y}_n)$. To calculate the output weights $\mathbf{B}$ is to compute the Moore–Penrose generalized inverse of the matrix $\mathbf{H}$, which we denote as $\mathbf{H}^{-1}$.

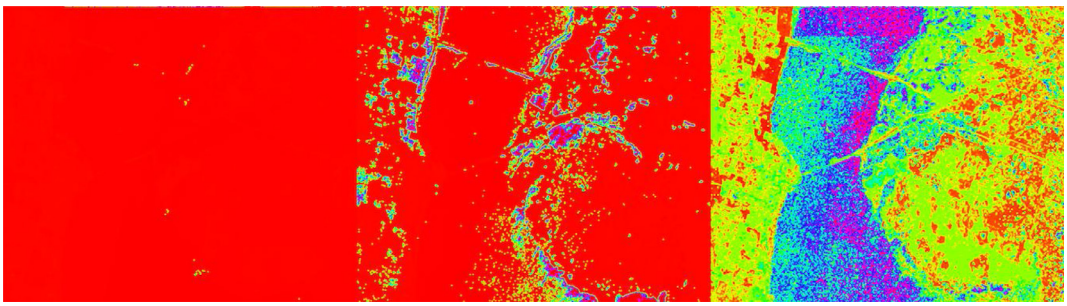


Fig. 3. Color visualization of example metrics.

Table 1
Statistical features implemented in algorithm.

Abbr.	Description
hsvr	Red channel from pseudocolor HSV2RGB conversion.
hsvg	Green channel from pseudocolor HSV2RGB conversion.
hsvb	Blue channel from pseudocolor HSV2RGB conversion.
min	Lowest value in signature.
min_idx	Index of lowest value in signature.
max	Highest value in signature.
max_idx	Index of highest value in signature.
mean	Mean value of signature.
median	Median value of signature.
maxmin	Difference between highest and lowest value in signature
maxmin_dist	Distance between indexes of highest and lowest value in signature.
std	Standard deviation from set of signature values.
var	Variance from set of signature values.
mode	Mode of quantified set of signature values.

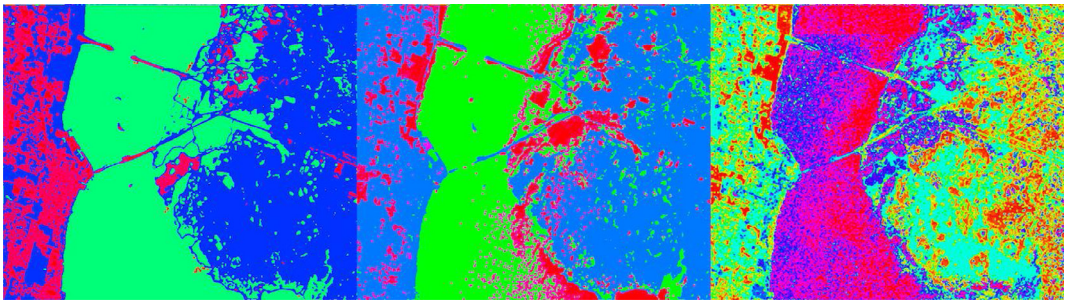


Fig. 4. Color visualization of example metrics after anisotropic diffusion, normalization and histogram equalization.

Basic ELM algorithm proceeds in three main steps:

1. Generate randomly the bias matrix $\mathbf{b} = (\mathbf{b}_1, \mathbf{b}_2, \ldots \mathbf{b}_N)^T$ and weight matrix $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots \mathbf{w}_N)^T$.
2. Calculate $\mathbf{H}$ according to the Eq. (5).
3. Calculate the matrix of output weights $\mathbf{B} = \mathbf{H}^{-1}\mathbf{Y}$

However, there is a need for regularization in ELMs, which was reported as one of the crucial factors affecting their performance. To obtain it we can use an orthogonal projection to get the Moore–Penrose pseudoinverse of $\mathbf{H}$:

$$\mathbf{H}^{-1^*} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T \tag{6}$$

where $\mathbf{H}^T$ is transposed matrix $\mathbf{H}$. This allows use to add a ridge parameter $\frac{1}{\lambda}$ to the diagonal of $(\mathbf{H}^T\mathbf{H})$. This is known as ridge-regression regularization approach [7] that results in a more stable solution. After applying this you calculate the matrix of output weights in step 3 as follows:

$$\mathbf{B} = \left(\frac{\mathbf{I}}{\lambda} + \mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{Y}, \tag{7}$$

where $\mathbf{I}$ is an identity matrix of equal size to $\mathbf{H}$.

4.2. Proposed ensemble architecture

ELMs can be considered as unstable classifiers due to their random nature. As the entire learning process relies on the randomly set weights in the first layer one cannot assure that given model will return highly efficient performance for every initialization. Therefore ensemble learning paradigm [42] started to attract the attention of ELMs community in recent years [3,8,29].

Most of the solutions proposed in this area uses the random generation of input weights as a diversification procedure to obtain a pool of base classifiers. This solution is based on the concept that different random initializations of ELMs will be sufficient to provide mutually complementary classifiers and that this will reduce the probability of using a weak model for classification. However, one may discuss the efficacy of such a method, as methods based on varying the input [19] or output [17] spaces were reported to deliver superior performance in varying multi-class scenarios.

Ensembles of ELMs use mainly voting procedures to combine individual outputs of base classifiers, usually assuming that each base classifier is equally important to the final decision making process (majority voting approach) [9]. One should note that such combination methods cannot take advantage of local specialization

of its base classifiers and do not assume a varied quality of its base members.

In this paper, we address these two important issues in forming ensembles of ELMs: how to create a pool of base classifiers with high individual quality and mutual diversity, and how to combine their individual outputs in the most efficient manner.

4.2.1. Forming pool of ELM classifiers

We propose to investigate the possibility of constructing ELMs committees on the basis of Random Subspace method (RSM) [18].

This method assumes that in the training set $\mathcal{TS}$ we have at our disposal n objects, where x_j is the jth training sample described as a d-dimensional feature vector in given feature space $\mathcal{F}$. Our aim is to construct an ensemble consisting of L classifiers. In RSM each base classifier is constructed using r features, where $r < d$ and features in r are selected randomly from $\mathcal{F}$.

One may see that RSM allows us to train a given number of classifiers, here each is based on a randomly reduced feature space. We assume that all of feature subspaces are of identical size r and that there is a possible overlap between these subspaces.

RSM is especially efficient for high-dimensional data, where it brings a benefit of both diversification of the committee members and simplification of their individual training procedures (as each base classifier work in a reduced space). However there is no clear indicator how many classifiers we should construct using RSM, therefore often the overproduce-and-select approach is used [23].

RSM seems as a highly attractive method for hyperspectral data analysis, as here we deal with high-dimensional datasets (equivalent to the number of bands used) [43]. However high number of features can on one hand be beneficial to RSM (higher potential for diversification), but on the other will lead to very high number of classifiers being trained to actually cover the entire original feature space and to obtain good accuracy.

Therefore, we propose to combine RSM method with our statistical features described in Section 3. They allow to obtain a highly compressed feature space, extracting 14 different statistical channels. Such a new feature space offers a more compact representation, while still being able to benefit from RSM. Below we present justification for a good behavior of such a reduced space as an input for RSM.

The first issue that must be taken into consideration when designing a RSM-based ensemble is the complete coverage of the original feature space. That is a situation in which every single original feature is used by at least one classifier in the ensemble. As RSM does not rank the importance of features, then it should use all of available information to prevent when one or more features are randomly discarded and actually never used in RSM. One may calculate the probability of complete coverage of given RSM-based ensemble as follows:

$$P(\text{coverage}) = 1 - \left(1 - \frac{r}{d}\right)^{L}. \tag{8}$$

Here we can see that probability of full coverage decreases with the growing original space dimensional d and increases with the size of the ensemble L. This shows that for high-dimensional data we will require large ensembles when forming them on the basis of RSM.

Second issue important for RSM is the diversity of base classifiers. As we create each feature subspace in a random way there is some chance that certain classifiers will be trained on identical or fairly similar set of features. This of course does not contribute to the efficacy of ensemble being constructed, but only increases it overall computational complexity. One may calculate the probability of RSM-based ensemble consisting of classifiers with nonidentical subset of features as:

$$P(\text{non–id}) = \left(1 - \frac{1}{\binom{d}{r}}\right)^{L(L-1)/2}. \tag{9}$$

Here we can see that the probability of having a pool of non-identical classifiers in RSM increases with the dimensionality d of the original feature space and decreases with the size of the ensemble L.

Therefore, one can see that these two goals are contradictory with the respect to the size of the ensemble.

We propose to create compact ensembles with RSM method based on our extracted statistical features. When transforming the original feature space into 14-dimensional one it is easier to obtain a small ensemble with full coverage, which at the same time increases the probability of classifiers within it being non-identical.

To further boost the quality of proposed compact ensemble we introduce a trained classifier combination approach for ELMs.

