

Spis treści

1	Wstęp	2
2	Informacje podstawowe	3
2.1	Imię i nazwisko	3
2.2	Posiadane dyplomy i stopnie naukowe	3
2.3	Zatrudnienie w jednostkach naukowych	3
3	Wskazanie osiągnięcia habilitacyjnego	4
3.1	Wprowadzenie	4
3.2	Zakres badań	8
3.3	Pozostałe osiągnięcia	16
4	Szczegółowy opis wyników	20
4.1	Eksploracja i interpretacja danych za pomocą płytkich modeli	20
4.2	Eksploracja i interpretacja danych za pomocą głębokich modeli	28
4.3	Podsumowanie i dalsze kierunki badań	45
4.4	Literatura	46
5	Informacja o wykazywaniu się istotną aktywnością naukową realizowaną w więcej niż jednej uczelni lub instytucji naukowej	55
5.1	Staże i wizyty naukowe	55
5.2	Udział w grantach realizowanych we współpracy krajowej	55
5.3	Współpraca krajowa i międzynarodowa potwierdzona publikacjami	57
6	Informacja o osiągnięciach dydaktycznych, organizacyjnych oraz popularyzujących naukę	58
6.1	Osiągnięcia dydaktyczne	58
6.2	Osiągnięcia organizacyjne	58
6.3	Osiągnięcia popularyzujące naukę	59

1 Wstęp

Centralnym punktem niniejszego autoreferatu jest informacja o osiągnięciu naukowym, o których mowa w art. 219 ust. 1 pkt. 2 Ustawy z dnia 20 lipca 2018 roku o szkolnictwie wyższym i nauce (Dz. U. 2018, poz. 1668 z późn. zm.). Jako osiągnięcie naukowe przedstawiony został cykl powiązanych tematycznie artykułów naukowych opublikowanych w czasopiśmie naukowych lub w recenzowanych materiałach z konferencji międzynarodowych. Osiągnięcie opiera się na 9 publikacjach, które dotyczą eksploracji i interpretacji danych za pomocą płytkich i głębokich modeli uczenia maszynowego. Podjęta tematyka mieści się zatem w takich obszarach jak klastrowanie (ang. clustering), redukcja wymiarowości (ang. dimensionality reduction) oraz rozpoznawanie wzorców (ang. discovering patterns). Poza wskazaniem i udokumentowaniem głównego osiągnięcia naukowego omówiono także inne osiągnięcia naukowe dotyczące przetwarzania danych niekompletnych, częściowo-nadzorowanego grupowania, wykrywania anomalii, wyjaśnialnego i interpretowalnego uczenia maszynowego oraz rzadkich reprezentacji.

W autoreferacie zawarte są także informacje o istotnej aktywności naukowej realizowanej w więcej niż jednej uczelni lub instytucji naukowej, w szczególności zagranicznej. Aktywność, o której mowa w art. 219 ust. 1 pkt. 3 Ustawy, znajduje potwierdzenie w stażu i wizytach naukowych zrealizowanych w zagranicznych ośrodkach badawczych. Podczas wyjazdów nawiązano (bądź kontynuowano już istniejącą) współpracę, która zaowocowała poszerzeniem horyzontów badawczych oraz powstaniem publikacji naukowych. Wspomnianą aktywność potwierdza również udział w grantach, realizowanych we współpracy z badaczami z więcej niż jednej uczelni lub instytucji naukowej. Poza sformalizowaną współpracą, zostały także nawiązane kontakty nieformalne, których efektem są liczne publikacje naukowe.

Dokument został opracowany na podstawie wytycznych Rady Doskonałości Naukowej i zawiera wszystkie informacje zawarte we wzorze zamieszczonym pod linkiem¹, choć struktura dokumentu została nieznacznie zmieniona. Rozdział 2 zawiera podstawowe informacje dotyczące posiadanych dyplomów oraz dotychczasowego zatrudnienia. W rozdziale 3 przedstawiono podstawowe informacje na temat osiągnięcia naukowego wraz z wykazem prac. Szczegółowy opis osiągnięcia znajduje się w rozdziale 4, zaś rozdział 5 dokumentuje aktywność realizowaną w więcej niż jednej jednostce naukowej. Ostatni rozdział jest z kolei poświęcony osiągnięciom dydaktycznym, organizacyjnym i popularyzującym naukę. Wszystkie osiągnięcia dokumentujące dorobek naukowy są zawarte w załączniku 5.

¹<https://www.rdn.gov.pl/dl/53/attachment/9d4d4a/3%20Autoreferat.pdf>

3 Wskazanie osiągnięcia habilitacyjnego

Niniejszy autoreferat przedstawia osiągnięcie habilitacyjne zatytułowane:

Płytkie i głębokie modele uczenia maszynowego w eksploracji i interpretacji danych.

W dalszej części rozdziału przedstawiono motywację do podjęcia badań, a także wykaz prac wchodzących w skład osiągnięcia wraz z ich zwięzłym omówieniem, a także zamieszczono krótki opis pozostałych obszarów badawczych, które nie wchodzą w skład głównego osiągnięcia. Szczegółowy opis prac badawczych oraz uzyskanych wyników znajduje się w rozdziale 4.

3.1 Wprowadzenie

Eksploracja danych (ang. data mining), znana również jako data mining, stanowi niezwykle istotny proces pozyskiwania wiedzy poprzez odkrywanie ukrytych wzorców, zależności i informacji w rozległych zbiorach danych [71]. W erze dynamicznego rozwoju Internetu, w której dane płyną i gromadzą się w olbrzymich ilościach, pojawia się nieodzowna konieczność rozwijania i tworzenia innowacyjnych technik eksploracji danych, które mogłyby sprostać współczesnym wyzwaniom takim jak:

- duży wolumen danych,
- niska jakość danych,
- wysoka wymiarowość danych,
- potrzeba interpretacji danych.

Uszczegóławiając, przechowywanie i analiza dużych ilości danych generuje koszty i wymaga specjalistycznej infrastruktury. Zbiory danych są często niekompletne lub zaszumione, co wymaga zainicjowania czasochłonnego procesu ich czyszczenia i przetwarzania. Dodatkowo, dane wielowymiarowe stanowią wyzwanie zarówno pod względem analizy, jak i wizualizacji, a tradycyjne techniki eksploracji danych tracą skuteczność w miarę wzrostu liczby wymiarów. Dlatego konieczne jest zastosowanie zaawansowanych metod, takich jak redukcja wymiarowości. Wreszcie, nawet w przypadku, gdy eksploracja danych daje precyzyjne wyniki, interpretacja i wyciąganie wniosków z płynących ze złożonych zbiorów informacji stanowi wyzwanie, szczególnie dla osób, które nie posiadają specjalistycznej wiedzy eksperckiej. Zrozumienie i wyciągnięcie merytorycznych wniosków z ogromnych ilości danych wymaga bowiem nie tylko umiejętności analitycznych, ale również kontekstu i głębokiej wiedzy z danej dziedziny. Dlatego właśnie takie osoby mogą mieć trudności z pełnym wykorzystaniem potencjału wyników eksploracji danych oraz ze zrozumieniem ich znaczenia.

Dzięki stale rosnącej ilości dostępnych danych oraz zaawansowanym technikom ich analizy, eksploracja danych staje się nieocenionym narzędziem wykorzystywanym (w szczególności) w opiece zdrowotnej [51], finansach [32], marketingu [91] i produkcji [42]. W opiece zdrowotnej techniki eksploracji danych są stosowane do analizy dużych zbiorów danych medycznych w celu identyfikacji czynników ryzyka, przewidywania rozwoju choroby i poprawy wyników leczenia, w finansach jest wykorzystywana do wykrywania nieuczciwych transakcji, przewidywania cen akcji i identyfikowania możliwości inwestycyjnych, zaś w marketingu pomaga

w identyfikacji segmentów klientów, personalizacji reklam i optymalizacji strategii cenowych. Z kolei, w produkcji eksploracja danych jest wykorzystywana do optymalizacji procesów produkcyjnych, wykrywania awarii sprzętu i poprawy jakości produktów.

Celem przeprowadzonych badań było wprowadzenie nowych narzędzi uczenia maszynowego dedykowanych eksploracji i interpretacji danych. Techniki takie jak redukcja wymiarowości, grupowanie podprzestrzeni i odkrywanie wzorców mogą pomóc w wydobyciu wiedzy (ang. knowledge discovery) z danych wielowymiarowych, a ponieważ ilość danych stale rośnie, konieczne jest dalsze rozwijanie nowych narzędzi, które mogą skutecznie analizować i interpretować zawarte w nich informacje.

W skład osiągnięcia naukowego wchodzi 9 prac opublikowanych w czasopiśmie z listy JCR (5 prac) lub w materiałach konferencyjnych posiadających kategorię A* (1 praca), A (2 prace) i B (1 praca) według rankingu CORE². Średnia liczba punktów MEiN przypadających na jedną publikację wynosi³ 144.5, a średni Impact Factor (IF) prac to⁴ 6.693. W siedmiu pracach wnioskodawca jest pierwszym autorem, a w jednej drugim.

Poniższa lista prac, stanowiących osiągnięcie naukowe, została uporządkowana tematycznie i będzie później omawiana w takiej kolejności. Dla każdej z prac podano aktualną liczbę punktów MEiN oraz wskaźnik IF, a także liczbę cytowań (bez autocytowań) na podstawie baz Web of Science (WoS), Google Scholar (GS) i Scopus.

[A1] Łukasz Struski, Jacek Tabor, Przemysław Spurek.

Lossy compression approach to subspace clustering.

Information Sciences (435), DOI: 10.1016/j.ins.2017.12.056, p. 161-183, 2018.

Punkty MEiN: 200, IF: 8.233.

Cytowania: 14 (WoS), 15 (GS), 13 (Scopus)

Wkład: *Zdefiniowałem problem badawczy i kierowałem pracami w projekcie. Zapropnowałem nowy algorytm grupowania podprzestrzeni SuMC oparty na teorii informacji oraz kompresji stratnej. Sprawowałem nadzór nad planowaniem, wykonaniem i analizą eksperymentów, a także redagowałem pracę. Mój wkład szacuję na 70%.*

[A2] Łukasz Struski, Przemysław Spurek, Jacek Tabor, Marek Śmieja.

Projected memory clustering.

Pattern Recognition Letters (123), DOI: 10.1016/j.patrec.2019.02.023, p. 9-15, 2019.

Punkty MEiN: 100, IF: 4.757.

Cytowania: 0 (WoS), 0 (GS), 0 (Scopus)

Wkład: *Zaproponowałem algorytm do wykrywania klastrów opisanych przez podprzestrzenie afiniczne równoległe do głównych osi układu współrzędnych dla wysokowymiarowych danych, a także kierowałem pracami w projekcie. Wykonałem większość eksperymentów oraz przeprowadziłem ich analizę, jak również zredagowałem większość pracy. Mój wkład szacuję na 60%.*

²<http://portal.core.edu.au/conf-ranks>

³Liczba punktów MEiN i wskaźnik impact factor aktualne na rok 2022.

⁴Średni wskaźnik IF nie uwzględnia publikacji konferencyjnych, które nie posiadają tego wskaźnika.

- [A3] Łukasz Struski, Marek Śmieja, Jacek Tabor.
Pointed Subspace Approach to Incomplete Data.
 Journal of Classification (37), DOI: 10.1007/s00357-019-9304-3, p. 42-57, 2020.
 Punkty MEiN: 140, IF: 1.333.
 Cytowania: 0 (WoS), 1 (GS), 0 (Scopus)
 Wkład: *Zdefiniowałem problem badawczy i kierowałem pracami w projekcie. Zaproponowałem uogólnienie reprezentacji niekompletnych danych do podprzestrzeni afinicznych. Wykonałem większość eksperymentów oraz przeprowadzałem ich analizę, a także redagowałem pracę. Mój wkład szacuję na 70%.*
- [A4] Łukasz Struski, Michał Sadowski, Tomasz Danel, Jacek Tabor, Igor T. Podolak.
Feature-Based Interpolation and Geodesics in the Latent Spaces of Generative Models.
 IEEE Transactions on Neural Networks and Learning Systems, DOI: 10.1109/TNNLS.2023.3251848, p. 1–15, 2023.
 Punkty MEiN: 200, IF: 14.255.
 Cytowania: 0 (WoS), 0 (GS), 0 (Scopus)
 Wkład: *Brąłem udział w zdefiniowaniu problemu badawczego i kierowałem pracami w projekcie. Zaproponowałem różniczkowalny model interpolacyjny zwracający krzywą sparametryzowaną przez jeden parametr oraz wykonałem jego implementację. Sprawowałem nadzór nad eksperymentami, wykonując większość z nich (wraz z ich analizą), a także uczestniczyłem w redagowaniu pracy. Mój wkład szacuję na 50%.*
- [A5] Marcin Przewięźlikowski, Marek Śmieja, Łukasz Struski.
Estimating conditional density of missing values using deep Gaussian mixture model.
 International Conference on Neural Information Processing (ICONIP), DOI: 10.1007/978-3-030-63836-8_19, p. 220–231, 2020.
 Punkty MEiN: 140, Core rank: B⁵.
 Cytowania: 0 (WoS), 7 (GS), 0 (Scopus)
 Wkład: *Zdefiniowałem problem badawczy i kierowałem pracami w projekcie. Opracowałem i częściowo zaimplementowałem fazę estymacji funkcji gęstości prawdopodobieństwa brakujących wartości. Sprawowałem nadzór nad planowaniem i analizą eksperymentów, a także uczestniczyłem w redagowaniu pracy. Mój wkład szacuję na 40%.*
- [A6] Łukasz Struski, Marcin Mazur, Paweł Batorski, Przemysław Spurek, Jacek Tabor.
Bounding Evidence and Estimating Log-Likelihood in VAE.
 Artificial Intelligence and Statistics Conference (AISTATS), p. 5036–5051, 2023.
 Punkty MEiN: 140, Core rank: A.
 Cytowania: 0 (WoS), 0 (GS), 0 (Scopus)
 Wkład: *Brąłem udział w zdefiniowaniu problemu badawczego i nadzorowałem postępy prac. Opracowałem algorytm do szacowania szerokości luki między dolnym i górnym ograniczeniem log-likelihood, a także go zaimplementowałem. Wykonałem wszystkie*

⁵W roku publikacji, konferencja miała ranking A, od tego czasu jej ranking spadł na poziom B.

eksperymenty i brałem udział w ich analizie, a także uczestniczyłem w redagowaniu pracy. Mój wkład szacuję na 45%.

- [A7] Łukasz Struski, Szymon Knop, Przemysław Spurek, Wiktor Daniec, Jacek Tabor.
LocoGAN – Locally convolutional GAN.
Computer Vision and Image Understanding (221), DOI: 10.1016/j.cviu.2022.103462, p. 103462, 2022.
Punkty MEiN: 100, IF: 4.886.
Cytowania: 0 (WoS), 5 (GS), 0 (Scopus)
Wkład: *Kierowałem pracami w projekcie. Opracowałem i zaimplementowałem metodę lokalnego uczenia modelu generatywnego na częściach wektora szumu. Zaplanowałem i przeprowadziłem większość eksperymentów (wraz z analizą wyników), a także redagowałem pracę. Mój wkład szacuję na 40%.*
- [A8] Łukasz Struski, Tomasz Danel, Marek Śmieja, Jacek Tabor, Bartosz Zieliński.
SONGs: Self-Organizing Neural Graphs.
Winter Conference on Applications of Computer Vision (WACV), DOI: 10.1109/WACV56688.2023.00384, p. 3848–3857, 2023.
Punkty MEiN: 140, Core rank: A.
Cytowania: 0 (WoS), 1 (GS), 0 (Scopus)
Wkład: *Zdefiniowałem problem badawczy i kierowałem pracami w projekcie. Zaproponowałem i zaimplementowałem algorytm uczenia struktury grafów decyzyjnych oparty na procesach Markowa. Wykonałem większość eksperymentów oraz przeprowadziłem ich analizę, a także uczestniczyłem w redagowaniu pracy. Mój wkład szacuję na 60%.*
- [A9] Dawid Rymarczyk, Łukasz Struski, Michał Górszczak, Koryna Lewandowska, Jacek Tabor, Bartosz Zieliński.
Interpretable Image Classification with Differentiable Prototypes Assignment.
European Conference on Computer Vision (ECCV), DOI: 10.1007/978-3-031-19775-8_21, p. 351–368, 2022.
Punkty MEiN: 140, Core rank: A*.
Cytowania: 0 (WoS), 16 (GS), 0 (Scopus)
Wkład: *Przeprowadziłem analizę teoretyczną metody, zaproponowałem wykorzystanie metody Gumbel-Softmax do przypisania części prototypowych do klas oraz regularyzację ortogonalności na slotach prototypowych. Uczestniczyłem również w planowaniu eksperymentów i analizie wyników. Mój wkład szacuję na 30%.*

Przedstawione prace obejmują zarówno rozważania teoretyczne, jak i praktyczne zastosowania. Ich głównym celem jest opracowanie teoretycznych podstaw dla proponowanych rozwiązań przy jednoczesnym zachowaniu lub poprawie ich dokładności i efektywności, w stosunku do najlepszych istniejących rozwiązań.

Podsumowując wymienione powyżej pozycje w kontekście wskazanego osiągnięcia, można stwierdzić iż:

- prace [A1, A2, A3] proponują nowe metody klastrowania oraz reprezentacji niekompletnych danych za pomocą nisko-wymiarowych podprzestrzeni,
- prace [A4, A5, A6, A7] wprowadzają metody oraz modele poszukujące właściwych wzorców w modelach generatywnych,
- prace [A8, A9] opisują sposoby ulepszenia interpretowalnych sieci neuronowych lub struktur decyzyjnych do wyjaśniania danych.

3.2 Zakres badań

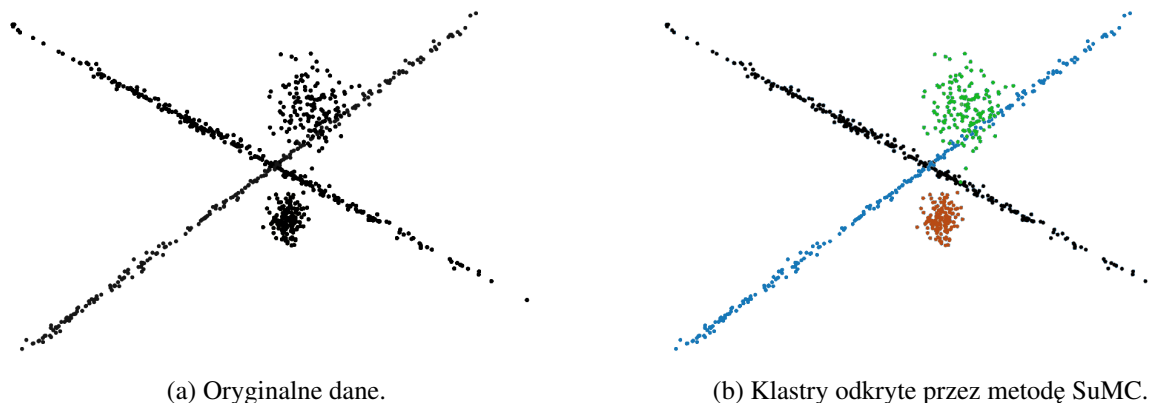
W procesie eksploracji danych wykorzystuje się różnorodne techniki uczenia maszynowego, mające na celu odkrycie ukrytych wzorców, relacji i trendów w danych, które nie są łatwo dostrzegalne dla zwykłego obserwatora. Głównym celem tego procesu jest wyekstrahowanie takiej wiedzy z surowych danych, która może być wykorzystana do podejmowania lepszych decyzji, rozwiązywania skomplikowanych problemów oraz głębszego zrozumienia konkretnego systemu lub zjawiska.

Metody realizujące powyższe zadania można podzielić na dwa rodzaje: płytkie i głębokie. Płytkie metody charakteryzują się prostymi regułami i klarownymi zależnościami między zmiennymi, co umożliwia łatwiejsze zrozumienie i wyjaśnienie wzorców obecnych w danych. Z tego powodu powszechnie uważa się je za transparentne i wiarygodne. Użytkownicy mogą zrozumieć, jak prognozy są generowane na podstawie danych wejściowych, a także co jest istotne we wrażliwych dziedzinach, takich jak np. opieka medyczna. Płytkie modele doskonale sprawdzają się przy mniejszych zbiorach danych, umożliwiając efektywny wybór zmiennych i redukcję wymiarowości. Są niezwykle przydatne w przypadku ograniczeń czasowych i zasobowych, a także w sytuacjach, gdzie istotne jest klarowne zrozumienie podstawowych wzorców.

Jednak wraz ze wzrostem złożoności i rozmiaru zbiorów danych, płytkie modele mogą mieć trudności z uchwyceniem skomplikowanych relacji i wzorców. Z pomocą przychodzi tutaj modele głębokie, które pozwalają nam badać dane w sposób wykraczający poza możliwości płytkich algorytmów. Dzięki zdolności do nauki hierarchicznych reprezentacji i wyłapywania złożonych zależności w danych, głębokie modele potrafią wydobyć bardziej szczegółowe wzorce i poprawić skuteczność predykcji. Są one szczególnie przydatne do obsługi dużych zbiorów danych, które charakteryzują się dużą liczbą obiektów i wymiarów.

Dzięki pojawieniu się zarówno płytkich, jak i głębokich modeli uczenia maszynowego, eksploracja danych dokonała znacznego postępu pod względem możliwości przetwarzania zbiorów danych i odkrywania ukrytych wzorców. Wybór konkretnego rozwiązania zależy od rozważanego problemu, dostępnych zasobów i czasu. Użycie płytkich modeli jest pragmatycznym i skutecznym wyborem w przypadku zadań eksploracji mniejszych zbiorów danych ze względu na ich prostotę i możliwość interpretacji wyników. Natomiast wybór głębokich modeli zazwyczaj prowadzi do osiągnięcia bardzo dobrych wyników kosztem interpretowalności i uzasadnienia podejmowanych decyzji.

W pierwszej części badań skoncentrowano się na **eksploracji i interpretacji danych za pomocą płytkich modeli**. W szczególności skupiono się na opisanych poniżej obszarach badawczych.



Rysunek 1: Sztucznie wygenerowany zbiór danych z czterema klastrami (dwa zero-wymiarowe i dwa jedno-wymiarowe). Po lewej stronie zaprezentowano dane bez podziału na grupy, zaś po prawej, wynik działania metody SuMC określający przynależność punktów do danej podprzestrzeni. Rysunek pochodzi z pracy [A1].

Grupowanie danych [A1, A2] Grupowanie danych (ang. clustering) jest jednym z najważniejszych zagadnień eksploracji danych [10, 49] i uczenia maszynowego [97, 107]. Przez dziesięciolecia problem ten identyfikowano w wielu zastosowaniach i dziedzinach, takich jak wizja komputerowa i rozpoznawanie wzorców [77, 78], fizyka [85], biznes i marketing [80, 98], medycyna [7, 33] itp. Jego ogólnym celem jest zorganizowanie zbioru punktów w zbiór grup, dla których zmaksymalizowane jest „podobieństwo” wewnątrz poszczególnych grup, a zminimalizowane „podobieństwo” między elementami różnych grup.

Chociaż opracowano znaczną liczbę metod grupowania, tylko kilka z nich jest w stanie wydajnie pracować z wysoko-wymiarowymi danymi. W celu rozwiązania tego problemu powstały metody *grupowania podprzestrzeni* (ang. subspace clustering). W przeciwieństwie do tradycyjnych technik grupowania, które traktują cały zbiór danych jako pojedynczą przestrzeń, algorytmy klastrowania podprzestrzeni traktują każdą podprzestrzeń oddzielnie, co pozwala na dokładniejszą identyfikację klastrów na podstawie odpowiednich cech. Takie podejście może pomóc w zidentyfikowaniu ukrytych relacji lub wzorców, które mogą nie być widoczne w danych pełnowymiarowych.

W pracy [A1] przedstawiono nowy algorytm grupowania podprzestrzeni o nazwie SuMC (Subspace Memory Clustering) oparty na teorii informacji [1, 90], zasadzie minimalnej długości opisu (ang. Minimum Description Length Principle, MDLP) [41] i kompresji stratnej [61]. Algorytm ten rozwiązuje jednocześnie dwa podstawowe problemy grupowania podprzestrzeni, a mianowicie sam określa liczbę klastrów oraz ich optymalne wymiary (zob. rysunek 1).

Po przeprowadzeniu szeregu eksperymentów stwierdzono, że zaproponowany model uzyskuje wyniki lepsze lub porównywalne z wiodącymi metodami grupowania podprzestrzeni. Wykazano także, że model ten może być efektywnie wykorzystywany do segmentacji obrazów lub kompresji danych (zob. rysunek 2).

Grupowanie podprzestrzeni to skuteczna klasa algorytmów, które doskonale radzą sobie z odkrywaniem grup opisanych przez różnorodne podprzestrzenie afiniczne. Niestety, arbitralny

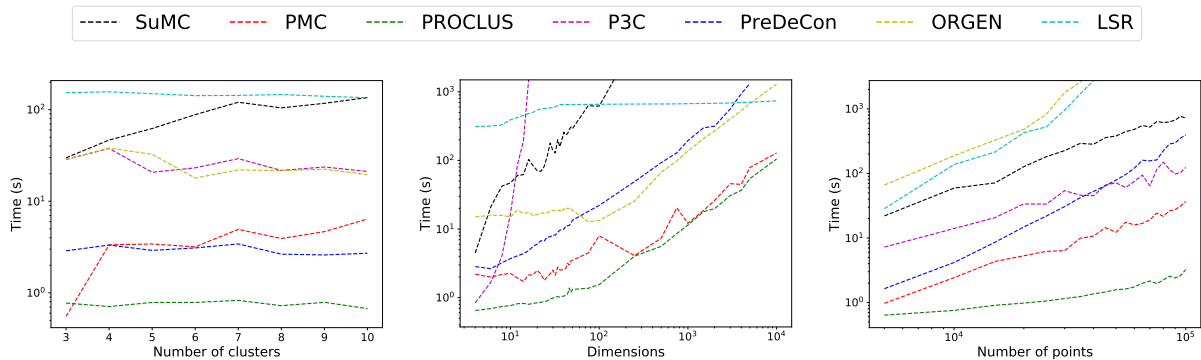


(a) Obrazek oryginalny.



(b) Segmentacja SuMC (wykryto 4 obszary).

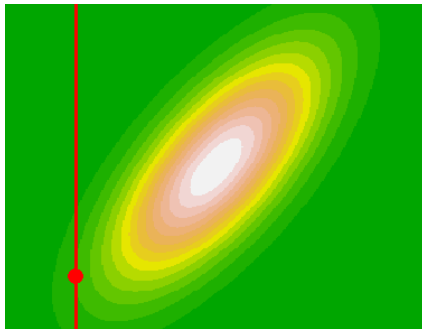
Rysunek 2: Oryginalny obraz wraz z wynikami segmentacji uzyskanych za pomocą metody SuMC. Rysunek pochodzi z pracy [A1].



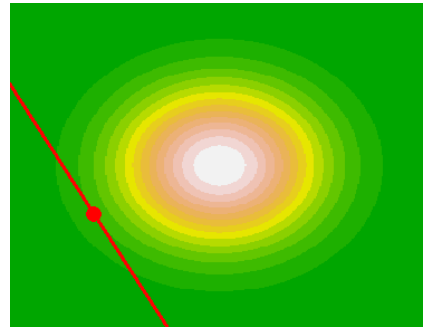
Rysunek 3: Czas przetwarzania danych za pomocą różnych metod, ze względu na liczbę klastrów, wymiar danych oraz liczbę punktów. Rysunek pochodzi z pracy [A2].

wybór tych podprzestrzeni może negatywnie wpływać na złożoność obliczeniową algorytmów, co częściowo ogranicza ich praktyczne zastosowanie w przypadku dużych zbiorów danych. Na szczęście istnieje alternatywna metoda grupowania, znana jako grupowanie rzutowe (ang. projected clustering) [3, 13], która znacząco zmniejsza koszt obliczeniowy. Wykorzystuje ona podprzestrzenie afiniczne równoległe do osi układu współrzędnych, co umożliwia bardziej efektywne przetwarzanie danych. W ramach tego podejścia, klaster jest definiowany jako para dwóch zbiorów – jeden z nich zawiera podzbiór danych należących do danego klastra, a drugi opisuje ich atrybuty definiujące nisko-wymiarową podprzestrzeń. Dzięki temu każdy klaster jest opisany przez najbardziej istotne atrybuty, które są kluczowe dla danego zestawu danych.

Kontynuując badania nad grupowaniem wysoko-wymiarowych danych, w pracy [A2] zaproponowano szybki model klastrowania PMC (Projected Memory Clustering), oparty na podprzestrzeniach afinicznych równoległych do osi układu współrzędnych. Ciekawym odkryciem okazało się to, że przy wykorzystaniu takich podprzestrzeni nie jest konieczne obliczanie macierzy kowariancji danych. Jest to istotne spostrzeżenie, ponieważ obliczenia macierzy kowariancji mają złożoność sześcienną względem wymiaru przestrzeni, co znacząco zwiększa obciążenie obliczeniowe metody SuMC. Dzięki wykorzystanej modyfikacji możemy efektywnie wykry-



(a) Przed transformacją wybielania.



(b) Wynik transformacji wybielania (whitening).

Rysunek 4: Wybielanie danych punktu z \mathbb{R}^2 z brakującym atrybutem na drugiej współrzędnej.

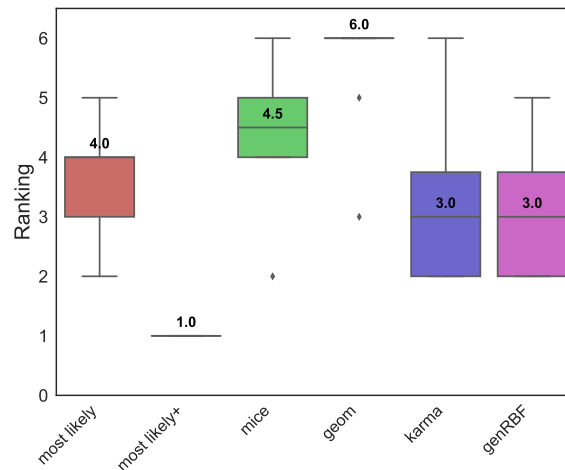
wać klastry w zbiorach danych o liczbie atrybutów większej niż 10^6 . Podobnie jak w metodzie SuMC, liczba klastrów i wymiary poszczególnych podprzestrzeni są dobierane automatycznie.

Przeprowadzone eksperymenty na różnych typach danych wykazały, że PMC wykrywa struktury klastrowe skuteczniej niż powiązane metody projekcji (ang. projection clustering). Co więcej, metoda PMC charakteryzuje się konkurencyjnym czasem przetwarzania danych, jak pokazuje rysunek 3, będąc tylko nieznacznie wolniejszą od metody PROCLUS [3], osiągnącej jednocześnie gorsze wyniki (zaob. tabele 2 i 3 z rozdziale 4). Warto zauważyć, że oba algorytmy mają identyczną złożoność obliczeniową, co podkreśla efektywność PMC w porównaniu do innych metod.

Reprezentacja niekompletnych danych [A3] Dane zgromadzone w dużych zbiorach są często zaszumione lub niekompletne, co z kolei może wpłynąć na precyzję oraz efektywność algorytmów służących do ich eksploracji. Z tego powodu istotne jest rozwijanie skutecznych narzędzi, które umożliwią pracę z takimi danymi.

W standardowych podejściach niepełne dane są reprezentowane jako wektory z brakującymi atrybutami wypełnionymi wartością statystyczną np. średnią [23, 30, 65]. W pracy [A3] uogólniono to podejście i przedstawiono niekompletne dane jako podprzestrzenie afiniczne. Pozwala to na wykonywanie różnych przekształceń afinicznych, takich jak wybielanie (ang. whitening transformation) lub redukcja wymiarowości danych (zob. rysunek 4). Ponadto, w przeciwieństwie do większości technik imputacji [69], taka reprezentacja zachowuje informacje o współrzędnych, których brakuje. Aby ją wykorzystać w praktycznych zadaniach klasyfikacyjnych, takie uogólnione brakujące dane osadzamy w przestrzeni wektorowej i definiujemy na niej iloczyn skalarny. Wyniki przeprowadzonych eksperymentów jednoznacznie potwierdzają, że zastosowanie klasyfikatora SVM na proponowanej reprezentacji niepełnych danych daje najlepsze rezultaty w porównaniu do innych rozważanych metod. Można to zaobserwować na rysunku 5, który obrazuje wyraźną przewagę tej metody pod względem wydajności.

W drugiej części badań podjęto prace nad **eksploracją i interpretacją danych za pomocą głębokich modeli**. Obecnie do przetwarzania dużej ilości danych wykorzystuje się głębokie sieci neuronowe. Jednak aby móc zaufać tym modelom i wyciągnąć wartościowe informacje



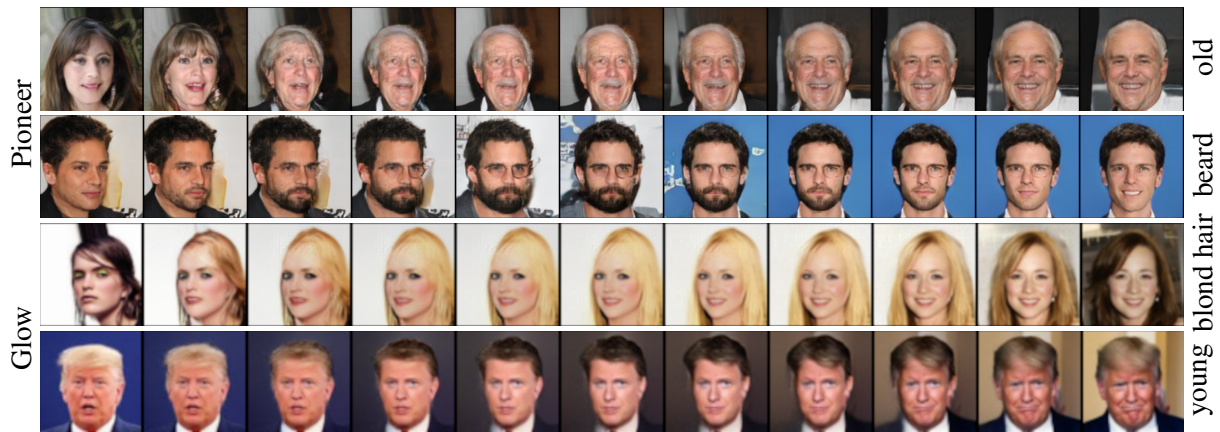
Rysunek 5: Ranking wydajności metod dla wybranych zbiorów z repozytorium UCI [63]. Proces usuwania atrybutów był częściowo determinowany przez odpowiadające im etykiety klas. Warto zauważyć, że nasza metoda „most likely+” osiąga najwyższe rezultaty, co potwierdzają uzyskane wyniki. Rysunek pochodzi z pracy [A3].

z ich predykcji, istotne jest dokładne zrozumienie wyników. Ponadto, kluczowa jest umiejętność wykrywania błędów, nieprawidłowości, czy nawet potencjalnych uprzedzeń (bias) [38] w tych modelach. Ważnym krokiem jest w tym przypadku badanie przestrzeni ukrytych, w których modele tworzą i reprezentują różnorodne wzorce i struktury danych. Poprzez eksplorację tych przestrzeni jesteśmy w stanie identyfikować istotne cechy i zależności, które mają wpływ na podejmowane przez modele decyzje. Dlatego w tej części skupiono się na następujących obszarach badawczych.