4.2.2. Trained combination of ELMs

ELMs output continuous values for each of classes being considered. Therefore, we may consider such outputs as support values in form $F_m(x)$ which represents classifiers' support that object x belongs to m-th class. According to this the final class outputted by a single ELM classifier will be established according to maximum rule (winner-takes-all).

We propose to consider a weighted classifier combination based continuous outputs of ELMs for each of considered classes.

Assume that we have a pool of L classifiers $\Pi = \{\Psi^{(1)}, \Psi^{(2)}, \ldots, \Psi^{(L)}\}$. For a given object x, each individual classifier decides whether it belongs to class $m \in \mathcal{M} = \{1, \ldots, M\}$ based on the values of discriminants. Let $F_m^{(l)}(x)$ denote a function that is assigned to class m for a given value of x and that is used by the lth classifier Ψ^l. The combined classifier Ψ uses the following decision rule [20]:

$$\Psi(x) = m \iff \hat{F}_m(x) = \max_{k \in \mathcal{M}} \hat{F}_k(x). \tag{10}$$

There is a number of proposal on how to assign weights to classifiers. We propose to use an approach in which weights dependent on the classifier and class number: Weight w_m^l is assigned to the lth classifier and the ith class:

$$\hat{F}_m(x) = \sum_{l=1}^{L} w_m^l F_m^l(x). \tag{11}$$

with the respect to constraint:

$$\sum_{l=1}^{L} w_m^l = 1. \tag{12}$$

Here, the given classifier weights assigned to different classes may differ, which allows us to obtain a highly flexible ensemble structure. This way we can exploit the local competencies of each base ELM classifier. A single ELM may have assigned high weights for classes which are recognized accurately by it and low weights for classes in which it is deemed as non-competent. This is in line with the idea of ELMs ensembles, as we may obtain different base models due to the random initialization process. Additionally, it may counter the drawbacks of RSM, as it controls the degree of importance of each base ELM and can reduce the negative effects produced by base models trained on weak feature subspaces.

We need an efficient method to compute the weights assigned to each class and classifier, thus obtaining a trained combiner. We propose to use a highly efficient perceptron-based combiner. Here we delegate a single perceptron for each class as an aggregation function, which may be trained with any standard procedure used

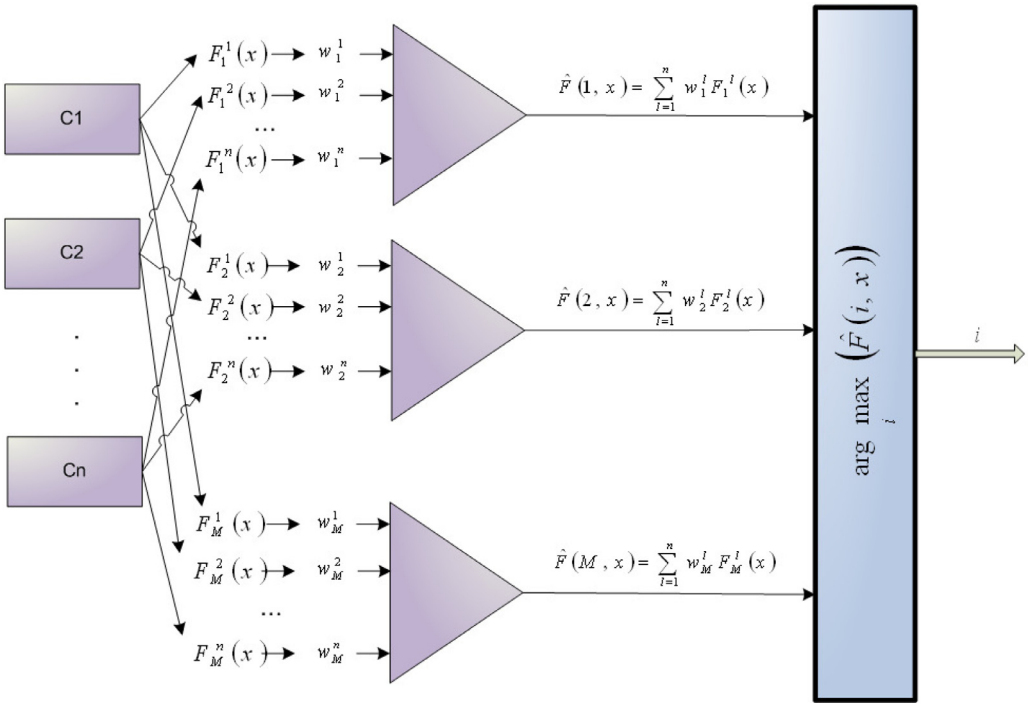


Fig. 5. The idea of the trained combiner, which is a linear combination of the support functions returned by the individual classifiers. It is implemented as an one-layer perceptron, where one perceptron fuser is constructed for each of the classes under consideration.

in neural network learning. The input weights established during the learning process are then used as the weights assigned to each of the base classifiers. The implementation of the proposed combiner is presented in Fig. 5.

5. Experimental study

The experimental study was designed to provide answers to the three following questions:

- Is the proposed 14-channel statistical representation superior to using pixel-based one?
- Is there any benefit from using RSM-based ensembles of ELMs.
- Is the proposed trained combiner more efficient for combination of ELMs than popular voting approach?

In the following subsections we will present details about datasets used, set-up of our experiments, obtained results and their meaning.

5.1. Datasets

In experiments we are using hyperspectral imaging database provided by Group of the Computational Intelligence from *Universidad del Pais Vasco* (UPV/EHU).[2] It consists of seven images described by ground truth maps.

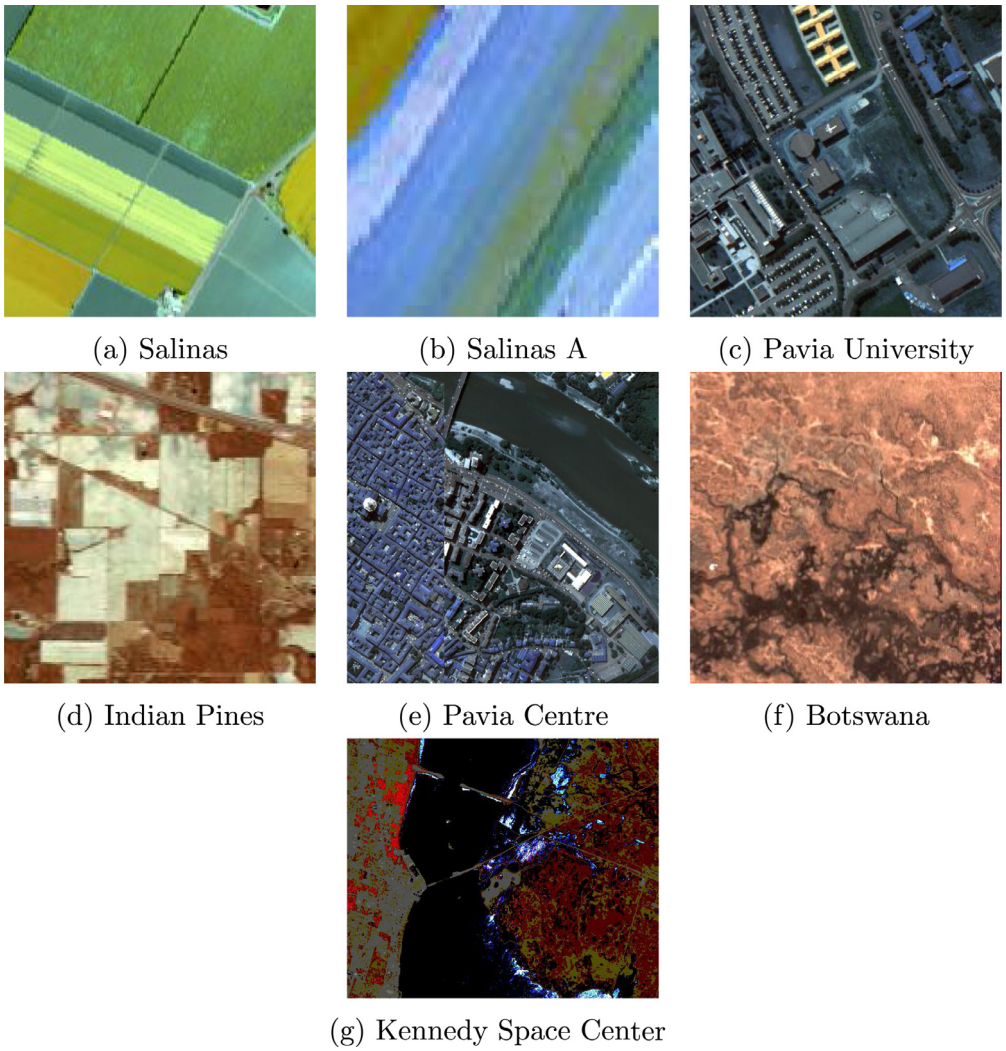
- *Salinas* scene, collected by the AVIRIS sensor over Salinas Valley, California, with high spatial resolution (3.7-meter pixels). It includes vegetables, bare soils, and vineyard fields.
- *Salinas A*, which is an small sub-scene of Salinas image.
- *Indian Pines*, gathered by AVIRIS sensor over the Indian Pines test site in North-western Indiana. Scene contains two-thirds agriculture, and one-third forest or other natural perennial vegetation. There are two major dual lane highways, a rail line, as well as low density housing, other built structures, and smaller roads. Since the scene is taken in June some of the crops present, corn, soybeans, are in early stages of growth with less than 5% coverage.
- *Pavia Centre* and *Pavia University*, acquired by the ROSIS sensor during a flight campaign over Pavia, nothern Italy. The geometric resolution is 1.3 m. Pavia scenes were provided by Prof. Paolo Gamba from the Telecommunications and Remote Sensing Laboratory, Pavia university (Italy).
- *Botswana*, acquired by the NASA EO-1 over the Okavango Delta, Botswana in 2001–2004. The Hyperion sensor on EO-1 acquires data at 30 m pixel resolution over a 7.7 km strip in 242 bands covering the 400–2500 nm portion of the spectrum in 10 nm windows. Preprocessing of the data was performed by the UT Center for Space Research to mitigate the effects of bad detectors, inter-detector miscalibration, and intermittent anomalies. Identified classes are representing the land cover types in seasonal swamps, occasional swamps, and drier woodlands located in the distal portion of the Delta.
- *Kennedy Space Center* (KSC), acquired by AVIRIS sensor over the Kennedy Space Center, Florida, on March 23, 1996. Data, acquired from an altitude of approximately 20 km, have a spatial resolution of 18 m. Training data were selected using land cover maps derived from color infrared photography provided by the Kennedy Space Center and Landsat Thematic Mapper (TM) imagery. The vegetation classification scheme was developed by KSC personnel in an effort to define functional types that are discernable at the spatial resolution of Landsat. Discrimination of land cover for this environment is difficult due to the similarity of spectral signatures for certain vegetation types.