Modele generatywne [A4, A5, A6, A7] Modele generatywne pełnią kluczową rolę w uczeniu maszynowym, ponieważ pozwalają na generowanie nowych danych, które nie występowały w zbiorze treningowym. Dzięki temu mogą być one wykorzystywane do rozwiązywania problemów, takich jak generowanie obrazów, dźwięków, tekstu, a nawet tworzenie kompletnych symulacji świata rzeczywistego. Modele generatywne znajdują również zastosowanie w dziedzinach takich jak medycyna, biologia czy inżynieria, gdzie pomagają w modelowaniu i symulowaniu złożonych procesów.

Jednak proces uczenia modeli generatywnych jest skomplikowany i wymaga dużej ilości danych treningowych oraz odpowiedniej architektury i hiperparametrów. Dodatkowo, modele generatywne mogą być podatne na niepożądane zachowania, takie jak *mode collapse* (generowanie tylko jednego lub kilku wzorców danych) czy generowanie nierealistycznych obiektów. Dlatego warto wykorzystywać już nauczone modele do ewaluacji i eksploracji przestrzeni ukrytej, na przykład poprzez interpolację. Dzięki temu można lepiej zrozumieć, jakie cechy lub kombinacje cech są istotne dla generowania wysokiej jakości danych oraz odkryć interesujące wzorce i struktury.

W pracy [A4] zdefiniowano ogólne podejście do procesu interpolacji, które wykorzystuje sztuczne sieci neuronowe. To podejście umożliwia jednocześnie poszukiwanie krzywych geo-



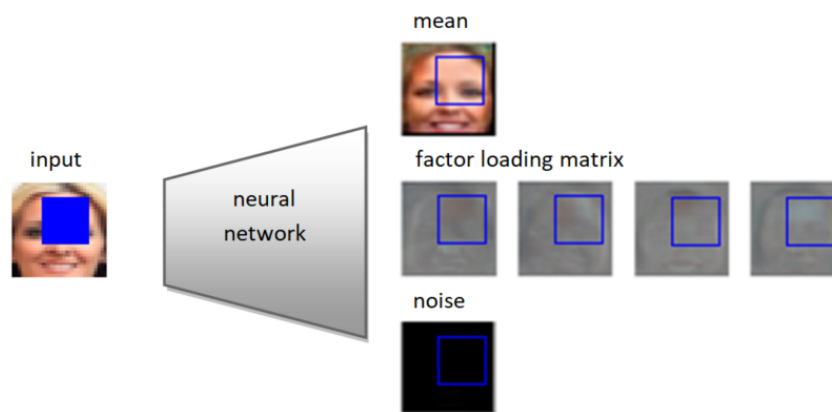
Rysunek 6: Przykłady interpolacji wykorzystujące dwa różne modele generatywne. W pierwszych dwóch rzędach zastosowano wstępnie przeszkolony model Pioneer, podczas gdy w ostatnich rzędach użyto pre-trenowanego modelu Glow. Na prawej stronie obrazka zdefiniowano cechę, która jest pożądana w ścieżce interpolacyjnej. Rysunek pochodzi z pracy [A4].

dezyjnych [A4, sekcja IV] oraz ścieżek interpolacyjnych o określonych cechach, co zostało zobrazowane na rysunku 6. Przeprowadzone eksperymenty na różnych modelach generatywnych (DCGAN [83], Glow [53], Pioneer [46], StackGAN [113]) oraz na różnych zbiorach danych (MNIST [60], SVHN [75], CelebA [66], CUB-200 [104], dane o związkach chemicznych) jednoznacznie wskazują, że zaproponowany model przewyższa liniową i sferyczną interpolację w odnajdywaniu ścieżek interpolacyjnych zawierających daną cechę nawet o ponad 90% (patrz [A4, tabele I i II]).

Z kolei w pracy [A5] badano przestrzeń niekompletnych obrazów w celu ich uzupełnienia (ang. image imputation). Zaprezentowano metodę DMFA (ang. Deep Mixture of Factor Analyzers), która wykorzystuje głęboką sieć neuronową oraz techniki mieszaniny rozkładów normalnych (ang. Gaussian Mixture Model, GMM) do estymacji funkcji gęstości prawdopodobieństwa brakujących wartości (zob. rysunek 7).

Efektywność proponowanego podejścia została zweryfikowana na kilku zbiorach danych, takich jak MNIST [59], Fashion-MNIST [109] i CelebA [66]. Badania wykazały, że metoda DMFA osiąga lepsze wyniki w porównaniu do MFA [87], uwzględniając wartości logarytmicznej funkcji wiarygodności (ang. log-likelihood) dla brakujących obszarów obrazków. Dodatkowo, imputacje uzyskane przez zastąpienie brakujących wartości średnim wektorem zwracanym przez model DMFA wykazują znacznie lepsze rezultaty w porównaniu do MFA. Szczegółowe wyniki przedstawione są w rozdziale 4 w tabeli 5 oraz na rysunku 18.

W kolejnej pracy [A6] skoncentrowano się na porównaniu wytrenowanych modeli generatywnych. W przypadku modeli opartych na funkcjach kosztu, takich jak VAE [54, 86], które wykorzystują dolne ograniczenie wiarygodności danych (ang. Evidence Lower Bound, ELBO), trudno jest dokładnie porównać wpływ treningu między różnymi modelami, ponieważ znamy jedynie dolne ograniczenie, a nie dokładną wiarygodność danych. Z tego powodu, w pracy [A6] zaproponowano skuteczne górne ograniczenie, które umożliwia efektywne przybliżenie wartości wiarygodności danych, co pozwala na lepsze porównanie jakości danych generowanych przez

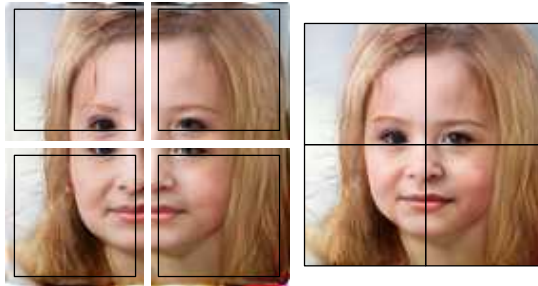


Rysunek 7: Schemat modelu DMFA. Model DMFA na wejściu przyjmuje niepełny obraz (z obszarem wewnątrz niebieskiego kwadrata), a zwraca parametry warunkowej gęstości rozkładu normalnego: średnią, macierz czynników (użyto 4) i macierz szumów. Rysunek pochodzi z [A5].

różne modele. Ponadto przeprowadzono obszerne badania teoretyczne i eksperymentalne, które obejmowały porównanie z innymi górnymi ograniczeniami dla log-wiarygodności danych oraz wykorzystanie tego oszacowania jako narzędzia do oceny modeli wytrenowanych na różnych dolnych ograniczeniach.

Uzyskane wyniki potwierdzają większą skuteczność proponowanego oszacowania w porównaniu do innych rozważanych metod. Jedynie dla modeli VAE trenowanych na zbiorach danych SVHN [75] i CelebA [66], oszacowanie TVO_{50} -VG-B [68] uzyskało nieznacznie lepsze wyniki (patrz [A6, tabela 1]). Ponadto otrzymane wyniki wskazują na to, że proponowane podejście jest jedynym, które gwarantuje zmniejszenie obliczonej luki wariacyjnej (ang. variational gap) wraz z wzrostem liczby próbek (warto również zauważyć, że w przypadku innych metod, ich granice mogą rosnać wraz z rozmiarem próbki).

W pracy [A7] badano struktury szumu używanego do generowania obrazów przy użyciu modeli generatywnych. Celem badań było opracowanie prostego, łatwego do zrozumienia i modyfikacji modelu, który umożliwiłby generowanie obrazów. W dziedzinie kryminalistyki, szczególnie przy opracowywaniu portretów pamięciowych, opisuje się szczegółowe cechy zewnętrzne, takie jak twarz osoby. Proces ten opiera się na ogólnym szablonie, do którego wprowadza się szczegółowe informacje, takie jak położenie oczu czy kształt nosa. W wyniku badań opracowano model LocoGAN, który różni się od tradycyjnych modeli GAN przez zdolność generowania tylko lokalnych fragmentów obrazów, co przedstawiono na rysunku 8. Dzięki lokalnemu uczeniu oraz wykorzystaniu kanałów pozycyjnych, model ten osiąga niezwykłą zdolność do przeszczepiania różnych fragmentów obrazów na inne obrazy, a także generowania obrazów o pełnej okresowości o niemal nieskończonej długości. Architektura modelu LocoGAN jest prosta, ale posiada kilka ważnych zalet w porównaniu do standardowych modeli GAN. Przede wszystkim umożliwia on tworzenie obrazów o różnych rozmiarach i uczenie na jednym rozmiarze szumu, co pozwala uniknąć zwiększania liczby parametrów sieci, związanego z użyciem obrazów o większej rozdzielczości.



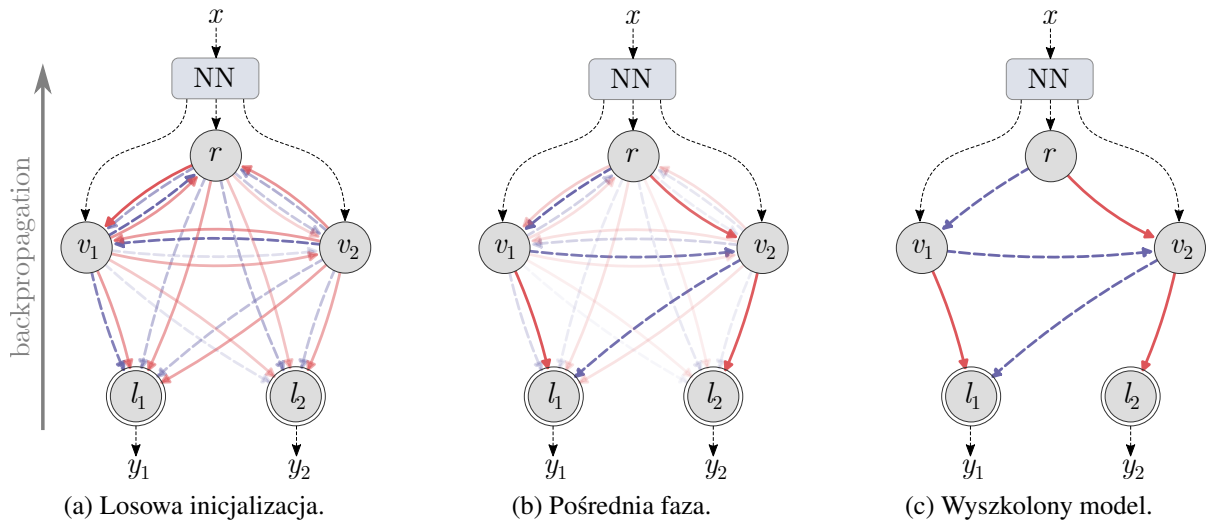
Rysunek 8: Wyniki lokalnego uczenia się modelu LocoGAN. Po lewej stronie zaprezentowano wynik osobno wygenerowanych czterech części obrazu, zaś po prawej fragmenty (kwadranty) połączone w jeden pełny obraz. Rysunek pochodzi z pracy [A7].

Interpretowalność i wyjaśnialność decyzji podejmowanych przez głębokie modele [A8, A9]

W ostatnich latach intensywnie rozwijane są badania dotyczące łączenia głębokich sieci neuronowych z innymi strukturami, takimi jak drzewa decyzyjne i grafy. Podejścia takie mają trzy główne zalety: są łatwe do zrozumienia (ponieważ opierają się na sekwencyjnych decyzjach), mogą działać szybciej i oferują hierarchię klas. Niestety, drzewa decyzyjne mają podstawową wadę w porównaniu z grafami decyzyjnymi – nie są w stanie ponownie używać węzłów decyzyjnych, co prowadzi do potrzeby stosowania dużej liczby węzłów i połączeń między nimi. Mimo tego, grafy decyzyjne rzadko są wykorzystywane w głębokim uczeniu, ponieważ brakuje skutecznych technik ich trenowania opartych na gradiencie. W pracy [A8] przedstawiono ogólny paradygmat bazujący na procesach Markowa, który pozwala na efektywne uczenie specjalnego typu grafów decyzyjnych, zwanych samoorganizującymi się grafami neuronowymi (ang. Self-Organizing Neural Graph, SONG, zob. rysunek 9). Eksperymenty przeprowadzone na licznych zbiorach danych jednoznacznie potwierdzają, że proponowana metoda działa równie dobrze lub lepiej niż inne istniejące modele decyzyjne. Co więcej SONG osiąga porównywalne wyniki przy mniejszej liczbie węzłów. Jest to zgodne z wcześniejszymi badaniami przedstawionymi w pracy [94], które dowodzą, że grafy decyzyjne znacznie redukują zapotrzebowanie na pamięć, a jednocześnie poprawiają zdolność generalizacji sieci.

Z kolei w pracy [A9] zaprezentowano nowy, interpretowalny model oparty na prototypach, które można postrzegać jako wyuczone reprezentacje podsumowujące cechy klasy lub grupy podobnych instancji. Modele oparte na prototypach [21, 101], wykorzystujące głębokie sieci neuronowe, często stają przed wyzwaniami optymalizacyjnymi. Kryje się za tym kilka powodów. Przede wszystkim, takie modele zakładają osobne prototypy dla każdej klasy, co wymaga wieloetapowej optymalizacji. Ponadto modele te podejmują decyzje na podstawie braku prototypu (jest to tzw. proces wnioskowania negatywnego), co nie zawsze jest skuteczne, a także mogą generować niejasne prototypy.

W celu rozwiązania powyższych problemów, został opracowany nowy, interpretowalny model oparty na prototypach [A9]. Model ten umożliwi współdzielenie części prototypowych pomiędzy klasami, co znacznie zmniejsza ich liczbę i upraszcza proces uczenia. Dodatkowo, automatyczne przypisywanie prototypów do klas jest w pełni różniczkowalne, co pozwala na uzyskanie bardziej optymalnego modelu. Co więcej, model skupia się na bardziej istotnych



Rysunek 9: Etapy szkolenia modelu SONG, który wykorzystuje spadek gradientu do modyfikowania struktury grafu i prawdopodobieństw poszczególnych połączeń. Bazując na danych wejściowych x , sieć neuronowa (NN) wyodrębnia reprezentację wektorową, która jest przekazywana do części grafowej w celu uzyskania predykcji dla każdej klasy (y_1 i y_2). Na początku uczenia graf ma korzeń r , węzły v_1 i v_2 , liście l_1 i l_2 oraz losowo zainicjowane krawędzie (a). W kolejnych iteracjach treningowych znaczenie wybranych krawędzi rośnie (b), ostatecznie dając rzadki graf binarny, z dwoma silnymi krawędziami wychodzącymi z każdego węzła (c). Rysunek pochodzi z pracy [A8].

cechach wizualnych poprzez wprowadzenie nowej funkcji podobieństwa, która wprowadza kontrast pomiędzy częścią prototypów a tłem.

W oparciu o liczne eksperymenty pokazano, że proponowane podejście uzyskuje wysoką skuteczność na zbiorach danych CUB-200-2011 i Stanford Cars, przy znacznie ograniczonej liczbie prototypów. Dodatkowo, przeprowadzono badanie użytkowników, które potwierdza, że uzyskane części prototypowe są bardziej specyficzne (ang. salient) niż te uzyskane za pomocą metod konkurencyjnych.

3.3 Pozostałe osiągnięcia

Pozostałe (zaprezentowane poniżej) obszary zainteresowań naukowych wnioskodawcy koncentrują się wokół sztucznej inteligencji i wykorzystują szeroką gamę technik badawczych, od płytkich metod uczenia maszynowego po głębokie sieci neuronowe:

(1) Przetwarzanie danych niekompletnych:

[B1] Marcin Przewięźlikowski, Marek Śmieja, Łukasz Struski, Jacek Tabor.

MisConv: Convolutional Neural Networks for Missing Data.

In IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), p. 2060–2069, 2022.

- [B2] Marcin Sendera, Przemysław Spurek, Łukasz Struski.
Missing Glow Phenomenon: learning disentangled representation of missing data.
International Conference on Neural Information Processing (ICONIP), p. 196–204, 2021.
- [B3] Marek Śmieja, Maciej Kołomycki, Łukasz Struski, Mateusz Juda, Mario A.T. Figueiredo.
Can auto-encoders help with filling missing data?
In ICLR Workshop on Integration of Deep Neural Models and Differential Equations (DeepDiffEq), p. 1–6, 2020.
- [B4] Tomasz Danel, Marek Śmieja, Łukasz Struski, Przemysław Spurek, Łukasz Maziar-ka.
Processing of incomplete images by (graph) convolutional neural networks.
In International Conference on Neural Information Processing (ICONIP), p. 512—523, 2020.
- [B5] Marek Śmieja, Maciej Kołomycki, Łukasz Struski, Mateusz Juda, Mario A.T. Figueiredo.
Iterative imputation of missing data using auto-encoder dynamics.
In International Conference on Neural Information Processing (ICONIP), p. 258—269, 2020.
- [B6] Marek Śmieja, Łukasz Struski, Jacek Tabor, Mateusz Marzec.
Generalized RBF kernel for incomplete data.
Knowledge-Based Systems 173, p. 150–162, 2019.
- [B7] Marek Śmieja, Łukasz Struski, Jacek Tabor, Bartosz Zieliński, Przemysław Spurek.
Processing of missing data by neural networks.
Conference on Neural Information Processing Systems (NeurIPS), Advances in Neural Information Processing Systems 31, p. 2724–2734, 2018.
- [B8] Łukasz Struski, Marek Śmieja, Bartosz Zieliński, Jacek Tabor.
Regression SVM for incomplete data.
Schedae Informaticae 26, p. 23–35, 2017.

Jednym z głównych obszarów zainteresowań badawczych habilitanta jest analiza niekompletnych danych. Wprowadził on szereg metod, które mogą przetwarzać niekompletne dane np. poprzez modelowanie rozkładów brakujących wartości za pomocą operacji macierzowych [B1], wykorzystując głębokie modele poprzez aproksymację gradientu gęstości logarytmicznej z błędem rekonstrukcji albo wykorzystując przetwarzaną reprezentację grafowych sieci konwolucyjnych [B2, B3, B4, B5]. W pracy [B6] skonstruowano jądro, które uogólnia standardowe jądro Gaussa RBF na przypadek niekompletnych danych. Zamiast korzystać z typowych technik imputacji, które uzupełniają brakujące atrybuty pojedynczymi wartościami, modelujemy możliwe wyniki dla brakujących współrzędnych za pomocą rozkładu danych. Prace [B7, B8] adaptują jądra SVM i sieci neuronowe, modelując brakujące

dane za pomocą rozkładu danych. Następnie funkcja jądrowa lub funkcja aktywacji jest uśredniania w sposób analityczny, uwzględniający wszystkie możliwe wypełnienia.

(2) Częściowo-nadzorowane grupowanie:

- [C9] Marek Śmieja, Łukasz Struski, Mario A.T. Figueiredo.
A classification-based approach to semi-supervised clustering with pairwise constraints.
Neural Networks 127, p. 193–203, 2020.
- [C10] Marek Śmieja, Łukasz Struski, Jacek Tabor.
Semi-supervised model-based clustering with controlled clusters leakage.
Expert Systems with Applications 85, p. 146–157. 2017.

W artykule [C9] zaproponowano nową sieć neuronową o nazwie *S3C2* do częściowo-nadzorowanego grupowania. Podejście to rozkłada zadanie na dwa prostsze zadania klasyfikacyjne, w których pierwszy etap wykorzystuje syjamskie sieci neuronowe do oznaczenia nieoznakowanych par punktów jako konieczne lub niemożliwe połączenie, a drugi etap wykorzystuje nadzorowaną metodę grupowania opartą na sieci neuronowej. Z kolei w pracy [C10] zdefiniowano częściowo-nadzorowaną wersję modelu mieszaniny rozkładów normalnych o nazwie *C3L* do znajdowania klastrów w częściowo skategoryzowanych zbiorach danych. Podejście to integruje informacje z początkowej klasyfikacji z prawdziwą strukturą danych poprzez pobieranie klastrów podobnych do Gaussa zawartych w jednej z początkowych klas w ramach wcześniej określonego poziomu ufności. Proponowane podejścia prezentują wysoką skuteczność, która została potwierdzona w wielu eksperymentach i na wielu zbiorach danych.

(3) Wykrywanie anomalii:

- [D11] Łukasz Maziarka, Marcin Sendera, Marek Śmieja, Łukasz Struski, Jacek Tabor, Przemysław Spurek.
OneFlow: One-class flow for anomaly detection based on a minimal volume region.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(11), p. 8508–8519, 2022.
- [D12] Marcin Sendera, Marek Śmieja, Łukasz Maziarka, Łukasz Struski, Przemysław Spurek, Jacek Tabor.
Flow-based SVDD for anomaly detection.
In International Conference on Neural Information Processing (ICONIP), p. 419–431, 2019.

Następny obszar badań dotyczy wykrywania anomalii. W oparciu o model flow i estymator kwantyli Bernsteina zbudowano nowy model *OneFlow* do wykrywania anomalii, który znajduje minimalny obszar ograniczający „pożądane” dane [D11]. Druga praca [D12] opisuje głęboki model klasyfikacyjny *FlowSVDD*, który realizuje zasadę SVDD (ang. Support Vector Data Description). W przeciwieństwie do innych podejść związanych z zasadą

SVDD, FlowSVDD jest tworzony przy użyciu modeli opartych na modelach flow, które w naturalny sposób zapobiegają degradacji hipersfery do pojedynczego punktu.

(4) Wyjaśnialne i interpretowalne uczenie maszynowe:

[E13] Mikołaj Sacha, Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, Bartosz Zieliński.
ProtoSeg: Interpretable Semantic Segmentation with Prototypical Parts.
In Winter Conference on Applications of Computer Vision (WACV), 2023.

[E14] Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, Bartosz Zieliński.
Protopshare: Prototypical parts sharing for similarity discovery in interpretable image classification.
In ACM International Conference on Knowledge Discovery and Data Mining (KDD), 2021.

W kolejnych pracach [E13, E14] rozszerzono funkcjonalność modeli wykorzystujących mechanizm prototypowych części [21, 62, 73]. Praca [E13] przedstawia nowy model semantycznej segmentacji obrazu, który odkrywa koncepcje semantyczne i osiąga wysoką skuteczność. Natomiast praca [E14] wprowadza model, który wydajniej zarządza prototypowymi częściami między klasami, co prowadzi do znajdowania bardziej spójnych prototypów oraz zwiększa odporność modelu na zakłócenia.

(5) Rzadkie reprezentacje:

[F15] Klaudia Bałazy, Łukasz Struski, Marek Śmieja, Jacek Tabor.
r-softmax: Generalized Softmax with Controllable Sparsity Rate.
In International Conference on Computational Sciences (ICCS), p. 1–8, 2023.

[F16] Tomasz Danel, Przemysław Spurek, Jacek Tabor, Marek Śmieja, Łukasz Struski, Agnieszka Słowik, Łukasz Maziarka.
Spatial graph convolutional networks.
In International Conference on Neural Information Processing (ICONIP), p. 668—675, 2020.

Ostatni kierunek badań jest związany z rzadkimi reprezentacjami. Praca [F15] wprowadza modyfikację funkcji *softmax*, która generuje rzadkie rozkłady prawdopodobieństwa z kontrolowanym współczynnikiem rzadkości. Z kolei praca [F16] przedstawia sieć konwołucyjną grafów przestrzennych (SGCN), która wykorzystuje cechy przestrzenne do efektywnego uczenia z grafów.

4 Szczegółowy opis wyników

W niniejszym rozdziale zostaną szczegółowo omówione wyniki uzyskane w ramach wskazanego osiągnięcia naukowego. W opisie wyróżnimy dwa wątki, z których jeden dotyczyć będzie modeli płytkich, zaś drugi głębokich.

Niektóre rysunki zostały zapożyczone z odpowiednich publikacji i w konsekwencji zawierają opis w języku angielskim. Rysunki wykonane tylko na potrzeby niniejszego dokumentu są opisane w języku polskim. W tabelkach, zaś część dziesiętna liczby rzeczywiste jest oddzielana kropką (zgodnie z konwencją anglojęzyczną), aby zachować spójność z publikacjami, z których pochodzą.

4.1 Eksploracja i interpretacja danych za pomocą płytkich modeli

4.1.1 Grupowanie danych [A1, A2]

W dziedzinie klastrowania danych istnieją dwa podstawowe podejścia: klasyczne grupowanie danych oraz klastrowanie podprzestrzeni (ang. subspace clustering). Oba te podejścia mają na celu identyfikację naturalnych grup danych, jednak różnią się w metodologii i założeniach. Pierwsze z nich polega na grupowaniu obiektów na podstawie ich podobieństw i różnic w całej przestrzeni cech, zaś drugie skupia się na identyfikacji grup w konkretnych podprzestrzeniach przestrzeni cech. Klastrowanie podprzestrzeni, uwzględniające ich specyfikę, może zapewnić bardziej precyzyjną i dogłębną analizę danych w porównaniu do tradycyjnych algorytmów klastrowania, zwłaszcza dla danych o wysokim wymiarze. Ze względu na wysoką skuteczność tych technik oraz rosnące znaczenie danych o dużym wymiarze, wiele badań skupiło się na opracowaniu nowych metod klastrowania podprzestrzeni, takich jak SSC [31] i jego rozszerzenie [110], LRR [22], LRSSC [102], TSC [45] oraz GSC [79].

Główną wadą powyższych metod jest fakt, że nie zwracają one optymalnych podprzestrzeni. Powód tej wady jest zależny od metody, a zatem może być różny. Na przykład SSC zakłada, że każdy punkt w pojedynczej podprzestrzeni jest rzadką kombinacją kilku innych punktów ze zbioru danych [82]. Założenie to może się nie sprawdzić w praktyce, zwłaszcza gdy podprzestrzenie w dużym stopniu się pokrywają. Z drugiej strony, LRR gwarantuje dokładne klastrowanie tylko wtedy, gdy podprzestrzenie nie mają przecięć, podczas gdy LRSSC gwarantuje tylko poprawne sąsiedztwa, co nie zawsze prowadzi do dokładnego klastrowania. W związku z tym, opisane poniższej pracy koncentrują się na pokonaniu tych ograniczeń i dążą do osiągnięcia lepszych rezultatów.

Praca [A1] Jednym z wyzwań klastrowania podprzestrzeni jest określenie, które podprzestrzenie są istotne dla grupowania danych. Aby rozwiązać ten problem w pracy [A1] wykorzystano teorię entropii Shannona [1, 90], złożoność Kolmogorowa [61] oraz zasadę minimalnej długości opisu (ang. minimum description length principle, MDLP) [41]. Podobnie jak przy zasadzie brzytwy Ockhama, starano się opisywać punkty (czyli obiekty z danych) jak najbardziej zwięźle (co jest również podstawową zasadą stojącą za MDLP). Głównym celem było zminimalizowanie

błędu kwadratowego przy jednoczesnym utrzymaniu całkowitej ilości pamięci na stałym poziomie, podobnie jak w koncepcji pochodzącej z transmisji danych kanałem o stałej przepustowości.

Przejdźmy zatem do opisu problemu w sposób bardziej formalny. Niech dany będzie zbiór danych $X \subset \mathbb{R}^N$, gdzie $N \in \mathbb{N}$, który chcemy podzielić na $k \in \mathbb{N}$ klastrów X_1, \dots, X_k ($X = X_1 \cup \dots \cup X_k$) tak, aby każdy klaster X_i był reprezentowany przez podprzestrzeń V_i w taki sposób, aby kwadrat błędu między punktami zbioru X_i , a podprzestrzenią V_i był jak najmniejszy. W konsekwencji, błąd kwadratowy pojedynczego klastra X_i jest dany wzorem

$$\text{error}(X_i, V_i) = \sum_{x \in X_i} \text{dist}(x, V_i)^2,$$

gdzie $\text{dist}(x, V_i)$ opisuje odległość między punktem, a jego ortogonalną projekcją na podprzestrzeń V_i . Ten błąd można wyrazić za pomocą wartości własnych macierzy kowariancji, o czym mówi następujące twierdzenie, które pochodzi z pracy [70].

Twierdzenie 1. Niech $N \in \mathbb{N}$, $X \subset \mathbb{R}^N$ będzie danym zbiorem danych, a $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$ będą uporządkowanymi wartościami własnymi odpowiadającymi wektorom własnym v_1, \dots, v_N macierzy kowariancji. Wtedy minimalny błąd aproksymacji kwadratowej można wyrazić jako

$$\text{error}(X, m) = \text{error}(X, V) = |X| \cdot \sum_{i=m+1}^N \lambda_i \quad \text{dla } m \leq N \quad (1)$$

gdzie $|\cdot|$ oznacza moc zbioru, podprzestrzeń V jest zdefiniowana wzorem

$$V = \text{mean}_X + \text{span}(v_1, \dots, v_m),$$

zaś mean_X to średnia ze współrzędnych wektorów ze zbioru X .

Powyższe twierdzenie pozwala znaleźć optymalną podprzestrzeń o zadanym wymiarze. Ze względów praktycznych, związanych z optymalizacją całkowitego błędu klastrowania, zmodyfikowano tą funkcję do przypadku ciągłego (dla rzeczywistych wartości liczby m) w sposób afiniczny:

$$\text{error}(X, m) := (\lfloor m \rfloor + 1 - m) \cdot \text{error}(X, \lfloor m \rfloor) + (m - \lfloor m \rfloor) \cdot \text{error}(X, \lfloor m \rfloor + 1).$$

Zatem głównym problemem optymalizacyjnym jest znalezienie podziału zbioru X na k klastrów X_1, \dots, X_k oraz ustalenie wymiarów n_1, \dots, n_k dla odpowiadających im podprzestrzeni V_i tak, aby minimalizowały one całkowity błąd kwadratowy:

$$\text{error}(X_1, n_1; \dots; X_k, n_k) = \text{error}(X_1, n_1) + \dots + \text{error}(X_k, n_k), \quad (2)$$

spełniając jednocześnie warunek głoścący, że suma pamięci użytej do opisu poszczególnych klastrów nie przekracza pamięci przydzielonej do opisu całego zbioru danych. Warunek ten można zapisać w sposób uproszczony jako następującą nierówność:

$$r \cdot (n_1 \cdot |X_1| + \dots + n_k \cdot |X_k|) \leq n \cdot |X|,$$

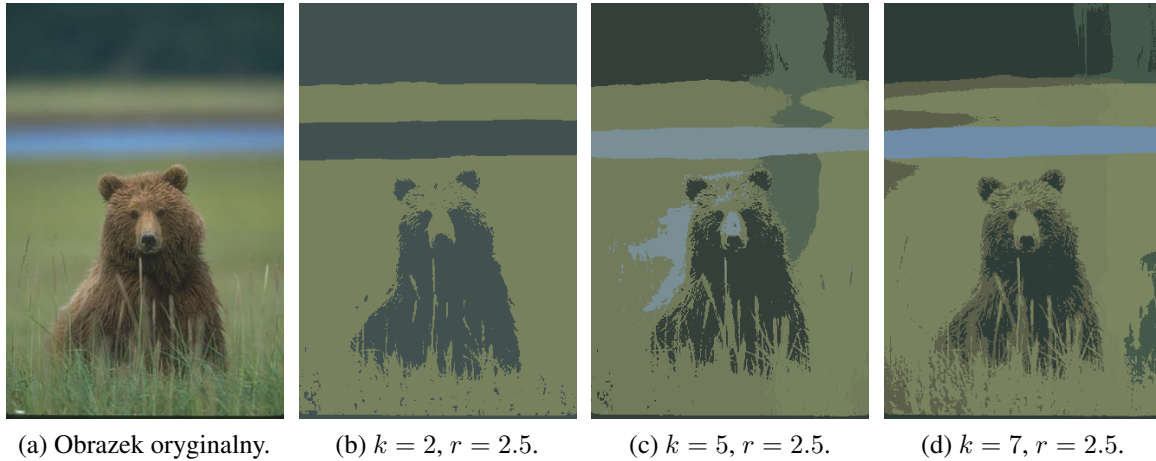
gdzie $r > 1$ oznacza współczynnik kompresji danych.