Detailed informations about images are included in Table 2, and cropped previews are shown in Fig. 6.

5.2. Set-up

We propose to compare the proposed method with widely used single-model ELMs and their voting ensembles. Additionally, we

[2] http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes



(a) Salinas (b) Salinas A (c) Pavia University

(d) Indian Pines (e) Pavia Centre (f) Botswana

(g) Kennedy Space Center

Fig. 6. Cropped preview of datasets.

Table 2
Parameters of datasets.

Name	Spatial res.			Depth	Classes	Sensor
Salinas	512	×	217	224	16	AVIRIS
Salinas A	86	×	83	224	6	AVIRIS
Indian Pines	145	×	145	224	16	AVIRIS
Pavia University	610	×	610	103	9	ROSIS
Pavia Centre	1096	×	1096	102	9	ROSIS
Botswana	256	×	1476	242	14	Hyperion
KSC	614	×	512	224	14	AVIRIS

present results obtained by Support Vector Machine (SVM) [5] in one-versus-all (OVA) and one-versus-one (OVO) settings [16].

We use a 5x2 fold CV combined F-test [2] for simultaneous training/testing and pairwise statistical analysis. It repeats two times five-fold cross-validation. The combined *F*-test is conducted by comparison of all versus all. As a test score the probability of rejecting the null hypothesis is adopted, i.e., that classifiers have the same error rates. As an alternative hypothesis, it is conjectured that tested classifiers have different error rates. A small difference in the error rate implies that the different algorithms construct two similar classifiers with similar error rates; thus, the hypothesis should not be rejected. For a large difference, the classifiers have different error rates and the hypothesis should be rejected.

Table 3
Details of classifier parameters used in the experiments. Presented are ranges of parameter values. The final values were established using internal 3 fold CV.

Algorithm	Parameters
ELM	No. of hidden neurons $\in$ [10;100] activation function = sigmoid $\lambda = 10$
SVM	$C \in$ [0.1;1.0] Tolerance parameter = 0.001 $\epsilon \in$ [1.0E−12;1.0E−8] Kernel = polynomial Polynomial degree $\in$ [1;3] OVA = maximum confidence strategy OVO = weighted voting strategy
RSM	No. of subspaces $\in$ [10;50] Size of subspaces = $\lfloor 0.4d \rfloor$
Perceptron combiner	Activation function = sigmoid Learning rule = MADALINE

For the combiner training purposes we utilize 10% of the training data.

Classifier parameters are optimized using internal 3 fold CV. Details regarding their parameter setting are presented in Table 3.

Table 4
Accuracies (%) of examined methods in hyperspectral image classification task. Small symbols under each result indicate the indexes of methods from which the considered one was statistically significantly better according to the combined 5x2 CV *F*-test.

Dataset	Pixel-based					Statistical-based		
	SVM^1_{OVA}	SVM^2_{OVO}	ELM^3	$RSM–ELM^4_{vot}$	$RSM–ELM^5_{per}$	ELM^6	$RSM–ELM^7_{vot}$	$RSM–ELM^8_{per}$
Salinas	68.83	73.04	70.46	72.98	74.40	71.03 1	73.45 1, 3	**76.92** *ALL*
Salinas A	98.20	**99.64**	97.61	98.04	98.51	97.92 –	98.45 3	99.02 3, 4, 6
Pavia U	88.62	90.76	88.93	91.58	93.99	90.93 1, 3	93.64 1, 2, 3, 4	**96.51** *ALL*
I Pines	61.94	64.73	62.86	63.34	65.83	65.18 1, 3, 4	67.06 1, 2, 3, 4, 5, 6	**68.94** *ALL*
Pavia C	89.67	90.70	90.18	91.18	93.65	93.24 1, 2, 3, 4	95.38 1, 2, 3, 4, 5, 6	**96.90** *ALL*
Botswana	96.04	**97.98**	93.18	94.14	94.97	93.21 –	94.09 –	94.91 3, 6
KSC	86.16	90.88	87.59	90.27	92.60	87.97 1	90.81 1, 3, 6	**92.89** 1, 2, 3, 4, 6, 7

Table 5
Average ensemble sizes (rounded) with standard deviations for examined ELMs committees.

Dataset	Pixel-based		Statistical-based	
	$RSM–ELM_{vot}$	$RSM–ELM_{per}$	$RSM–ELM_{vot}$	$RSM–ELM_{per}$
Salinas	36 ± 4.78	29 ± 3.18	27 ± 3.46	20 ± 2.66
Salinas A	25 ± 5.32	20 ± 2.74	24 ± 5.16	20 ± 2.70
Pavia U	17 ± 3.60	16 ± 1.72	15 ± 2.62	14 ± 1.18
I Pines	38 ± 5.91	30 ± 2.99	34 ± 3.58	26 ± 1.98
Pavia C	43 ± 6.71	35 ± 3.88	27 ± 6.02	23 ± 2.63
Botswana	31 ± 4.52	28 ± 3.81	32 ± 4.74	28 ± 3.70
KSC	33 ± 6.92	27 ± 2.89	31 ± 5.34	21 ± 2.02

5.3. Results and discussion

Detailed experimental results with respect to obtained accuracy and statistical significance are depicted in Table 4, while Table 5 presents the averaged sizes of formed ensembles.

Let us now take a closer look on the obtained results.

When comparing data representations approaches we can see a significant improvement obtained when using the proposed statistical-based feature extraction. Regardless of the type of classifier used (single or ensemble models) we can observe a significant accuracy gain in five datasets. For the remaining two (Salinas A and Botswana) results were similar to using full feature space. However, we must note that for all seven datasets using the proposed feature set did not lead to a decrease of classifiers performance. Thus we are able to conclude that the proposed approach offers an effective low-dimensional embedding that reduces the complexity of trained classifiers and offers highly discriminating features that can be used as an efficient input regardless of the base classifier selected.

When taking into account used classifiers we can see that SVM in OVO mode always outperforms single ELM. This can be explained by the fact, that while ELM tries to fit a single model for multi-class problem, SVM actually trains a number of binary classifiers on decomposed pairwise subtasks. Therefore, it can offer superior performance for hyperspectral data with high number of classes. This situation changes however when using ensemble learning. Ensembles of ELM, regardless of their combination method, offer significantly improved performance and are able to outperform SVMs. This shows that our Random Subspace-based ensemble has two beneficial properties: by using several models we reduce the variance caused by randomized neural network architectures and by training each model on a subset of features we further increase the ensemble diversity. Both these factors directly translate to improved accuracy.

When comparing combination methods we observe that the proposed trained combiner is able to greatly improve the efficacy of Extreme Learning Ensembles. By using a weighted combination based on support functions and assigning weights for each classifier and class we are able to vary the importance of each base model and exploit their local competencies. This offers much more flexibility than using voting strategy (which is popular for combining ELMs) as we do not assume that each classifier is equally competent for each of classes. This allows us to further counter the randomness embedded in ELMs training procedure and Random Subspaces method by reducing the importance of weaker or non-diverse models.