Wymiar danych	Miara podobieństwa	Liczba zbiorów dla których SuMC było			Średnia	Metoda
		lepsze	takie same	gorsze		
10	Rand	373	20	7	0.858	ORCLUS
					0.981	SuMC
	Jaccard	372	20	8	0.673	ORCLUS
					0.950	SuMC
12	Rand	373	121	6	0.873	ORCLUS
					0.985	SuMC
	Jaccard	373	120	7	0.718	ORCLUS
					0.967	SuMC
14	Rand	522	77	1	0.856	ORCLUS
					0.990	SuMC
	Jaccard	522	77	1	0.666	ORCLUS
					0.976	SuMC
16	Rand	560	139	1	0.855	ORCLUS
					0.990	SuMC
	Jaccard	561	138	1	0.676	ORCLUS
					0.975	SuMC
18	Rand	734	66	0	0.872	ORCLUS
					0.993	SuMC
	Jaccard	734	66	0	0.704	ORCLUS
					0.981	SuMC
20	Rand	758	141	1	0.868	ORCLUS
					0.988	SuMC
	Jaccard	757	141	2	0.704	ORCLUS
					0.972	SuMC

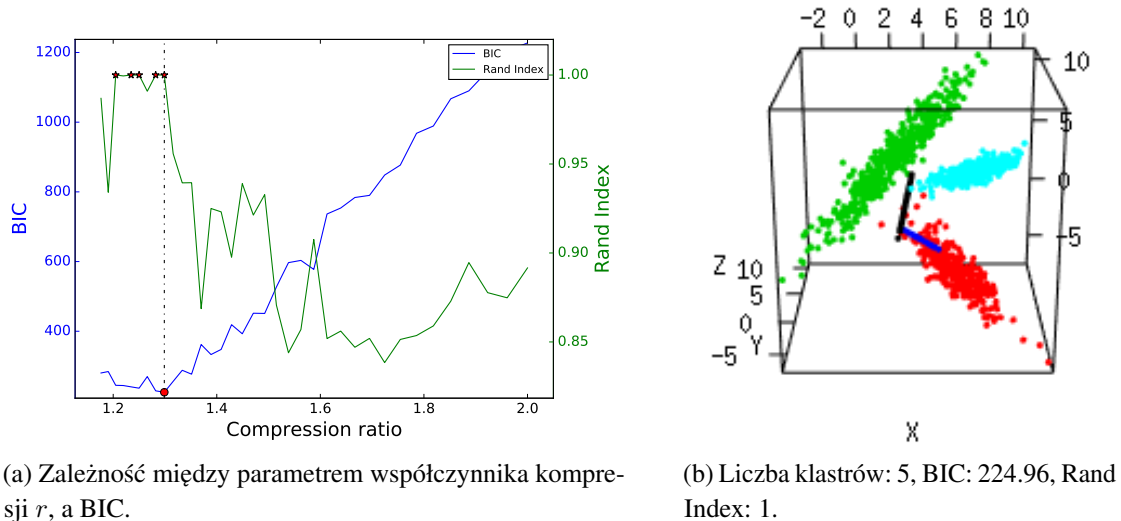
Tabela 1: Część wyników porównania metod SuMC i ORCLUS na syntetycznych zbiorach danych. W przeprowadzonym eksperymencie wygenerowano losowo 7400 zestawów danych, dla których metoda SuMC wygrała 6182 razy i przegrała 370 razy względem miary Rand (6184 do 373 względem miary Jaccarda). Pełny wykaz wyników można znaleźć w pracy [A1].

Ten nietrywialny problem optymalizacyjny został w pracy [A1] skutecznie rozwiązany przy użyciu zmodyfikowanej wersji algorytmu opartego na koncepcji Hartigana [43]. W ramach zaproponowanego rozwiązania zastosowano losową inicjalizację klastrów oraz procedurę redukcji niepotrzebnych klastrów. Przeprowadzone testy na danych sztucznych i rzeczywistych jednoznacznie wskazują, że metoda SuMC osiąga lepsze wyniki w porównaniu do metod takich jak ORCLUS [4] czy 4C [12]. Na przykład, na zestawach danych wygenerowanych losowo SuMC uzyskuje wyższy indeks Rand w ponad 90% przypadków, co można zaobserwować w tabeli 1.

Ponadto metoda SuMC z powodzeniem może być używana do rozwiązywania innych problemów, takich jak segmentacja lub kompresja obrazów. Rysunek 10 prezentuje kilka rezultatów tej metody w kontekście znalezienia atrakcyjnych struktur wizualnych w rzeczywistych zbiorach danych. Zauważamy, że metoda SuMC daje obiecujące wyniki, umożliwiając wykrycie obiektów pierwszoplanowych (zob. rysunek 10b), a także dostarczając szczegółowych informacji



Rysunek 10: Oryginalny obraz (481×321) i wyniki segmentacji uzyskane z metody SuMC (k to liczba klastrów, zaś r to współczynnik kompresji danych). Rysunek pochodzi z pracy [A1].



Rysunek 11: Na rysunku przedstawiono (po lewej stronie) estymację parametru r dla syntetycznego zbioru danych, który składa się z trzech 2-wymiarowych podprzestrzeni (z 300, 400 i 500 losowo wybranymi punktami) oraz dwóch 1-wymiarowych podprzestrzeni (każda z 200 punktami). Dodatkowo (po prawej stronie), przedstawiono wyniki klastrowania przy użyciu metody SuMC dla oszacowanego parametru r . Rysunek pochodzi z pracy [A1].

na obrazie poprzez zwiększenie liczby klastrów (zob. rysunek 10d).

Warto również podkreślić, że model SuMC wymaga użycia jedynie dwóch parametrów, w szczególności parametru r , który określa poziom kompresji danych. Rozsądnie dobraną wartość tego parametru można uzyskać przy użyciu bayesowskiego kryterium informacyjnego (ang. Bayesian Information Criterion, BIC) [89]. Rysunek 11a przedstawia zależność między współczynnikiem kompresji r , a wartością BIC. Natomiast na rysunku 11b pokazano wynik grupowania dla tych samych danych, uwzględniając oszacowaną wartość parametru r . Jak można

zauważyć, model SuMC doskonale wyodrębnia poszczególne grupy, identyfikując podprzestrzenie, w których znajdują się punkty ze zbioru danych.

Wyniki przeprowadzonych eksperymentów świadczą o wysokiej skuteczności metody SuMC. Niemniej jednak, dostrzeżono pewne ograniczenia, które zostały poddane dalszym badaniom. Jednym z głównych wyzwań było czasochłonne wyszukiwanie odpowiednich baz podprzestrzeni. W związku z tym, w kolejnym badaniu skoncentrowano się na zoptymalizowaniu tego procesu bez większej utraty skuteczności metody.

Praca [A2] Grupowanie podprzestrzeni to efektywna klasa algorytmów, która doskonale radzi sobie z identyfikacją grup opisanych przez różnorodne podprzestrzenie afiniczne. Jednak arbitralny wybór tych podprzestrzeni może negatywnie wpływać na złożoność obliczeniową algorytmów, co częściowo ogranicza ich praktyczne zastosowanie w przypadku dużych zbiorów danych. Na szczęście istnieje alternatywna metoda grupowania, znana jako grupowanie rzutowe (ang. projected clustering) [3, 13], która znacząco redukuje koszt obliczeniowy. Grupowanie rzutowe wykorzystuje podprzestrzenie afiniczne równoległe do osi układu współrzędnych, co umożliwia bardziej efektywne przetwarzanie danych. W ramach tego podejścia, klastery są definiowane jako para dwóch zbiorów – jeden zawiera podzbiór punktów należących do danego klastra, a drugi opisuje ich atrybuty definiujące nisko-wymiarową podprzestrzeń. Dzięki temu każdy klastery jest opisany przez najbardziej istotne atrybuty, które są kluczowe dla danego zestawu danych.

W pracy [A2] opracowano szybki algorytm PMC (ang. Projected Memory Clustering) grupowania oparty na równoległych podprzestrzeniach afinicznych. Technika ta pozwala interpretować każdy klastery zgodnie z najbardziej informatywnymi cechami, podobnie jak w przypadku co-clusteringu [24, 40]. Analogicznie jak w poprzednich badaniach, zastanawiano się, jak znaleźć optymalną podprzestrzeń dla X_i i jak obliczyć odpowiednią funkcję błędów.

Okazuje się, że w przypadku podprzestrzeni równoległych do osi współrzędnych nie trzeba obliczać macierzy kowariancji danych. Jest to istotne, ponieważ jej obliczenie posiada złożoność sześcienną względem wymiaru przestrzeni, co znacznie zwiększa złożoność obliczeniową metody SuMC. W rezultacie, możemy teraz przeformułować twierdzenie 1 w następujący sposób.

Twierdzenie 2. Niech $N \in \mathbb{N}$, $X \subset \mathbb{R}^N$ będzie zbiorem danych, a $d_i = \text{Var}(X^i)$ oznacza wariancję i -tej współrzędnej X . Przez e_1, \dots, e_N oznaczamy bazę kanoniczną \mathbb{R}^N , zaś przez $d_{(1)} \geq \dots \geq d_{(N)}$ elementy d_i posortowane w kolejności malejącej. Dodatkowo, wektory $e_{(1)}, \dots, e_{(N)}$ odpowiadają posortowanym liczbą $d_{(i)}$.

Wówczas minimalny błąd średnio-kwadratowy związany z projekcją na m -wymiarową podprzestrzeń afiniczną postaci $v + \text{span}(e_J)$, gdzie $J \subset \{1, \dots, N\}$, $|J| = m$, jest dany wzorem

$$\text{error}(X, V) = |X| \cdot \sum_{i=m+1}^N d_{(i)} \quad \text{dla } m \leq N \quad (3)$$

gdzie podprzestrzeń V jest określona jako

$$V = \text{mean}_X + \text{span}(e_{(1)}, \dots, e_{(m)}).$$

D	Metoda						
	P3C [72]	PreDeCon [13]	PROCLUS [3]	ORGEN [111]	LSR [67]	SuMC	PMC
3	0.999	0.015	0.417	0.517	0.502	0.813	0.597
4	0.920	0.534	0.531	0.701	0.580	0.835	0.699
5	1.000	0.717	0.339	0.751	0.509	0.789	0.752
10	1.000	0.470	0.583	0.587	0.718	0.741	0.329
20	0.009	1.000	0.545	0.933	0.998	0.795	0.780
30	—	0.710	0.398	0.805	0.742	0.979	1.000
35	—	0.696	0.616	0.571	0.662	0.839	0.389
40	—	0.977	0.614	0.771	0.897	0.856	1.000
45	—	1.000	1.000	0.851	0.768	0.919	1.000
50	—	0.712	0.637	0.784	0.851	0.962	1.000
60	—	1.000	1.000	1.000	0.963	1.000	1.000
70	—	0.994	0.928	0.924	0.897	0.989	1.000
80	—	1.000	0.524	0.458	0.393	1.000	1.000
90	—	1.000	0.260	0.418	0.496	0.791	0.979
100	—	0.936	0.594	0.621	0.719	1.000	1.000

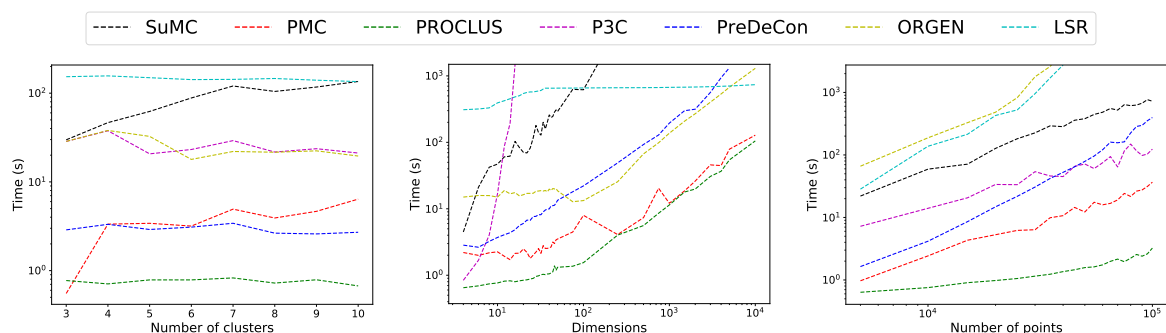
Tabela 2: Porównanie PMC i innych metod na syntetycznych zbiorach danych z wykorzystaniem Adjusted Rand Index. Tabela pochodzi z prac [A2].

Zbiór danych	Metoda						
	P3C [72]	PreDeCon [13]	PROCLUS [3]	ORGEN [111]	LSR [67]	SuMC	PMC
20news	—	—	0.040	0.078	0.023	—	0.109
5 – HT_{1A} [103]	—	—	0.218	0.189	0.175	—	0.428
CIFAR-100 [56]	—	—	0.349	—	0.205	—	0.412
farm-ads [63]	—	—	0.015	0.045	0.063	—	0.109
Gas [63]	—	0.519	0.553	0.982	0.488	0.774	0.472
Gesture [63]	—	0.172	0.060	0.147	0.154	0.179	0.179
MNIST [59]	—	—	0.292	0.397	0.196	0.408	0.410
Pendigits [63]	0.009	0.537	0.460	0.645	0.535	0.598	0.466
Seeds [63]	0.003	0.002	0.420	0.676	0.826	0.732	0.683

Tabela 3: Porównanie PMC i innych metod na rzeczywistych zbiorach danych z wykorzystaniem Adjusted Rand Index. Tabela pochodzi z pracy [A2].

Eksperymenty przeprowadzone na zbiorach danych syntetycznych (zob. tabela 2) i rzeczywistych (zob. tabela 3) wykazały, że proponowane podejście lepiej odwzorowuje pierwotną strukturę danych (mierzoną za pomocą Adjusted Rand Index (ARI) [48]) w porównaniu z innymi metodami grupowania. Co więcej, metoda ta jest w większości przypadków lepsza od metod klastrowania podprzestrzennego, szczególnie dla danych wielowymiarowych, co ma duże znaczenie praktyczne dla dużych zbiorów data.

W przypadku danych syntetycznych rozważano zbiory danych składające się z 10 000 punktów, które zostały wygenerowane jako suma k równoległych podprzestrzeni do osi układów współrzędnych zawartych w przedziale $[0, 1]^D$, gdzie wartość D zmieniano od 3 do 100. Liczba



Rysunek 12: Czasy przetwarzania dla rozważanych metod ze względu na liczbę klastrów, wymiar danych oraz liczbę punktów. Rysunek pochodzi z pracy [A2].

podprzestrzeni k była losowana ze zbioru wartości $\{2, \dots, D - 1\}$. Wyniki dla 100 losowych syntetycznych danych i dla każdej wartości D przedstawia tabela 2.

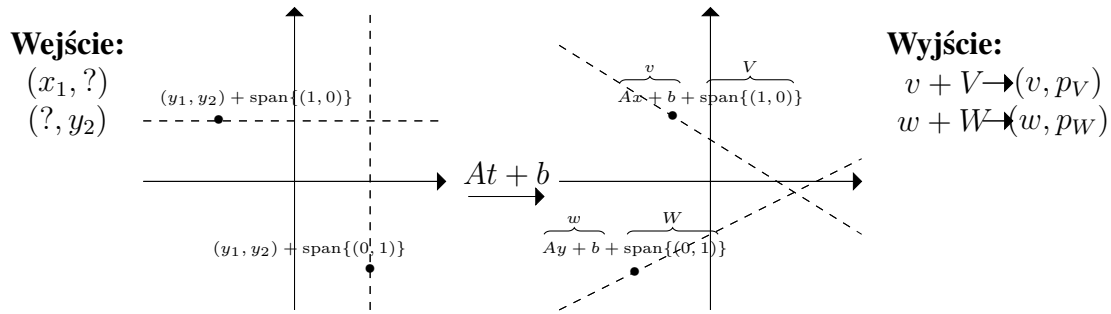
Wyniki przedstawione na rysunku 12 pokazują, że metoda PMC ma drugi najlepszy czas przetwarzania. Jest ona tylko nieznacznie wolniejsza niż metoda PROCLUS, ale z kolei osiąga lepsze od niej wyniki (zob. tabele 2 i 3). W rzeczywistości oba algorytmy mają taką samą teoretyczną złożoność obliczeniową.

4.1.2 Reprezentacja niekompletnych danych [A3]

Analiza niekompletnych danych jest istotnym zagadnieniem w inżynierii danych i uczeniu maszynowym, ponieważ braki w danych są powszechnym problemem w wielu dziedzinach, takich jak medycyna diagnostyczna [17], rozpoznawanie obiektów [8] i analiza chemiczna [95]. Imputacja, czyli uzupełnienie brakujących wartości, jest jedną z najpopularniejszych technik radzenia sobie z tym problemem [69], gdyż pozwala ona na zastosowanie standardowych metod uczenia maszynowego do pełnych danych. Najprostszym podejściem do imputacji jest zastąpienie brakujących wartości średnimi lub zerami. Istnieją również bardziej zaawansowane techniki imputacji [23, 30, 65], jednak większość z tych metod wiąże się z utratą informacji o brakujących atrybutach.

Brakujące dane są zwykle reprezentowane jako para (x, J_x) , gdzie $x \in \mathbb{R}^N$ jest wektorem danych z brakującymi składnikami, a $J_x \subset \{1, \dots, N\}$ określa indeksy brakujących atrybutów. Aby zachować informacje o brakujących atrybutach, można wykorzystać wektor flag binarnych [19], który wskazuje nieobecne składowe. W przypadku danego wektora x , wektor binarny $\mathbb{1}_{J_x}$ zawiera 1 na pozycjach odpowiadających brakującym atrybutom i 0 na pozycjach odpowiadających obecnym atrybutom.

W praktycznych problemach klasyfikacyjnych, przed nauczeniem klasyfikatora, często wykonuje się różne afiniczne przekształcenia danych, takie jak wybielanie lub redukcja wymiarowości. Wprowadzenie takich przekształceń wpływa również na wektory flag, które wskazują na brakujące składowe. Pojawia się zatem problem znalezienia takiego sposobu przekształcania wektorów flag w przypadku liniowych lub afinicznych odwzorowań danych, który zachowałby ich spójność.



Rysunek 13: Reprezentacja niekompletnych danych, ich transformacja afiniczna oraz imputacja jako rzutowanie na podprzestrzeń. Rysunek pochodzi z pracy [A3].

W pracy [A3] udowodniono, że powyższy problem można rozwiązać traktując niekompletne dane jako podprzestrzeń afiniczną. Innymi słowy, para (x, J_x) może być formalnie powiązana z podprzestrzenią afiniczną w \mathbb{R}^N zadaną jako $x + \text{span}\{e_j\}_{j \in J_x}$, gdzie $(e_j)_{j=1}^N$ oznacza bazę kanoniczną w \mathbb{R}^N , a x jest dowolnym punktem w \mathbb{R}^N , którego współrzędne pokrywają się poza zbiorem J_x . W konsekwencji, przez *uogólniony brakujący punkt danych* w \mathbb{R}^N rozumiano podprzestrzeń afiniczną $S_x = x + V \subset \mathbb{R}^N$, gdzie $V = S_x - x$ to podprzestrzeń liniowa.

Pokazano także, że powyższa definicja niekompletnych danych jest użyteczna do definiowania odwzorowań liniowych na niekompletnych danych. Na przykład dla odwzorowania liniowego $f : \mathbb{R}^N \ni w \rightarrow Aw + b$ można przekształcić uogólniony brakujący punkt danych $S_x = x + V$ w inny brakujący punkt danych w następujący sposób:

$$f(S_x) = \{Aw + b : w \in x + V\}.$$

Punkt x jest odwzorowywany na punkt $Ax + b$, podczas gdy liniowa część $f(S_x)$ jest dana przez $f(x + V) - f(x) = AV$. Przykładem takiej transformacji może być transformacja wybielania (ang. whitening transformation) [52], którą możemy zdefiniować jako

$$\text{Whitening}(x) = \Sigma^{-1/2}(x - m),$$

gdzie Σ to macierz kowariancji, a m średnia ze zbioru danych. Dla uogólnionych niekompletnych danych zdefiniowano tę operację jako

$$\text{Whitening}(x + V) = \Sigma^{-1/2}(x - m) + \Sigma^{-1/2}V.$$

Innymi słowy, przekształcamy punkt x w sposób klasyczny, a następnie transformujemy podprzestrzeń V w liniową podprzestrzeń $\Sigma^{-1/2}V$.

Powyższa reprezentacja jest użyteczna do zrozumienia i realizacji przekształceń afinicznych na niekompletnych danych. Niemniej jednak, typowe metody uczenia maszynowego wymagają wektorów lub pewnego rodzaju macierzy jądra (lub podobieństw) jako danych wejściowych. Dlatego też w pracy [A3] zaprezentowano sposób osadzenia uogólnionych brakujących danych w przestrzeni wektorowej poprzez identyfikację podprzestrzeni liniowej V za pomocą rzutu ortogonalnego $p_V : \mathbb{R}^N \rightarrow V$.

Dodatkowo, zaproponowano nową metodę imputacji, która wykorzystuje najbardziej prawdopodobne wartości do uzupełnienia brakujących atrybutów. Biorąc pod uwagę średnią m i

Dane z [63]	nasza [A3]	mice [18]	geom [20]	karma [44]	genRBF [B6]
australian	0.867	0.836	0.800	0.845	0.843
breast-cancer	0.955	0.951	0.952	0.943	0.950
bank	0.966	0.942	0.872	0.965	0.952
crashes	0.936	0.916	0.913	0.929	0.924
heart	0.829	0.802	0.808	0.826	0.827
liver disorders	0.687	0.685	0.584	0.680	0.662
pima	0.791	0.735	0.701	0.775	0.770
phoneme	0.851	0.811	0.763	0.836	0.838
ring	0.973	0.961	0.886	0.965	0.966
sonar	0.892	0.620	0.594	0.614	0.812

Tabela 4: Uzyskana dokładność klasyfikacji, w przypadku gdy usuwanie atrybutów było częściowo determinowane przez odpowiadające im etykiety klas. Tabelka pochodzi z pracy [A3].

macierz kowariancji Σ oszacowaną na podstawie kompletnych danych, zdefiniowano imputację dla punktu x jako rzutowanie

$$x - p_V^\Sigma(x - m)$$

na podprzestrzeń $x + V$, w odniesieniu do iloczynu skalarnego Mahalanobisa zadanego przez Σ (zob. rysunek 13). Z wielu przeprowadzonych eksperymentów wyraźnie wynika, że podejście oparte na podprzestrzeni zazwyczaj osiąga najwyższą wydajność spośród wszystkich rozważanych metod. W tabeli 4 przedstawiono wyniki jednego z takich eksperymentów, gdzie brak atrybutu był indukowany przez etykietę punktu.

4.2 Eksploracja i interpretacja danych za pomocą głębokich modeli

4.2.1 Modele generatywne [A4, A5, A6, A7]

W ostatnich latach modele generatywne wykorzystujące zmienne ukryte stały się popularnym tematem badań. Istnieje wiele różnych modeli generatywnych, w tym VAE [54] oraz GAN [39]. Oprócz tych podstawowych rozwiązań, rozwijane były także inne innowacyjne podejścia, takie jak na przykład model Glow [53].

W modelach generatywnych rozkład danych jest odwzorowywany w ukrytej przestrzeni (ang. latent space). Poprzez dekodowanie próbek z rozkładu ukrytego, można generować nowe obiekty, które pasują do rozkładu danych rzeczywistych. Ponadto, poprzez interpolację między różnymi reprezentacjami dwóch różnych obiektów w ukrytej przestrzeni, można odkrywać obiekty, które łączą w sobie cechy obu z nich. Taka elastyczność modeli otwiera wiele możliwości eksploracji i generowania nowych, interesujących danych. Ponadto, interpolacja pozwala ocenić jakość modeli [29, 58, 83, 93, 108], poprawić proces uczenia w przypadku niewielkich zbiorów danych [100] oraz zbadać geometrię przestrzeni ukrytej [5, 57, 81]. Dodatkowo, dekodowane interpolacje odgrywają istotną rolę w analizie zdolności generalizacji sieci [15, 28].



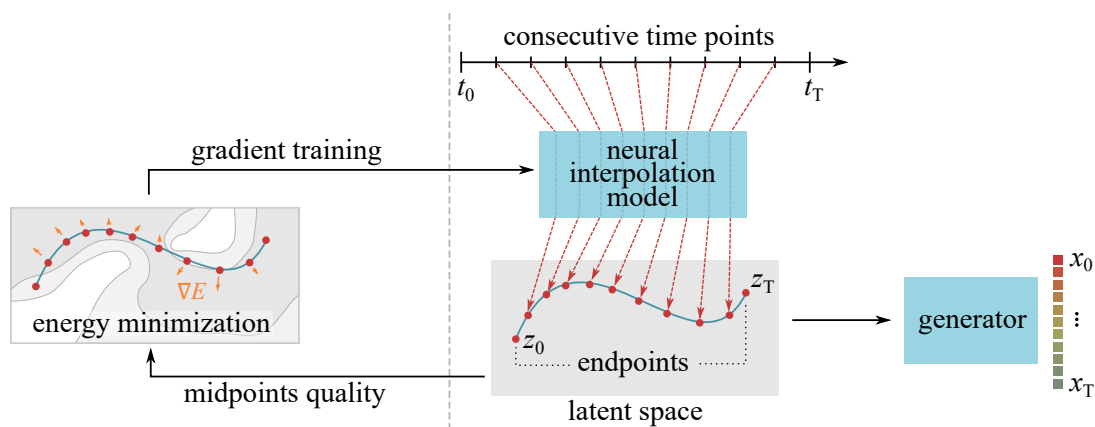
Rysunek 14: Przykładowe interpolacje w modelu DCGAN trenowanym na podkowie w \mathbb{R}^8 . W każdej parze górny rząd reprezentuje ścieżkę liniowej interpolacji, podczas gdy dolny pokazuje wynik proponowanej procedury optymalizacyjnej. Po prawej stronie podane są odpowiednie projekcje rozkładu na przestrzeni ukrytej (niebieskie kropki), interpolacje liniowe (na zielono) i nasze (linie czerwone, szare linie pokazują pośrednie etapy optymalizacji). Rysunek pochodzi z pracy [A4].

Interesującym zagadnieniem jest to, czy (jak sugerowano w [11, 27, 84, 105]) ukryta przestrzeń wytrenowanego modelu generatywnego oddaje semantyczną zawartość przestrzeni danych, włączając w to jej strukturę. Istotne jest, aby (jak wskazano w [6]) natura przestrzeni ukrytej, która jest w istocie różniczkowością Riemanna z lokalnie zdefiniowanym, płynnie zmieniającym się iloczynem skalarnym [26], odzwierciedlała tę cechę. W tym kontekście autorzy [6] proponują wykorzystanie geodezyjnej na różniczkowości Riemanna jako najkrótszej ścieżki podczas interpolacji między punktami. Rodzi się zatem pytanie o rodzaj interpolacji, jakich należy użyć w konkretnym modelu.

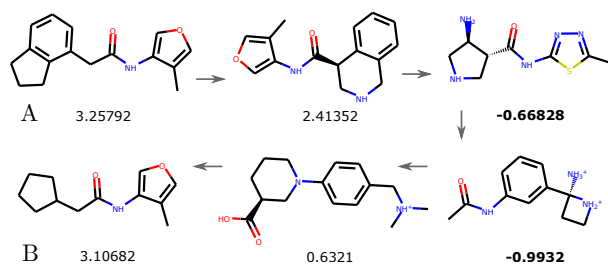
Kluczowym wnioskiem przedstawionym w [84] jest to, że przechodzenie przez ukrytą przestrzeń powinno prowadzić do semantycznych zmian. Te zmiany mogą objawiać się w postaci modyfikacji lub dodawania pewnych cech generowanych obiektów. Wykazano, że arytmetyka wektorowa w przestrzeni ukrytej ma odpowiednie znaczenie semantyczne w przestrzeni wyjściowej. Na przykład, punkty ukrytej przestrzeni reprezentujące „mężczyznę w okularach”, po odjęciu „mężczyzny bez okularów” i dodaniu „kobiety bez okularów”, generują obraz kobiety w okularach.

W pracy [A4] przedstawiono ogólne i ujednoczone podejście do interpolacji, które jednocześnie umożliwia poszukiwanie krzywych geodezyjnych i interpolacyjnych w przestrzeni ukrytej dla dowolnej gęstości, jak pokazano na rysunku 14. Wyniki, jakie przedstawiono w pracy [A4], są oparte na solidnych podstawach teoretycznych i bazują na wprowadzonej mierze jakości krzywej interpolacyjnej [A4, sekcje II i III]. Badania prezentowane w pracy pokazują, że maksymalizacja tej miary jest równoważna poszukiwaniu geodezyjnej [A4, sekcja IV].

W celu wygenerowania krzywej interpolacji wykorzystano sieć neuronową f_θ (zob. rysunek 15). Dla sekwencji skalarów $0 = t_0 < \dots < t_n = 1$, które odpowiadają pojedynczej partii batcha, sieć f_θ generuje sekwencję ukrytych punktów $z_i = f_\theta(t_i)$, które reprezentują zoptymalizowaną ścieżkę między dwoma punktami z_0 i z_1 . Aby spełnić warunki $\gamma_\theta(0) = z_0$ oraz



Rysunek 15: Interpolująca sieć odwzorowuje sekwencję punktów (po jednym na raz) z przedziału (t_0, \dots, t_T) , gdzie punkty końcowe t_0 i t_T odpowiadają obiektom danych wejściowych, między którymi przeprowadzana jest interpolacja. Sieć minimalizuje funkcję energii całej krzywej indukowanej na podstawie wcześniej zdefiniowanej funkcji jakości, np. gęstość prawdopodobieństwa danych. Rysunek pochodzi z pracy [A4].



Rysunek 16: Przykładowa zoptymalizowana ścieżka między związkami A i B, która minimalizuje współczynnik podziału oktanol/woda $\log P$ (wartości pokazane poniżej związków). Nasza metoda optymalizuje ścieżkę interpolacji w przestrzeni ukrytej wytrenowanego chemicznego modelu generatywnego, aby znaleźć podobne związki o zminimalizowanej wartości $\log P$. Rysunek pochodzi z pracy [A4].

$\gamma_\theta(1) = z_1$, zastosowano następującą transformację:

$$\gamma_\theta(t) = (1 - t)(z_0 - f_\theta(0)) + t(z_1 - f_\theta(1)) + f_\theta(t).$$

Parametry sieci θ są poprawiane w sposób iteracyjny poprzez wsteczną propagację do momentu, aż zostanie osiągnięta jego zbieżność. W każdej iteracji generowana jest losowa sekwencja n punktów $\{t_i\}_{i=1..n}$, spełniająca następujące warunki $t_i < t_{i+1}$ oraz $t_0 = 0$ i $t_n = 1$.

Przeprowadzono szereg eksperymentów na różnych modelach generatywnych, takich jak DCGAN [83], Glow [53], Pioneer [46], StackGAN [113], a także na różnych zbiorach danych, włączając MNIST [60], SVHN [75], CelebA [66], CUB-200 [104] oraz dane chemiczne. Wyniki wskazują, że proponowany model jest skuteczny dla różnych rozkładów danych. Dzięki proponowanemu podejściu mamy możliwość odkrywania nowych obiektów i struktur w danych, poprzez interpolację między różnymi przykładami. Na przykład, możemy badać jak zmieniają się twarze od jednego ujęcia do drugiego, czy też jak zmienia się struktura chemiczna związków



Rysunek 17: Przykładowa interpolacja między identycznymi obrazami przy użyciu różnych modeli generatywnych. Pierwsze dwa rzędy pokazują interpolacje dla wstępnie wytrenowanego modelu Pioneer, podczas gdy ostatnie dwa używają pre-trenowanego modelu Glow. Na prawej stronie obrazka zdefiniowano cechę, która jest pożądana w ścieżce interpolacyjnej. Rysunek pochodzi z pracy [A4].

(zob. rysunek 16). Ponadto, model umożliwia elastyczną kontrolę nad procesem interpolacji, co pozwala na tworzenie płynnych przejść między różnymi stanami danych.

Proponowane podejście pozwala również na wykonywanie interpolacji dla tego samego obrazu, umożliwiając manipulację pożądanymi cechami. Na przykład, jeśli mamy obraz mężczyzny bez brody, możemy wykonać interpolację dodając mu brodę, postarzając go lub odmładzając, a nawet zmieniając mu płeć (zob. rysunek 17). Dzięki temu uzyskujemy elastyczność w eksploatacji różnych wariantów tego samego obiektu, co jest szczególnie cenne w przypadku analizy obrazów i manipulacji ich cechami.

Praca [A5] W kolejnych badaniach skupiono się na analizie przestrzeni niekompletnych obrazów w celu ich uzupełnienia (ang. image imputation). W pracy [A5] przedstawiono metodę DMFA (Deep Mixture of Factor Analyzers), która wykorzystuje głęboką sieć neuronową oraz techniki mieszaniny rozkładów normalnych (Gaussian Mixture Model, GMM) do estymacji funkcji gęstości prawdopodobieństwa brakujących wartości.

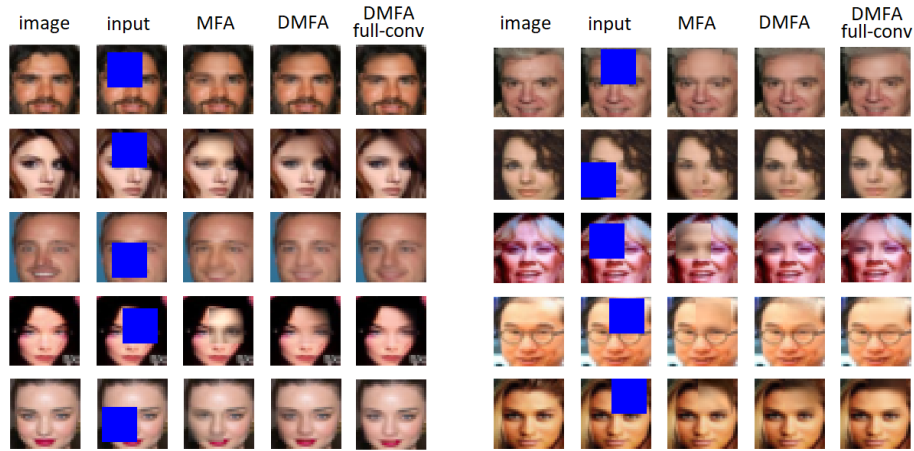
Podczas badań przeprowadzono analizę danych obrazowych o znacznych rozmiarach, które zawierały brakujące piksele. Standardowe zastosowanie metody GMM do takich danych jest mało efektywne i może prowadzić do wielu problemów numerycznych [87]. Dane obrazowe zazwyczaj są osadzone w przestrzeniach o niższych wymiarach, co sugeruje, że pełna macierz kowariancji może być niepotrzebna. Dlatego też w pracy [A5] zastosowano metodę mieszaniny czynników składowych (Mixture of Factor Analyzers, MFA) [36], która pozwala przedstawić macierz kowariancji jako

$$\Sigma = AA^T + D,$$

gdzie A jest macierzą składającą się z n czynników (n jest znacznie mniejsze niż wymiar danych), a D jest macierzą diagonalną. Zastosowanie MFA znacznie redukuje liczbę parametrów

Dane	Miara	MFA	DMFA
MNIST	NLL	58.10	-244.81
	MSE	18.59	12.96
Fashion-MNIST	NLL	-85.15	-252.49
	MSE	6.12	6.03
CelebA	NLL	-882.54	-1222.85
	MSE	9.82	7.73

Tabela 5: Ujemny logarytm wiarygodności (ang. Negative Log-Likelihood, NLL) i błąd średniokwadratowy (ang. Mean Square Error, MSE) najbardziej prawdopodobnej imputacji otrzymanej przez DMFA i MFA [87] (im mniejszy, tym lepszy). Tabela pochodzi z pracy [A5].