Finally, let us analyze the obtained ensemble sizes (please refer to Table 5). We can see that using our proposed feature extraction method leads to smaller ensembles with lower variance, as by using smaller feature space we reduce the number of classifiers needed for full coverage. Additionally, the usage of trained combiner allowed to further reduce the needed number of base classifiers, as we are now able to much more efficiently exploit the given set of learners than in case of voting procedures (where due to their randomness and equal importance we needed larger committees). This also leads to more stable ensembles, as variance in their size over folds of CV is greatly reduced.

6. Conclusions and future works

In this paper, we have addressed the problem of hyperspectral data classification from the perspective of data representation and learning schemes. A new feature extraction method based on statistical channels was proposed. It allowed for a low-dimensional embedding of the original feature space, thus reducing he complexity of analyzed data. We showed how to compute 14 diverse metrics from any hyperspectral image and process them to improve their discriminatory power.

On the basis of these features we proposed to form a novel ensemble of randomized neural networks. It used Extreme Learning Machines as base classifier and used Random Subspaces method to improve the diversity among ensemble members. This architecture was augmented with a trained classifier combination step that used a perceptron-based training method to compute weights. These were assigned to each classifier and class individually, thus resulting in a flexible exploitation of local competencies of base classifiers and efficient multi-class pattern recognition. Additionally, we showed that such an ensemble method combined with proposed feature representation not only boosts the accuracy, but also leads to forming more sparse committees.

Obtained results encourage us to work with statistical-based data representation and Extreme Learning Ensembles. In future we plan to apply our framework to semi-supervised scenario with limited access to class labels and self-labeling mechanisms.

Acknowledgment

This was supported by the Polish National Science Center under the grant no. DEC-2013/09/B/ST6/02264.

All experiments were carried out using computer equipment sponsored by EC under FP7, Coordination and Support Action, Grant Agreement Number 316097, ENGINE - European Research Centre of Network Intelligence for Innovation Enhancement (http://engine.pwr.wroc.pl/).

References

[1] A. Agarwal, T. El-Ghazawi, H. El-Askary, J. Le-Moigne, Efficient hierarchical-PCA dimension reduction for hyperspectral imagery, in: Proceedings of the IEEE International Symposium on Signal Processing and Information Technology, 2007, pp. 353–356, doi:10.1109/ISSPIT.2007.4458191.
[2] E. Alpaydin, Combined 5 x 2cv f test for comparing supervised classification learning algorithms, Neural Comput. 11 (8) (1999) 1885–1892.
[3] B. Ayerdi, M. Graña, Hybrid extreme rotation forest, Neural Netw. 52 (2014) 33–42.
[4] B. Ayerdi, M. Graña, Hyperspectral image nonlinear unmixing and reconstruction by ELM regression ensemble, Neurocomputing 174 (2016) 299–309.
[5] Y. Bazi, F. Melgani, Toward an optimal SVM classification system for hyperspectral remote sensing images, IEEE Trans. Geosci. Remote Sens. 44 (11–2) (2006) 3374–3385.
[6] K.P. Bennett, A. Demiriz, Semi-supervised support vector machines, in: Proceedings of the Advances in Neural Information Processing Systems, MIT Press, 1998, pp. 368–374.
[7] P. Buteneers, K. Caluwaerts, J. Dambre, D. Verstraeten, B. Schrauwen, Optimized parameter search for large datasets of the regularization parameter and feature selection for ridge regression, Neural Process. Lett. 38 (3) (2013) 403–416.
[8] J. Cao, S. Kwong, R. Wang, X. Li, K. Li, X. Kong, Class-specific soft voting based multiple extreme learning machines ensemble, Neurocomputing 149 (2015) 275–284.
[9] J. Cao, Z. Lin, G. Huang, N. Liu, Voting based extreme learning machine, Inf. Sci. 185 (1) (2012) 66–77.
[10] R. Clemen, Combining forecasts: a review and annotated bibliography, Int. J. Forecast. 5 (4) (1989) 559–583.
[11] Q. Dai, A competitive ensemble pruning approach based on cross-validation technique, Knowl. Based Syst. 37 (2013) 394–414.
[12] E.R. Davies, Machine Vision: Theory, Algorithms, Practicalities, Elsevier, 2004.
[13] T. Dietterich, Ensemble methods in machine learning, in: Multiple Classifier Systems, in: Lecture Notes in Computer Science, 1857, Springer, Berlin, Heidelberg, 2000, pp. 1–15.
[14] S. Ding, H. Zhao, Y. Zhang, X. Xu, R. Nie, Extreme learning machine: algorithm, theory and applications, Artif. Intell. Rev. 44 (1) (2015) 103–115.
[15] J. Durand, Y. Kerr, An improved decorrelation method for the efficient display of multispectral data, IEEE Trans. Geosci. Remote Sens. 27 (5) (1989) 611–619, doi:10.1109/TGRS.1989.35944.
[16] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, An overview of ensemble methods for binary classifiers in multi-class problems: experimental study on one-vs-one and one-vs-all schemes, Pattern Recognit. 44 (8) (2011) 1761–1776.
[17] M. Galar, A. Fernández, E. Barrenechea, F. Herrera, DRCW-OVO: distance-based relative competence weighting combination for one-vs-one strategy in multi–class problems, Pattern Recognit. 48 (1) (2015) 28–42.
[18] T.K. Ho, The random subspace method for constructing decision forests, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 832–844.
[19] K. Jackowski, B. Krawczyk, M. Woźniak, Improved adaptive splitting and selection: the hybrid training method of a classifier based on a feature space partitioning, Int. J. Neural Syst. 24 (3) (2014).
[20] R.A. Jacobs, Methods for combining experts' probability assessments, Neural Comput. 7 (5) (1995) 867–888.
[21] A. Jain, R. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Trans. Pattern Anal. Mach. Intell. 22 (1) (2000) 4–37, doi:10.1109/34.824819.
[22] L.C. Jain, M. Seera, C.P. Lim, P. Balasubramaniam, A review of online learning in supervised neural networks, Neural Comput. Appl. 25 (3–4) (2014) 491–509.
[23] A.H. Ko, R. Sabourin, L.E.S. de Oliveira, A. de Souza Britto Jr., The implication of data diversity for a classifier-free ensemble selection in random subspaces, in: Proceedings of the Ninteenthth International Conference on Pattern Recognition, Tampa, Florida, USA, 2008, pp. 1–5.
[24] B. Krawczyk, P. Ksieniewicz, M. Woźniak, Hyperspectral image analysis based on color channels and ensemble classifier, in: Proceedings of the Ninth International Conference on Hybrid Artificial Intelligence Systems, in: HAIS 2014, 8480, Springer-Verlag New York, Inc., New York, NY, USA, 2014, pp. 274–284.
[25] L. Kuncheva, L. Jain, Designing classifier fusion systems by genetic algorithms, IEEE Trans. Evolut. Comput. 4 (4) (2000) 327–336.
[26] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, 2004.
[27] J. Li, J.M. Bioucas-Dias, A. Plaza, Spectral-spatial classification of hyperspectral data using loopy belief propagation and active learning, IEEE Trans. Geosci. Remote Sens. 51 (2) (2013) 844–856.
[28] W. Li, C. Chen, H. Su, Q. Du, Local binary patterns and extreme learning machine for hyperspectral imagery classification, IEEE Trans. Geosci. Remote Sens. 53 (7) (2015) 3681–3693.
[29] N. Liu, H. Wang, Ensemble based extreme learning machine, IEEE Signal Process. Lett. 17 (8) (2010) 754–757.
[30] G.L. Marcialis and F.Roli. Fusion of face recognition algorithms for video-based surveillance systems. G.L. Foresti, C. Regazzoni, P. Varshney Eds, 235--250, 2003.
[31] R. Moreno, F. Corona, A. Lendasse, M. Graña, L.S. Galvão, Extreme learning machines for soybean classification in remote sensing hyperspectral images, Neurocomputing 128 (2014) 207–216.
[32] Y. Pao, G.H. Park, D.J. Sobajic, Learning and generalization characteristics of the random vector functional-link net, Neurocomputing 6 (2) (1994) 163–180.
[33] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Trans. Pattern Anal. Mach. Intell. 12 (7) (1990) 629–639, doi:10.1109/34.56205.
[34] J. Quinlan, C4.5: programs for machine learning, Morgan Kaufmann Series in Machine Learning, Morgan Kaufmann Publishers, 1993.
[35] S. Rajan, J. Ghosh, M.M. Crawford, An active learning approach to hyperspectral data classification., IEEE Trans. Geosci. Remote Sens. 46 (4) (2008) 1231–1242.
[36] W. Schmidt, M. Kraaijveld, R. Duin, Feedforward neural networks with random weights, in: Proceedings of the Eleventh IAPR International Conference on Pattern Recognition, II, 1992, pp. 1–4. Conference B: Pattern Recognition Methodology and Systems, Proceedings
[37] G. SVAETICHIN, Spectral response curves from single cones, Acta Physiol. Scand. 39 (134) (1956) 17–46. PMID: 13444020
[38] J. Tyo, A. Konsolakis, D. Diersen, R. Olsen, Principal-components-based display strategy for spectral imagery, IEEE Trans. Geosci. Remote Sens. 41 (3) (2003) 708–718, doi:10.1109/TGRS.2003.808879.
[39] M. van Erp, L. Vuurpijl, L. Schomaker, An overview and comparison of voting methods for pattern recognition, in: Proceedings. Eighth International Workshop on Frontiers in Handwriting Recognition, 2002, pp. 195–200.
[40] T. Wilk, M. Woźniak, Complexity and multithreaded implementation analysis of one class-classifiers fuzzy combiner, in: E. Corchado, M. Kurzynski, M. Wozniak (Eds.), Hybrid Artificial Intelligent Systems, Lecture Notes in Computer Science, 6679, Springer, Berlin, Heidelberg, 2011, pp. 237–244.
[41] D.H. Wolpert, The supervised learning no-free-lunch theorems, in: Proceedings of the Sixth Online World Conference on Soft Computing in Industrial Applications, 2001, pp. 25–42.
[42] M. Woźniak, M. Graña, E. Corchado, A survey of multiple classifier systems as hybrid systems, Inf. Fus. 16 (2014) 3–17.
[43] J. Xia, M.D. Mura, J. Chanussot, P. Du, X. He, Random subspace ensembles for hyperspectral image classification with extended morphological attribute profiles, IEEE Trans. Geosci. Remote Sens. 53 (9) (2015) 4768–4786.