Rysunek 18: Przykładowe wyniki imputacji wygenerowane przez MFA [87] i dwie wersje modelu DMFA. Rysunek pochodzi z pracy [A5].

w macierzy kowariancji, co jednocześnie rozwiązuje problem z jej przechowywaniem i odwracaniem.

Niech $x = (x_o, x_m)$ będzie wektorem reprezentującym niepełny obraz, gdzie x_o reprezentuje znane wartości pikseli, zaś x_m opisuje nieobecne piksele. Wówczas naturalne jest pytanie o rozkład brakujących wartości w kontekście dostępnych danych obserwowanych. W przypadku modeli gęstości odpowiedzią jest tu gęstość warunkowa [37]

$$p(x_m|x_o) = \frac{p(x_o, x_m)}{p(x_o)} = \frac{p(x)}{p(x_o)}.$$

W przeciwieństwie do wielu innych głębokich modeli generatywnych, takich jak GAN (Generative Adversarial Network) [39] lub VAE (Variational Autoencoder) [54], dla GMM możliwe jest znalezienie analitycznego wzoru na gęstość warunkową. Dla jednego rozkładu normalnego $N(\mu, \Sigma)$, gdzie $\mu = \begin{pmatrix} \mu_o \\ \mu_m \end{pmatrix}$ oraz $\Sigma = \begin{pmatrix} \Sigma_{oo} & \Sigma_{om} \\ \Sigma_{om}^T & \Sigma_{mm} \end{pmatrix}$, warunkowa gęstość jest dana

następującym wzorem:

$$p(x_m|x_o) = N(\hat{\mu}_m, \hat{\Sigma}_m),$$

gdzie

$$\begin{aligned}\hat{\mu}_m &= \mu_m + \Sigma_{om}\Sigma_{oo}^{-1}(x_o - \mu_o), \\ \hat{\Sigma}_m &= \Sigma_{mm} - \Sigma_{om}\Sigma_{oo}^{-1}\Sigma_{om}^T.\end{aligned}\tag{4}$$

Celem modelu DMFA jest wypełnienie brakujących pikseli, których liczba może się różnić dla różnych obrazów. Dlatego też głęboka sieć estymująca rozkład danych musi obliczać parametry rozkładu normalnego $N(\mu, \Sigma)$ dla wszystkich pikseli. Jednakże rozkład $N(\mu, \Sigma)$ nie musi reprezentować rozkładu na całej przestrzeni danych. Zatem nie musimy korzystać z równań gęstości warunkowej (4), ale możemy ograniczyć μ i Σ do brakujących pikseli w celu zdefiniowania gęstości warunkowej $p(x_m|x_o) = N(\mu_m, \Sigma_{mm})$, gdzie $\Sigma_{mm} = A_m A_m^T + D_{mm}$ dla zawężonych macierzy A i D , które zawierają tylko wiersze odpowiadające brakującym atrybutom. Zatem wystarczy trenować model DMFA poprzez minimalizację ujemnego logarytmu wiarygodności gęstości warunkowej $p(x_m|x_o)$, która jest zdefiniowana jako

$$-\log p(x_m|x_o) = -\log N(\mu_m, \Sigma_{mm})(x_m).$$

W rezultacie otrzymana gęstość rozkładu normalnego powinna charakteryzować się lepszą jakością w odniesieniu do brakujących danych niż rozkład warunkowy obliczony z klasycznej metody GMM.

Przeprowadzono wiele eksperymentów potwierdzających powyższą tezę na znanych zbiorach danych, takich jak MNIST [59], Fashion-MNIST [109] i CelebA [66]. Zaproponowana metoda DMFA osiąga lepsze wyniki (tzn. większy log-likelihood) dla brakujących wartości w porównaniu do MFA [87] (zob. tabela 5). Ponadto, imputacje uzyskane poprzez zastąpienie brakujących wartości wektorem średniej rozkładu wyznaczonego przez model DMFA prezentują wyraźnie lepsze rezultaty niż MFA, co można zaobserwować na rysunku 18.

Praca [A6] W pracy [A6] skupiono się na porównaniu wytrenowanych modeli generatywnych. Dwie najbardziej znane metryki, które oceniają wydajność modelu generatywnego pod względem zdolności generowania, to Inception Score (IS) [88] i Fréchet Inception Distance (FID) [47]. Miary te mają pewne ograniczenia. W szczególności, wiele modeli generatywnych jest szkolonych przy użyciu dolnego ograniczenia log-likelihood, takiego jak ELBO (Evidence Lower Bound), jak np. rodzina modeli VAE [54, 86], co nie zawsze dokładnie odzwierciedla rozkłady danych [16]. Ponadto, FID opiera się na przybliżeniu wyników przy użyciu rozkładów normalnych, co często nie pokrywa się z rzeczywistością [92].

W świetle powyższych uwag, w pracy [A6] zaproponowano skuteczne górne ograniczenie, które umożliwia efektywne przybliżenie wartości log-wiarygodności danych, co pozwala na lepsze porównanie jakości danych generowanych przez różne modele. Przechodząc zatem do przedstawienia problemu w sposób bardziej formalny, wprowadźmy odpowiednie oznaczenia i definicje.

Niech X będzie zmienną losową. Z klasycznej nierówności Jensena [35] dla każdej funkcji wklęsłej f mamy $f(\mathbb{E}X) \geq \mathbb{E}f(X)$. Naszym celem jest uzyskanie (przy powyższych ogólnych

założeniach) górnego oszacowania dla $f(\mathbb{E}X)$, które wymagałoby obliczenia jedynie wartości oczekiwanej jakiejś zmiennej losowej zależnej od X . W pracy [A6] udowodniono następujące twierdzenie.

Twierdzenie 3. *Niech f będzie gładką funkcją wklęsłą. Wtedy*

$$f(\mathbb{E}X) \leq \mathbb{E}(f(X) + (Y - X)f'(X)), \quad (5)$$

gdzie X i Y to dwie niezależne zmienne losowe o tym samym rozkładzie.

Ponieważ głównym celem jest oszacowanie wartości log-wiarygodności danych, założmy, że $f = \log$. Korzystając z twierdzenia 3 oraz nierówności Jensena, stwierdzono, że wartość $f(\mathbb{E}X)$ jest zawarta w przedziale

$$\mathbb{E} \log X \leq \log \mathbb{E}X \leq \mathbb{E} \log X + \mathbb{E} \frac{Y}{X} - 1, \quad (6)$$

gdzie X, Y to dwie niezależne zmienne losowych o tym samym rozkładzie. Dodatkowo udowodniono, że dla dowolnych funkcji f, g, h spełniających nierówność $f(a) \leq g(x) + ah(x)$, gdzie $a, x \in \mathbb{R}$, oraz dwóch niezależnych zmiennych losowych X, Y o tym samym rozkładzie, zachodzi poniższa nierówność:

$$f(\mathbb{E}X) \leq \mathbb{E}(g(X) + Yh(X)),$$

co z kolei prowadzi (przyjmując $g(X) = \log X - 1 + C$) do następujących oszacowań:

$$\mathbb{E} \log X \leq \log \mathbb{E}X \leq \mathbb{E} \log X - 1 + C + \exp(-C) \mathbb{E} \frac{Y}{X}. \quad (7)$$

Ostatecznie, stosując technikę próbkowania ważności (ang. importance sampling) w celu uzyskania dokładniejszego addytywnego przybliżenia $f(\mathbb{E}[X])$, możemy zastąpić zmienną X przez $\bar{X} = \frac{1}{k}(X_1 + \dots + X_k)$, gdzie X_1, \dots, X_k są próbkami zmiennej X . Rozważając wariacyjne modele generatywne, takie jak VAE, gdzie:

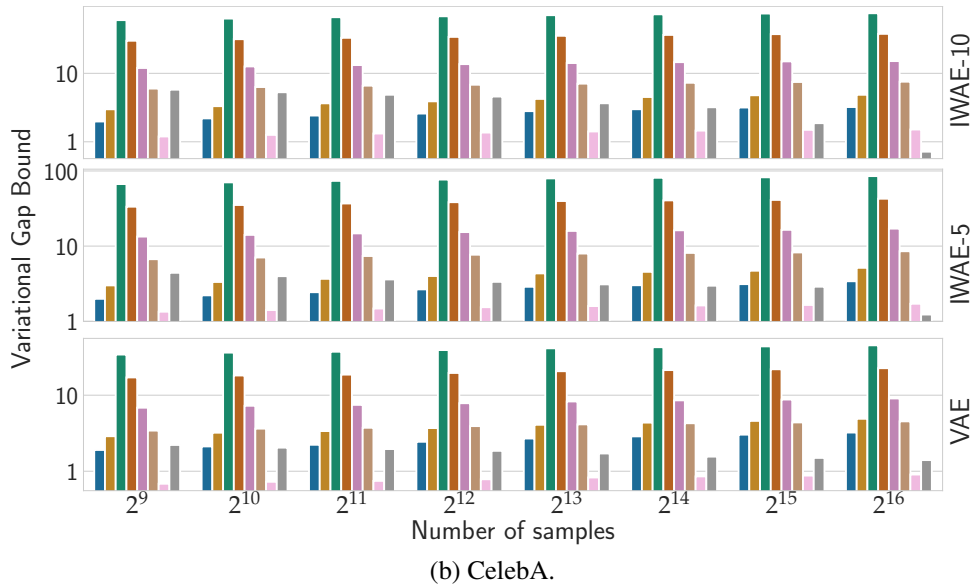
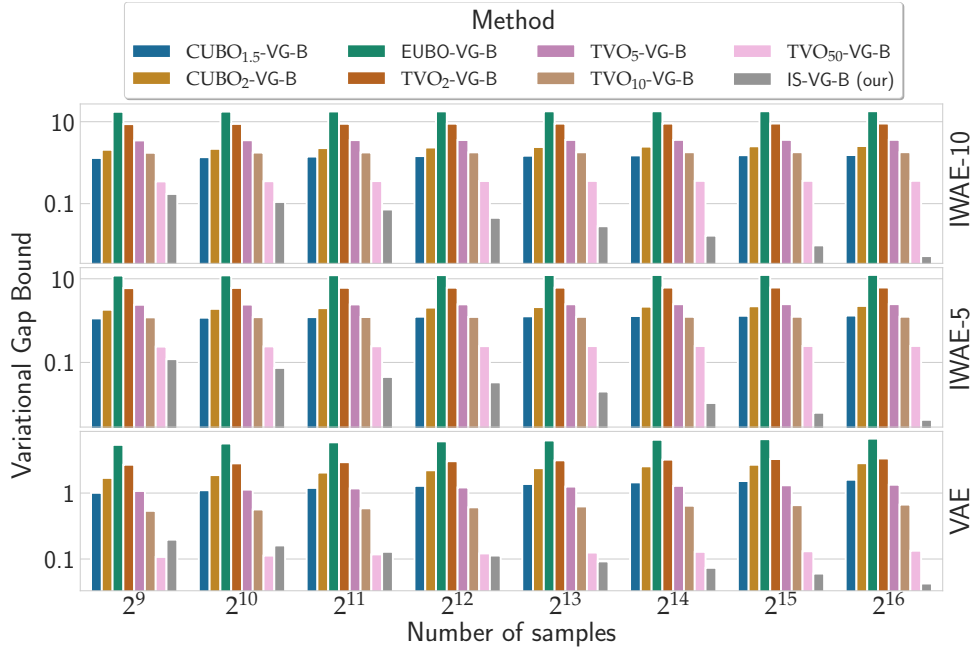
- $p(x)$ oznacza rozkład indukowany przez model VAE w przestrzeni wejściowej $X = \mathbb{R}^N$,
- $p(z)$ oznacza rozkład a priori w przestrzeni ukrytej Z ,
- $q(z|x)$ jest wariacyjnym enkoderem,
- $p(x|z)$ jest dekoderem wariacyjnym,

zdefiniowano dolne ograniczenie na log-likelihood jako

$$\text{IW-ELBO}_k = \mathbb{E}_{z_i \sim q(\cdot|x)} \log \frac{1}{k} \sum_{i=1}^k R(x, z_i),$$

gdzie $R(x, z) = \frac{p(x|z)p(z)}{q(z|x)}$. Natomiast górne ograniczenie zdefiniowano przez sumę dolnego ograniczenia na log-likelihood i górnego oszacowania dla luki wariacyjnej (ang. variational gap), które jest zdefiniowane wzorem

$$\text{IS-VG-B} = C - 1 + \exp(-C) \mathbb{E}_{z_i, \tilde{z}_i \sim q(\cdot|x)} \frac{\sum_{i=1}^k R(x, \tilde{z}_i)}{\sum_{i=1}^k R(x, z_i)},$$



Rysunek 19: Porównanie różnych oszacowań górnych rozmiaru luki wariacyjnej dla modeli VAE, IWAE-5 i IWAE-10, które zostały wcześniej wytrenowane na zbiorach danych MNIST i CelebA. Wszystkie obliczenia zostały uśrednione dla 3 zestawów próbek. Rysunki pochodzą z pracy [A6].

gdzie C to dowolna wartość dodatnia, z optimum równym

$$\log \mathbb{E}_{z_i, \tilde{z}_i \sim q(\cdot|x)} \frac{\sum_{i=1}^k R(x, \tilde{z}_i)}{\sum_{i=1}^k R(x, z_i)}.$$

Dla IS-VG-B oraz innych istniejących oszacowań górnych, takich jak CUBO-VG-B [25], EUBO-VG-B i TVO-VG-B [68] przeprowadzono serię eksperymentów na różnych zbiorach da-

Metoda	Rozdzielczość			
	64×64	128×128	256×256	512×512
LocoGAN	956 MB	956 MB	956 MB	956 MB
DCGAN	301 MB	691 MB	2090 MB	7539 MB

Tabela 6: Zużycie pamięci dla modelu LocoGAN do tworzenia obrazów o różnych rozdzielczościach z kawałków o stałej wielkości 64×64 . Dla porównania pokazano, jak pamięć ta wzrasta dla architektur DCGAN [83]. Tabela pochodzi z pracy [A7]

nych: MNIST [59], SVHN [75] i CelebA [66], wykorzystując różne modele VAE [54], IWAE-5 i IWAE-10 [16]. Wyniki uzyskane w tych eksperymentach potwierdzają większą skuteczność proponowanego w pracy [A6] rozwiązania w porównaniu do innych rozważanych, co można zaobserwować na rysunku 19. Jedyne dla modeli VAE trenowanych na zbiorach danych SVHN i CelebA, oszacowanie TVO₅₀-VG-B uzyskało nieznacznie lepsze oszacowania. Ponadto wyniki te wskazują, że oszacowanie proponowane w pracy [A6] jest jedynym, które gwarantuje zmniejszenie obliczonej luki wariacyjnej wraz ze wzrostem liczby próbek (w przypadku innych oszacowań może ona się nawet) nieznacznie zwiększać.

Praca [A7] W kolejnej pracy [A7] przedstawiono lokalny model generatywny LocoGAN (Locally Convolutional GAN), który w odróżnieniu od standardowych generatywnych modeli adwersarialnych (ang. Generative Adversarial Networks, GAN) umożliwia generowanie obrazów o dowolnej rozdzielczości.

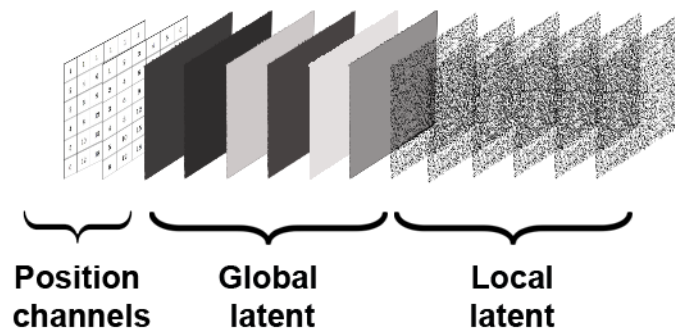
W klasycznym modelu GAN [39] przestrzeń ukryta jest reprezentowana przez wektor losowych wartości, który jest przekształcany w obraz o określonej rozdzielczości. W rezultacie, model GAN jest trenowany tylko dla jednej konkretnej rozdzielczości i generuje obrazy o tej samej wielkości. Stwarza to problem związany z wysokimi kosztami obliczeniowymi i ograniczeniami pamięciowymi (modele dla większych rozdzielczości wymagają większej liczby parametrów sieci, zob. tabela 6).

Model LocoGAN opiera się na innowacyjnej koncepcji uogólnionej przestrzeni ukrytej. W przeciwieństwie do tradycyjnych modeli GAN, które korzystają z wektora szumu o stałej wielkości, model LocoGAN wykorzystuje szum w postaci trójwymiarowego tensora o strukturze przypominającej obraz i zdolnego przyjąć dowolny rozmiar. Dzięki temu jest on w stanie generować zarówno pojedyncze fragmenty obrazu, jak i pełne obrazy.