Paweł Ksieniewicz is a research assistant at Wroclaw University of Technology, where he achieved M.Sc. degree in 2013 and Ph.D. degree in 2017. His research focuses on multidimensional data representation and image processing. Most of his papers concerns the hyperspectral imaging in context of data segmentation and visualization.

Bartosz Krawczyk is an assistant professor in the Department of Computer Science, Virginia Commonwealth University, Richmond VA, USA, where he heads the Machine Learning and Stream Mining Lab. He obtained his MSc and PhD degrees from Wroclaw University of Science and Technology, Wroclaw, Poland, in 2012 and 2015 respectively. His research is focused on machine learning, data streams, ensemble learning, class imbalance, one-class classifiers, and interdisciplinary applications of these methods. He has authored 35+ international journal papers and 80+ contributions to conferences. Dr Krawczyk was awarded with numerous prestigious awards for his scientific achievements like IEEE Richard Merwin Scholarship and IEEE Outstanding Leadership Award among others. He served as a Guest Editor in four journal special issues and as a chair of ten special session and workshops. He is a member of Program Committee for over 40 international conferences and a reviewer for 30 journals.

Micha Woźniak is a professor of computer science at the Department of Systems and Computer Networks, Wroclaw University of Technology, Poland. He received M.Sc. degree in biomedical engineering from the Wroclaw University of Technology in 1992, and Ph.D. and D.Sc. (habilitation) degrees in computer science in 1996 and 2007, respectively, from the same university. His research focuses on compound classification methods, hybrid artificial intelligence and medical informatics. Prof. Woźniak has published over 260 papers and three books. His recent one Hybrid classifiers: Method of Data, Knowledge, and Data Hybridization was published by Springer in 2014. He has been involved in several research projects related to the above-mentioned topics and has been a consultant of several commercial projects for wellknown Polish companies and public administration. Prof. Woźniak is a senior member of the IEEE and a member of the International Biometric Society.

[C11]
Paweł Ksieniewicz i Michał Woźniak. "Dealing with the task of imbalanced, multidimensional data classification using ensembles of exposers". W: *Proceedings of the First International Workshop on Learning with Imbalanced Domains: Theory and Applications*. Red. Paula Branco Luís Torgo i Nuno Moniz. T. 74. Proceedings of Machine Learning Research. PMLR, 22 Sep 2017, s. 164–175. URL: https://proceedings.mlr.press/v74/ksieniewicz17a.html

Proceedings of Machine Learning Research 74:164–175, 2017 Poster - LIDTA 2017

Dealing with the task of imbalanced, multidimensional data classification using ensembles of *exposers*

Paweł Ksieniewicz PAWEL.KSIENIEWICZ@PWR.EDU.PL
Michał Woźniak MICHAL.WOZNIAK@PWR.EDU.PL
Department of Systems and Computer Networks
Faculty of Electronics
Wroclaw University of Science and Technology

Editors: Luís Torgo, Bartosz Krawczyk, Paula Branco and Nuno Moniz.

Abstract

Recently, the problem of imbalanced data is the focus of intense research of machine learning community. Following work tries to utilize an approach of transforming the data space into another where classification task may become easier. Paper contains a proposition of a tool, based on a photographic metaphor to build a classifier ensemble, combined with a *random subspace* approach. Developed solution is insensitive to a sample size and robust to dimension increase, which allows a regularization of feature space, reducing the impact of biased classes. The proposed approach was evaluated on the basis of the computer experiments carried out on the benchmark and synthetic datasets.

Keywords: curse of dimensionality, imbalance data, random subspace, ensemble classifiers

1. Introduction

If the classes in available dataset are not represented equally, we are encountering a problem of *imbalanced data*. In extreme intensity of this situation, being relatively common, i.e., in the fraud detection problems (Phua et al., 2004) or in the medical data analysis (Mazurowski et al., 2008), where usually occurs a minority report on most important class (a fraud or a sickness), often happens the *accuracy paradox* (Valverde-Albacete and Peláez-Moreno, 2014). It means that for a strongly uneven distribution of classes, higher *accuracy*, being the most common measure of classifier performance in literature (Demsar, 2006), does not indicate the greater discriminative power.

Two most popular approaches (Krawczyk, 2016) to deal with problems caused by *imbalanced data* are (i) *inbuilt mechanisms*, which change the classification rules to enforce a bias toward the minority class using e.g., cost-sensitive approach (Krawczyk et al., 2014) and (ii) *data preprocessing methods*, which modify the data distribution to change the balance between classes. The preprocessing approach uses *oversampling* of a minority class, applied usually when overall number of objects in dataset is relatively low (He and Garcia, 2009) and *undersampling* of a majority class (Japkowicz and Stephen, 2002), popular for large datasets (Liu et al., 2009). Both of them have its flaws. *Oversampling*, especially conducted before *cross-validation* comes with a risk of overfitting, while *undersampling* may lead to significant decrease of classifiers discrimination power due to removal of informative objects.

Despite the risk of *overtraining* and problems to measure the classifiers quality in a cases of *imbalanced datasets*, one of the biggest problems in *pattern recognition* is the *curse of dimensionality* (Bellman, 1961). According to it, the growth of feature vector, causes that the generalizations are becoming exponentially harder. So it is necessary not to forget about the structure of single object itself. While it is a vector of d dimensions, not all of them are equally relevant for *pattern recognition* methods. Some of them, by injecting nothing more than a noise, may have only negative aspect on the quality of our algorithms (Segura et al., 2003). Some of them may turn out to be exploitable only in relation to others (Bishop, 2006). Having all this information in mind, it is important to know about dimensionality reduction techniques.

The *feature selection* aims to choose a subset of original feature set, without changing its values. While it was proved as an useful and effective technique (Liu and Motoda, 2009), it leaves original features untouched, remaining their physical meanings to be still interpretable by a human being (Li and Liu, 2016). This advantage encourages to use in real world applications (Kursa and Rudnicki, 2010), because it causes a loss only of data irrelevant for classification (Rudnicki et al., 2015). The second technique, *feature extraction* (Stapor, 2011), tries to create a function to map d dimensional vector into smaller s dimensional one. Even since we reuse information from even a whole vector, original features are here replaced by their scalar products and we are no longer able to interpret them directly.

While feature selection is rather a discrete process, the optimization of feature extraction methods is quite *smoother* and often leads for a more fitting solution (Koller and Sahami, 1996), by employing information from wider range than only resulting subspace and precisely setting the impact of original features onto the transformation result.

On the other hand, there is no need to hold down only one algorithm into a single problem. We can also compose structures of a multiple classifier system, which combines a set of classifiers to provide a common decision of the *classifier ensemble* (Kuncheva, 2014). The main element of this structure is a *pool of classifiers*. The most important aim of good selection of the member classifiers, is to provide their high diversity, which means that each classifier should make an independent decision (Dietterich, 2000). We can try to ensure it by differencing the input and output data, or by differencing the classifier models. One of the popular methods, is a *random subspace* approach, originally implemented for *decision trees*, which brought the *random forests* (Ho, 1998) and later applied with success also for SVM*'s*, *linear classifiers* or k-NN. It tries to weaken the correlation of classifiers in *ensemble*, by letting each of them learn on a reduced, random subset of features, introducing the approach of random *feature selection*.

The main contributions of this work are:

- the proposition of the exposer – a visualization tool for numerical data distribution, usable as *Exposer Classifier* – the *supervised learning* method to use it in a task of decision making, being the component to build *Exposer Classifier Ensemble* – the *Random Subspace* based approach of classifier ensemble.
- an experimental evaluation of the proposed concept presenting the detailed results that offers an in-depth insight into the importance of selecting proper examples for the oversampling procedure. A dedicated website[1] presents detailed results.

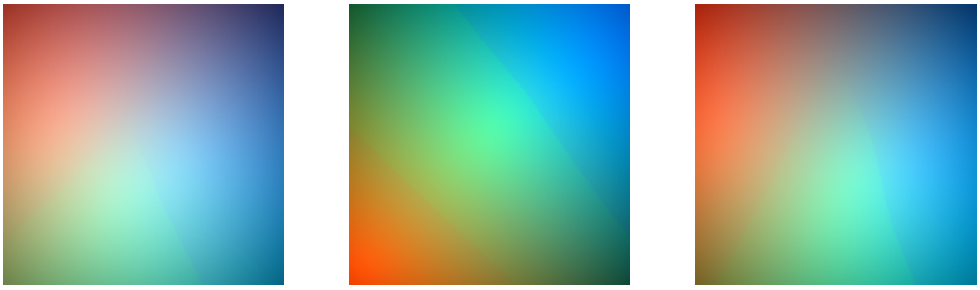
1. http://ksienie.com/ecml17/

- a set of conclusions that will allow to design efficient classifiers for datasets.

2. Exposers

Following section aims to explain the idea of *exposer*, a data representation, based on visualization tools of numerical data distribution, creating, from a regular dataset, the spacial-spectral structure similar to a multi-spectral image (Ksieniewicz et al., 2017). However, *exposer* is not intended to work as a preprocessing tool, but as a classifier itself. Moreover, composing a set of *exposers*, generated on different feature subsets of a same training set, leads to *Exposer Classifier Ensemble* (ECE).

A proposed data structure is drawing from tenets of *histogram* and a *scatter plot*. Like in *histogram*, the range of values is divided into a series of intervals, but like in a *scatter plot*, not the single value, but the combination of them is analyzed. The rule of adjacency is broken here, so each object may fall into more than one of the bins. Exemplary *exposers*, prepared for two-dimensional feature subsets of *iris* dataset are presented in Figure 1.



(*a*) 1:2 (*b*) 3:4 (*c*) 2:3

Figure 1: Exemplary exposers for *iris* dataset. Digits points the indexes of features.

The procedure of creating an *exposer* is inspired by the process of plate light exposure in the chemical photography. Hence, its control parameters are plate grain (*exposer* counterpart of *histograms* bin) and a light dispersion factor (named a *radius*, as a relative width of a bin according to a range of values). Instead of exposing the photographic plate coated with light-sensitive chemicals to the light source, a numerical representation matrix is exposed to the beams projected from the data samples. Procedure *takes a photography* of a subset of features of the data samples, where intensity of a *light* in every point is a density aggregation of the data samples falling in its neighborhood.

Exemplary process of exposing is illustrated in Figure 2. We have a dataset which consists of four objects, each described by four features[2] and assigned to one of three classes (R, G and B). The grain parameter is fixed at 10 and selected subspace uses pair of first and third feature. At the first step, *subspace positioning*, every object in dataset is placed on a two-dimensional grid, divided by 10 in every dimension, according to values of chosen features. The exposer consists of three layers, one per every class in dataset, and every layer

2. To clarify illustration, all the feature values in exemplary dataset are integers in range 0–10. Real implementation normalizes original values as floats in range 0–1.

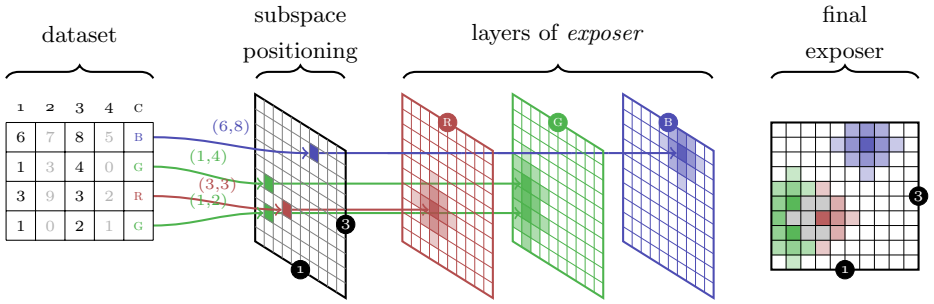


Figure 2: Exposing four samples covering three classes on two-dimensional subspace (1,3).

is influenced only by objects labeled accordingly, by adding the negative distance (where closer means higher value) between positioned centre of object to corresponding cell.

The most important difference between classic photography and *exposers* is a redefinition of the concept of color, being not the classical RGB vector nor a real *spectral signature*. For *exposers* it consists not of classical three spectral channels (Svaetichin, 1956), but of one dimension per class of the dataset. Hence, the representation matrix has as many layers as classes. The representation matrix exposure process sensitizes each layer projecting to it only objects from the corresponding class. There may be assumed, that *exposing* procedure generates some kind of multispectral imaging from the data, which counterpart of *spectral signature* is interpreted as a *support vector* during the classification procedure.

To prepare a mathematical description of *exposer*, let $\mathcal{DS}$ denote a set of n examples, where each of them x_k is represented by the d-dimensional feature vector and its label i_k from the finite set of available labels $\mathcal{M}$.

$$\begin{aligned} \mathcal{X} &\subseteq \Re^d \\ \mathcal{M} &= \{1, 2, \ldots, M\} \\ x_k &= [x_k^{(1)}, x_k^{(2)}, \ldots, x_k^{(d)}], & x_k \in \mathcal{X} \\ i_k &= [i_1, i_2, \ldots, i_n], & i_k \in \mathcal{M} \\ \mathcal{DS} &= \Big\{(x_1, i_1), (x_2, i_2), \ldots, (x_n, i_n)\Big\} \end{aligned} \tag{1}$$

Variety of possible exposers comes from a set Λ of $\binom{d}{s}$ combinations λ_i, where s is the chosen exposer dimension, i.e., number of chosen features.

$$\begin{aligned} \Lambda &= \{\lambda_1, \lambda_2, \ldots, \lambda_L\}, \quad |\Lambda| = L = \tbinom{d}{s} \\ \lambda_i &= [l_1, l_2, \ldots, , l_s], \quad l_j \in \{1, 2, \ldots, d\}, \quad l_1 \neq l_2 \neq \ldots \neq l_s, \end{aligned} \tag{2}$$

Representation of *exposer* ($\mathcal{E}$) is a s-dimensional cube

$$\begin{aligned} \mathcal{E}_m &\in G^s = \underbrace{G \times G \times \ldots \times G}_{s} \\ \mathcal{E} &= \{\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_M\}. \end{aligned} \tag{3}$$

And every point positioned by *loc* gives us a vector of M values v

$$\begin{aligned} \mathcal{E}^{(loc)} &= [v_1, v_2, \ldots, v_M]^T \\ loc &= [loc_1, loc_2, \ldots, loc_s]^T \end{aligned} \tag{4}$$

A single value of *exposer* is a sum of all positive differences between a given radius r and a distance from a real vector of grid cell central point *loc* to the location loc_k of samples in a subspace, labeled accordingly to the mth layer.

$$\begin{aligned} \mathcal{E}_m^{(loc)} &= \textstyle\sum_{k=1}^{n} \Big[d(loc, loc_k) < r \ \wedge \ i_k = m \Big] \cdot \Big(r - d(loc, loc_k) \Big) \\ loc_k &= [x_k^{(\lambda_1)}, x_k^{(\lambda_2)}, \ldots, x_k^{(\lambda_s)}]^T \end{aligned} \tag{5}$$

which is a discretization on the g quants in every spacial dimension created by given λ_i.

According to the photographic metaphor, the previous expression may be imagined as a projection by exposure, where every x_k sample is the *photon* with location described with features chosen by combination λ, affecting the image in r radius.

For the classification task, there is an *exposer* $\mathcal{E}$, exposed using a subspace λ of a *learning set* $\mathcal{LS}$ and a testing sample x_k from a *testing set* $\mathcal{TS}$ to classify.

$$x_k = [x_k^{(1)}, x_k^{(2)}, \ldots, x_k^{(d)}]^T, \qquad x_k \in \mathcal{TS} \tag{6}$$

Classifier Ψ makes a decision on the basis of the class index with the maximum value from a value vector corresponding to the testing sample.

$$\Psi(x_k) = \underset{m \in \mathcal{M}}{argmax} \left(\mathcal{E}_m^{(loc_k)} \right) \tag{7}$$

Increase of the feature vector also rapidly increases a number of possible *exposers* to obtain. It creates a possibility to gather a set of *exposer* classifiers into an ensemble. To establish such ensemble, a prediction procedure needs a little enhancement. There is a testing sample x_k to classify, but this time we have an ensemble of *exposers* (Π) build around the set of combinations Λ', being a subset of all possible combinations

$$\begin{aligned} \Lambda' &\subset \Lambda, & |\Lambda'| = N, \qquad N < L \\ \Pi &= \{\Psi_1, \Psi_2, \ldots, \Psi_N\}, & \Psi : \mathcal{X} \leftarrow \mathcal{M} \end{aligned} \tag{8}$$

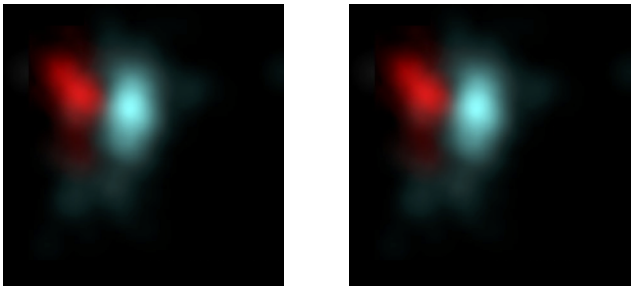
Exposer may be visualized as a regular RGB substitution if number of classes, like in example from Figure 2, matches the number of color channels observable by eyes. To make a color visualization for number of classes other than 3, the *hue, saturation, value* (HSV) (Smith, 1978) interpretation of *exposer* point was proposed. The value (V) is the maximum of $\mathcal{E}^{(loc)}$ vector and the *hue* (H) is an angle of its index normalized to 360°.

$$\begin{aligned} V(\mathcal{E}^{(loc)}) &= max(\mathcal{E}^{(loc)}) \\ H(\mathcal{E}^{(loc)}) &= \underset{i \in \mathcal{M}}{argmax}(\mathcal{E}^{(loc)}) \cdot \frac{360°}{M} \end{aligned} \tag{9}$$

Although the vector $\mathcal{E}^{(loc)}$, like a spectral signature of multi-spectral data, does not point a specific color of a visible spectrum, we can still interpret the grey concealed in it in the same way as in the classic color theory, meaning an equal presence of any value building it. Black corresponds to values close to nothing, while white stands for the maximum values. According to it, *saturation* is defined as a difference between the minimum and maximum of the $\mathcal{E}^{(loc)}$.

$$S(\mathcal{E}^{(loc)}) = max(\mathcal{E}^{(loc)}) - min(\mathcal{E}^{(loc)}) \tag{10}$$

Figure 3 presents exemplary *exposers*, with relatively small radius parameter, based on the chosen feature pairs, acquired for the *yeast3* dataset. The dataset consists of 1483 samples described by 8 features, divided into two classes. The visualization aggregates both layers into one by colour coding each class (*red* – positive, *green* – negative) modulated by the layer intensity.



(*a*) 1:3 (*b*) 1:4

Figure 3: *Exposers* created using two example combinations for *yeast3* dataset.

The left image presents a two feature subspace, where colors are highly mixing. There is a high presence of grays, and *class colors* are faded, so intuitively, this pair of features should provide a weak discrimination power. The image on the right side shows a two feature subspace where mixing of colors is minimal, so it may be assumed, that classes are highly separable in this subspace.

Finally, a three level classifier ensemble is proposed, which idea is depicted in Figure 4. At the lowest level there is a set of monochrome layers, each of which defines a member

classifier, characterized by a combination of features, denoted by λ_i, and a weight Θ_i used to combine its output with the remaining classifiers into an *exposer*. At the top level, member classifiers are combined into an ECE ensemble.

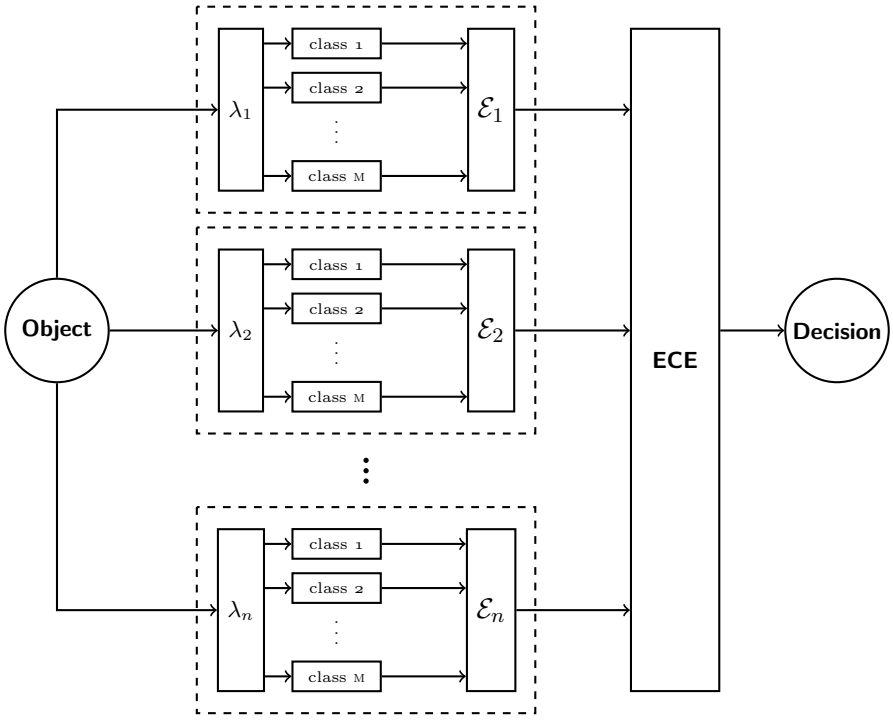


Figure 4: ECE classification diagram

3. Experimental evaluation

Set-up The experimental evaluation of proposed method was realized under *Weles* framework (Ksieniewicz, 2017). Its full code is available to access as a GIT repository on *Github* page of *Department of Systems and Computer Networks*[3] and published in in *Python Package Index*. Quality of all the written *Python* code is statically analyzed using *Code Climate* and is continuously integrated using *Travis* CI. Current quality metrics and test coverages are listed in the homepage of every repository.

For the evaluation of algorithm capabilities, eight benchmark datasets were used. All of them contains imbalanced binary problems. Four of them are coming from UCIMLR (Lichman, 2013) and four are synthetic, generated data with various number of features (from 20 to 200) and fixed 9:1 imbalance ratio. Also five reference classification algorithms were selected. To use them without a need of a new implementation, *Weles* framework was enhanced by the adapter able to wrap classifiers from a popular machine learning library *scikit-learn* (Pedregosa et al., 2011), to encapsulate prepared selection in a separate classes.

3. *https://github.com/w4k2*

The classifiers were configured with most popular parameters, to provide a standard, *state-of-art* pool for proper comparison of proposed method. Prepared datasets and classifiers pool were cross-tested using measure of *balanced accuracy*(Brodersen et al., 2010) (Table 1).

Table 1: Balanced accuracy achieved by chosen reference algorithms and data characteristics.

Dataset	KNN Minimal Distance Classifier	GNB Bayesian Classifier	DTC Rule-Based Classifier	MLP Artificial Neural Network	SVC Support Vector Machine	Characteristics Samples	 Features	 Imbalance ratio
balance	0.768	0.786	0.720	0.635	0.785	625	4	6:1
ionosphere	0.692	0.845	0.795	0.592	0.590	351	34	2:1
wisconsin	0.943	0.954	0.898	0.500	0.946	699	9	2:1
yeast3	0.814	0.575	0.811	0.858	0.500	1484	8	8:1
synthetic20	0.565	0.768	0.702	0.707	0.500	4000	20	9:1
synthetic50	0.548	0.765	0.867	0.500	0.500	4000	50	9:1
synthetic100	0.561	0.767	0.788	0.789	0.500	4000	100	9:1
synthetic200	0.511	0.623	0.881	0.500	0.500	4000	200	9:1

Experiment The experiments were conducted under 5×2 *cross-validation* (Alpaydin, 2014) and presented results are a BAC value. Red color means that the result is statistically significantly better than results presented in black, and statistical significancy was measured using T-test. Last row of result tables will contain the result obtained by the best and the worst reference classifier, extended by a difference between score of ECE and leading reference solution.

Table 2: Experiment configuration table

Fixed parameters	
dimensions	2
approach	random
limit	30
Studied parameters	
grain	$4 - 32$
radius	$.1 - .5$

As a test of ECE, it is necessary to establish the values of parameters of single *exposer* configuration: its *radius* and *grain*, both of which being exponential reason of growth of the computational time needed to process a training set. For datasets with feature vector larger than 8 values, there is no chance to obtain *exposers* for all possible subspaces in a reasonable time. To deal with this problem, a *random subspace* approach was used. The planned experiment uses eight benchmark datasets. For a low computational complexity, only two-dimensional subspaces are considered.

Discussion Comparison of balanced accuracy between ECE and reference classifiers for imbalanced binary problems is presented in Table 3. It is worth observing that the ECE

Table 3: Balanced accuracy achieved on all tested datasets using ECE.

(*a*) balance

Radius	Grain			
	4	8	16	32
0.1	0.500	0.500	0.653	0.699
0.2	0.500	0.653	0.699	0.701
0.3	0.690	0.705	0.704	0.720
0.4	0.690	0.730	0.735	0.736
0.5	0.724	0.732	0.730	0.728

-5.0%, best 0.786 (GNB), worst 0.635 (MLP)

(*b*) ionosphere

Radius	Grain			
	4	8	16	32
0.1	0.500	0.500	0.500	0.863
0.2	0.500	0.763	0.842	0.859
0.3	0.500	0.794	0.863	0.878
0.4	0.500	0.824	0.797	0.840
0.5	0.710	0.740	0.757	0.752

+3.3%, best 0.845 (GNB), worst 0.590 (SVC)

(*c*) wisconsin

0.1	0.500	0.500	0.927	0.957
0.2	0.500	0.955	0.965	0.964
0.3	0.959	0.968	0.969	0.969
0.4	0.958	0.967	0.968	0.970
0.5	0.967	0.966	0.970	0.970

+1.6%, best 0.954 (GNB), worst 0.500 (MLP)

(*d*) yeast3

0.1	0.500	0.500	0.708	0.817
0.2	0.500	0.771	0.869	0.881
0.3	0.832	0.840	0.873	0.880
0.4	0.830	0.848	0.874	0.883
0.5	0.852	0.855	0.884	0.889

+3.1%, best 0.858 (MLP), worst 0.500 (SVC)

(*e*) synthetic20

0.1	0.500	0.500	0.548	0.527
0.2	0.500	0.583	0.696	0.613
0.3	0.642	0.650	0.760	0.818
0.4	0.744	0.839	0.839	0.880
0.5	0.764	0.816	0.838	0.840

+11.2%, best 0.768 (GNB), worst 0.500 (SVC)

(*f*) synthetic50

0.1	0.500	0.500	0.512	0.503
0.2	0.500	0.530	0.615	0.687
0.3	0.583	0.508	0.771	0.699
0.4	0.661	0.802	0.833	0.732
0.5	0.677	0.798	0.764	0.887

+2.0%, best 0.867 (DTC), worst 0.500 (MLP)

(*g*) synthetic100

0.1	0.500	0.500	0.507	0.500
0.2	0.500	0.537	0.621	0.577
0.3	0.560	0.827	0.620	0.868
0.4	0.521	0.843	0.593	0.513
0.5	0.611	0.739	0.873	0.890

+10.1%, best 0.789 (MLP), worst 0.500 (SVC)

(*h*) synthetic200

0.1	0.500	0.500	0.501	0.500
0.2	0.500	0.500	0.510	0.501
0.3	0.533	0.529	0.504	0.662
0.4	0.506	0.661	0.694	0.494
0.5	0.532	0.635	0.602	0.816

-6.5%, best 0.881 (DTC), worst 0.500 (MLP)

often achieves its high effectiveness on relatively low values of tested parameters (.3 radius and grain of 32). However, the hardest, high-dimensional and highly imbalanced problems of *synthetic*20 and *synthetic*50 shows, that further growth of grain parameter may have positive impact on classification.

In most of cases (the only exception is *balance* dataset), proposed solution outperforms all of reference methods, in one case even by over 10%. It is never the worst solution in the competitive pool.

The algorithm, by employing the information of features distribution, takes all the advantages from the characteristics of algorithms similar to *bayesian classifiers*, achieving the best results with imbalanced data from the reference classifiers pool. Moreover it remains immune to high-dimensional data due to *random subspace* approach. Connecting it with method of making decisions typical to *minimal distance classifiers*, occurs to be a solid solution for problematic cases of both imbalanced and high-dimensional datasets.

4. Conclusions

Following paper presented a classifier, formulated as an ensemble of subspace projections spanned on combined features of data. In the tested approach, each image corresponds to a feature pair subspaces.

ECE allows a dynamical feature extraction, adjusting itself to every tested object. The fitting procedure randomly selects *attributes* using *random subspace* method and *feature extraction* is done at the classification stage, which makes it a two-level *feature reduction* and *classification* method. All the attributes are fully interpretable, but invertible, taking the best advantages of both *feature selection* and *extraction* methods. It is possible due to discretization of the *exposer* feature space.

It was shown, that this approach led to create classifier that is competitive to existing ones and able to outperform them for some kind of data, proving that it can be used for real-life applications. The method shows robustness to the *curse of dimensionality*, which allows processing the big data problems and a high performance with imbalanced problems.

Acknowledgments

This work was supported by the Polish National Science Center under the grant no. UMO-2015/19/B/ST6/01597 and by the statutory fund of the Faculty of Electronics, Wroclaw University of Science and Technology.

References

Ethem Alpaydin. *Introduction to Machine Learning.* MIT Press, August 2014. ISBN 0262028182.

Richard Bellman. On the reduction of dimensionality for classes of dynamic programming processes. *Journal of Mathematical Analysis and Applications*, 3(2):358–360, October 1961. doi: 10.1016/0022-247X(61)90062-2.

Christopher Bishop. *Pattern Recognition and Machine Learning.* Springer-Verlag New York, 2006. ISBN 978-0-387-31073-2.

Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. The Balanced Accuracy and Its Posterior Distribution. In *2010 20th International Conference on Pattern Recognition (ICPR)*, pages 3121–3124. IEEE, 2010. ISBN 978-1-4244-7542-1. doi: 10.1109/ICPR.2010.764.

Janez Demsar. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7:1–30, January 2006.

Thomas G Dietterich. Ensemble Methods in Machine Learning. In *Multiple Classifier Systems*, pages 1–15. Springer, Berlin, Heidelberg, Berlin, Heidelberg, June 2000. ISBN 978-3-540-67704-8. doi: 10.1007/3-540-45014-9_1.

H He and E A Garcia. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21:1263–1284, September 2009. doi: 10.1109/TKDE.2008.239.

T K Ho. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine ...*, 1998. doi: 10.1109/34.709601.

Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intell. Data Anal.*, 6(5):429–449, 2002.

Daphne Koller and Mehran Sahami. Toward Optimal Feature Selection. In *Machine Learning, Proceedings of the Thirteenth International Conference (ICML '96), Bari, Italy, July 3-6, 1996*, pages 284–292, 1996.

B. Krawczyk, M. Woźniak, and G. Schaefer. Cost-sensitive decision tree ensembles for effective imbalanced classification. *Applied Soft Computing*, 14:554–562, 2014.

Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, Nov 2016. ISSN 2192-6360. doi: 10.1007/s13748-016-0094-0.

Paweł Ksieniewicz. Weles - A Machine Learning library as simple as medieval village. *GitHub repository, https://github.com/w4k2/weles*, 2017.

Paweł Ksieniewicz, Bartosz Krawczyk, and Michal Wozniak. Ensemble of Extreme Learning Machines with Trained Classifier Combination and Statistical Features for Hyperspectral Data. *Neurocomputing*, pages –, 2017. doi: https://doi.org/10.1016/j.neucom.2016.04.076.

L.I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, 2014. ISBN 9781118914557.

Miron B Kursa and Witold R Rudnicki. Feature Selection with the Boruta Package . *Journal of Statistical Software*, 36(11):1–13, September 2010. doi: 10.18637/jss.v036.i11.

Jundong Li and Huan Liu. Challenges of Feature Selection for Big Data Analytics. *arXiv.org*, November 2016.

M Lichman. UCI Machine Learning Repository. [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science., 2013.

Huan Liu and Hiroshi Motoda. *Computational Methods of Feature Selection*, volume 45. Inf. Process. Manage., Boca Raton, FL, 2009. ISBN ISBN 978-1-58488-878-9. doi: 10.1016/j.ipm.2009.03.003.

Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory Undersampling for Class-Imbalance Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550, April 2009. doi: 10.1109/TSMCB.2008.2007853.

Maciej A Mazurowski, Piotr A Habas, Jacek M Zurada, Joseph Y Lo, Jay A Baker, and Georgia D Tourassi. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural Networks*, 21(2-3): 427–436, March 2008. doi: 10.1016/j.neunet.2007.12.031.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

Clifton Phua, Damminda Alahakoon, and Vincent Lee. Minority report in fraud detection: classification of skewed data. *Acm sigkdd explorations newsletter*, 6(1):50–59, June 2004. doi: 10.1145/1007730.1007738.

Witold R Rudnicki, Mariusz Wrzesien, and Wieslaw Paja. All Relevant Feature Selection Methods and Applications. *Feature Selection for Data and Pattern Recognition*, 584 (Chapter 2):11–28, 2015. doi: 10.1007/978-3-662-45620-0_2.

José C Segura, Javier Ramirez, M Carmen Benitez, Ángel de la Torre, and Antonio J Rubio. Improved feature extraction based on spectral noise reduction and nonlinear feature normalization. In *EUROSPEECH-2003*, pages 353–356, September 2003.

Alvy Ray Smith. Color gamut transform pairs. *ACM SIGGRAPH Computer Graphics*, 12 (3):12–19, August 1978. doi: 10.1145/965139.807361.

Katarzyna Stapor. *Metody klasyfikacji obiektów w wizji komputerowej.* Wydawnictwo Naukowe PWN, 2011. ISBN 978-8-3011-6581-9, 9788301165819.

G Svaetichin. Spectral response curves from single cones. *Acta Physiologica Scandinavica*, 39(134):17–46, 1956.

Francisco J Valverde-Albacete and Carmen Peláez-Moreno. 100Information Transfer Factor Explains the Accuracy Paradox. *PLOS ONE*, 9(1):e84217, January 2014. doi: 10.1371/journal.pone.0084217.