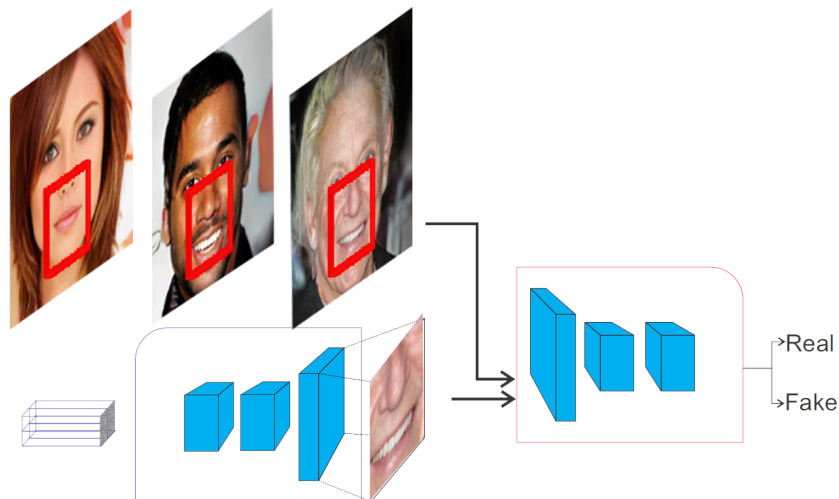
Aby zapewnić spójność fragmentów obrazów i uniknąć sytuacji, w których generator stworzyłby obraz z niepasującymi do siebie elementami, podzielono liczbę klatek na dwie części. Inspiracją tego pomysłu była praca [9], gdzie podobne rozwiązanie zostało zastosowane do generowania tekstur. Pierwsza część zawiera „globalny szum”, który koduje globalne cechy obrazu (jedna wartość dla całego kanału), takie jak kolor skóry czy kształt twarzy. Druga część natomiast zawiera „lokalny szum”, który odpowiada za generowanie lokalnych szczegółów (różny dla każdego piksela), na przykład zmianę położenia pasma włosów (zob. rysunek 20). Dodatkowo, na dwóch pierwszych kanałach dodano tzw. kanały pozycyjne, które przechowują



Rysunek 20: Rysunek przedstawia kilka wygenerowanych obrazków 128×128 z tym samym globalnym szumem, ale losowym szumem lokalnym. Widzimy, że lokalna modyfikacja szumu wprowadza zmiany (różne położenie włosów) do obrazu przy zachowaniu jego głównych cech. Rysunek pochodzi z pracy [A7].



Rysunek 21: Uogólniona reprezentacja szumu przypominająca strukturę obrazu. Globalny szum reprezentuje cechy globalne i jest taki sam dla każdego piksela w danym kanale. Lokalny szum to szum gaussowski bez korelacji między pikselami. Rysunek pochodzi z pracy [A7].



Rysunek 22: Schemat uczenia modelu LocoGAN. Podczas procesu uczenia wykorzystuje się fragmenty obrazka o stałej rozdzielczości, na przykład 64×64 piksele. Zarówno generator, jak i dyskryminator, otrzymują dane wejściowe, które zawierają dodatkowe kanały współrzędnych. Te kanały informują sieć o tym, jaką część obrazu przetwarza. Rysunek pochodzi z pracy [A7].



Rysunek 23: Rysunek przedstawia kilka próbek o różnej rozdzielczości, które zostały wygenerowane z modelu wytrenowanego na zbiorze danych LSUN [112]. Każda próbka składa się z ośmiu obrazów o rozdzielczości 128×192 , 128×224 w górnym rzędzie, 160×128 , 160×160 , 192×128 w środkowym rzędzie i 128×128 , 128×160 , 96×128 w najniższym rzędzie. Rysunek pochodzi z pracy [A7].

informacje o położeniu piksela. Rozwiązanie to jest inspirowane CoordConv GAN [64]. Pierwszy kanał odnosi się do pozycji piksela na osi X, a drugi na osi Y (zob. rysunek 21). Kanały pozycyjne są stosowane tylko na warstwach wejściowych generatora i dyskryminatora. Zakłada się, że w głębszych warstwach sieci te dodatkowe informacje nie są już potrzebne, ponieważ są one zawarte w danych wyjściowych poprzednich warstw i są przekazywane dalej.

Kolejnym kluczowym pomysłem w modelu LocoGAN jest wykorzystanie w pełni konwolucyjnej sieci generatora. Taka architektura gwarantuje niezmienniczość translacji, która jest istotna przy przetwarzaniu obrazów o dowolnej rozdzielczości, przy zachowaniu stałej liczby kanałów. Aby wytrenować model LocoGAN, wprowadzono koncepcję uczenia lokalnego fragmentu obrazka. Dyskryminator operuje tylko na wycinkach obrazu o ustalonym rozmiarze 64×64 (zob. rysunek 22). Zrezygnowano z wypełniania zerami w warstwach konwolucyjnych generatora, żeby umożliwić płynne łączenie fragmentów obrazu generowanych przez LocoGAN. Zerowe wypełnienie zwykle stosuje się w warstwach konwolucyjnych, aby zachować informacje o krawędziach obrazu. Zamiast tego, generowany jest nieco większy szum, który przypomina strukturę obrazu. Ta nadmiarowa część szumu obrazowego skutecznie pełni rolę podkładu dla kolejnych warstw, dzięki czemu unika się dodawania niepotrzebnych zerowych wypełnień w warstwach konwolucyjnych. Prowadzi to do generowania płynnego obrazu o docelowym rozmiarze. To innowacyjne podejście w modelu LocoGAN umożliwia generowanie płynnych i spójnych obrazów o dowolnej rozdzielczości, co można zobaczyć na rysunku 23.

Generator LocoGAN korzysta z innowacyjnej struktury ukrytej, w której szum globalny zawiera informacje o cechach globalnych, a dodatkowe kanały współrzędnych informują o generowanym obszarze obrazu. Dzięki temu możemy łatwo przenosić cechy, takie jak okulary czy uśmiech, między różnymi obrazkami. Korzystając z pełnej architektury konwolucyjnej generatora, przeszczepiono obrazopodobne obszary szumu, które są powiązane z istotnymi cechami, na odpowiednie miejsca w wygenerowanym obrazie. Ta zdolność modelu wynika bezpośrednio z wprowadzonej struktury ukrytej i nie wymaga dodatkowych technik dekodowania. Przykłady



Rysunek 24: Zestaw danych z eksperymentu przenoszenia atrybutów z jednego obrazu na drugi. W kolejnych rzędach dokonano zmian poprzez przetransferowanie wybranych elementów z obrazów znajdujących się w pierwszej kolumnie. Zmiany dotyczą tylko sąsiedztwa przeszczepionych elementów (uśmiechu, okularów) i nie wpływają na resztę obrazu. Rysunek pochodzi z pracy [A7].

tak wygenerowanych obrazów można zobaczyć na rysunku 24.

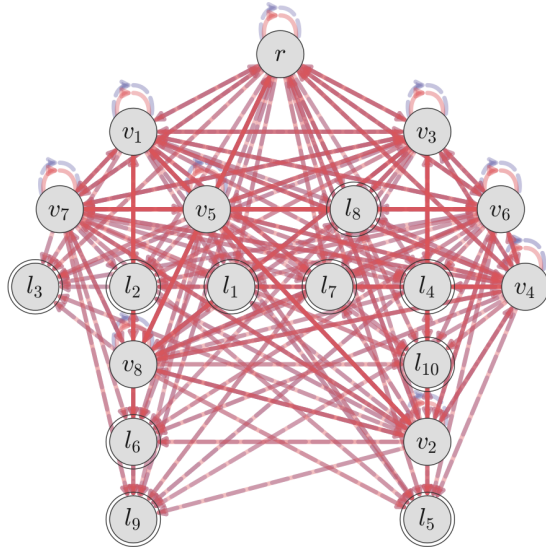
4.2.2 Interpretowalność i wyjaśnialność decyzji podejmowanych przez głębokie modele [A8, A9]

Sieci neuronowe, ze względu na możliwość przetwarzania ogromnej ilości danych, odgrywają obecnie niezwykle istotną rolę w eksploracji danych. Ponadto mają one unikalną zdolność do automatycznego wyodrębniania istotnych cech i reprezentacji danych, co pozwala na efektywne modelowanie skomplikowanych zależności między danymi wejściowymi, a wynikami sieci.

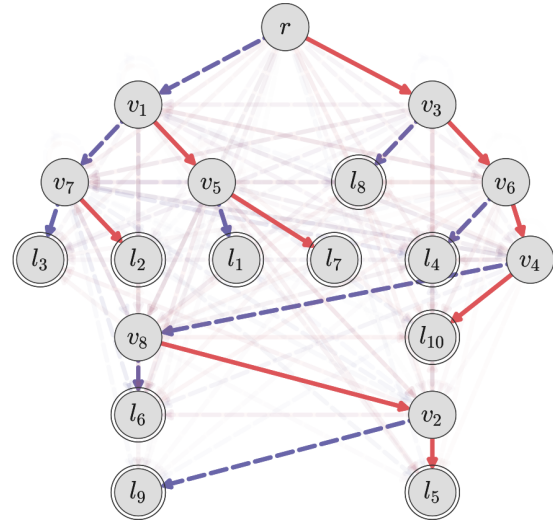
W ostatnich latach zaobserwowano dynamiczny rozwój technik polegających na łączeniu pozytywnych aspektów metod płytkich i głębokich, w celu zoptymalizowania procesu uczenia głębokiego oraz ułatwienia interpretacji jego wyników. Przykładem jednego z obiecujących rozwiązań są tutaj drzewa decyzyjne [34, 74, 96, 114], wykorzystywane do wyjaśniania decyzji sieci neuronowych. Jednak mają one pewne ograniczenia, takie jak rosnąca wraz z głębokością drzewa złożoność, powtarzające się węzły oraz suboptymalna struktura, co wpływa na ich skuteczność w kontekście uczenia głębokiego.

W pracy [A8] zaproponowano innowacyjny model SONG (ang. Self-Organizing Neural Graphs), który wykorzystuje nowy paradygmat trenowania oparty na procesach Markowa [76]. Metoda ta pozwala na jednoczesne uczenie optymalnej struktury grafu i prawdopodobieństw przejść. Dzięki wykorzystaniu techniki propagacji wstecznej, struktura grafu jest stopniowo korygowana podczas procesu uczenia. W rezultacie model zbiega do postaci binarnego skierowanego grafu acyklicznego (zob. twierdzeniu 4 oraz rysunek 25). Podejście to pozwala pokonać ograniczenia związane z drzewami decyzyjnymi, a jednocześnie oferuje efektywną technikę trenowania grafów decyzyjnych opartych na gradientach.

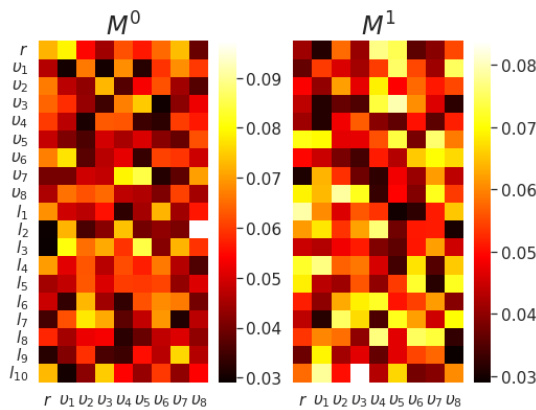
SONG jest zdefiniowany jako graf $G = (V, E^0, E^1)$, gdzie V reprezentuje zbiór węzłów, zaś E^0 i E^1 odpowiadają dwóm alternatywnym zestawom krawędzi. Krawędzie te są połączone



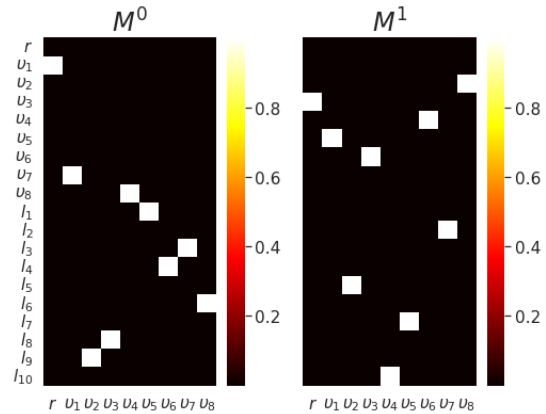
(a) Połączenia węzłów sieci SONG przed trenin-
giem.



(b) Połączenia węzłów SONG po treningu.



(c) M^0 , M^1 z wartościami początkowymi (przed
treningiem).



(d) M^0 , M^1 po treningu.

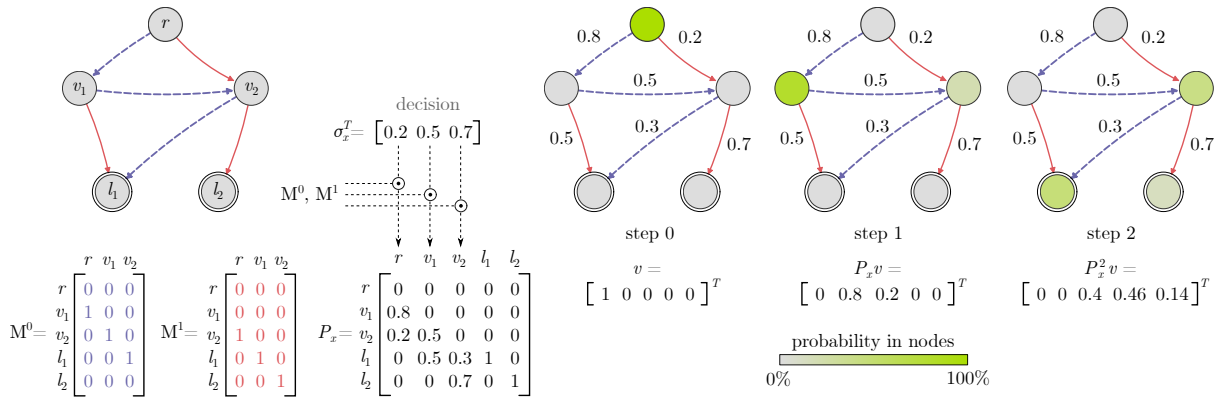
Rysunek 25: Wizualizacja grafów z krawędziami, które odpowiadają macierzom przejść M^0 (oznaczone na niebiesko) oraz M^1 (oznaczone na czerwono), przed i po treningu modelu SONG na zbiorze danych MNIST. Ukazuje ona efektywność binaryzacji połączeń podczas procesu treningu gradientowego. Rysunek pochodzi z pracy [A8].

z macierzami M^1 i M^2 , które zawierają wagi przejścia między węzłami. Stosując funkcję sigmoidalną σ określamy, z którego zbioru krawędzi należy skorzystać podczas przechodzenia z jednego węzła do drugiego. Ostatecznie, wagi poszczególnych krawędzi definiowano jako kombinację tych dwóch macierzy M^0 , M^1 , tj.

$$M_x = \mathbb{1}\sigma(x)^T \odot M^1 + \mathbb{1}(1 - \sigma(x))^T \odot M^0,$$

gdzie symbol \odot oznacza iloczyn Hadamarda, zaś wektor $\mathbb{1}$ składa się z samych jedynek.

Macierz M_x zawiera wagi przejścia od węzłów wewnętrznych do wszystkich węzłów grafu G. Jednakże, aby zastosować teorię procesów Markowa, konieczne jest rozszerzenie macierzy



Rysunek 26: Proces budowy macierzy przejść wraz z procesem uczenia modelu SONG. Po lewej stronie przedstawione są macierze M^0 i M^1 , przykładowy wektor decyzyjny $\sigma(x)$ oraz macierz P_x . Po prawej stronie przedstawione są trzy kolejne przejścia w grafie, zaczynające się od korzenia. Na początku, całkowite prawdopodobieństwo znajduje się w korzeniu. Jednak w kolejnych krokach rozkład prawdopodobieństwa jest rozdzielany między węzłami zgodnie z prawdopodobieństwami przejść, osiągając liście w trzecim kroku. Prawdopodobieństwa w liściach po wszystkich krokach reprezentują prawdopodobieństwa klas (liczba kroków jest uważana za hiperparametr metody). Rysunek pochodzi z pracy [A8].

Metoda	Letter [63]	Connect4 [63]	MNIST [60]
SDT bez destylacji	78.00 (511)	78.63 (255)	94.45 (255)
SDT	81.00 (511)	80.60 (255)	96.76 (255)
SONG-S-large (nasza)	86.25 (511)	82.82 (255)	95.74 (255)
SONG-S-small (nasza)	82.95 (64)	80.27 (8)	94.66 (64)

Tabela 7: Porównanie modeli SDT [34] i SONG na trzech zestawach danych. Dla każdego modelu (w nawiasach) podano dokładność oraz liczbę węzłów wewnętrznych. Model SONG-S-small zawiera minimalną liczbę węzłów niezbędnych do osiągnięcia dokładności porównywalnej z SDT. Natomiast model SONG-S-large używa takiej samej liczby węzłów wewnętrznych jak SDT. Warto zauważyć, że modele SONG są trenowane bez mechanizmu destylacji i zawsze osiągają lepsze wyniki niż SDT bez destylacji. Tabela pochodzi z pracy [A8].

M_x o kolumny odpowiadające liściom

$$P_x = \left[\begin{array}{c|c} M_x & 0 \\ \hline & I \end{array} \right],$$

gdzie 0 jest macierzą zerową, a I jest macierzą jednostkową. W wyniku tego procesu otrzymujemy kwadratową macierz stochastyczną, która opisuje przejścia w łańcuchu Markowa. Macierz P_x zawiera prawdopodobieństwa przejść między krawędziami w jednym kroku czasowym. Obliczając n -tą potęgę macierzy P_x otrzymujemy prawdopodobieństwo przejścia po n krokach. Aby uzyskać prawdopodobieństwo przejścia od korzenia do dowolnego węzła w grafie, w tym do liści reprezentujących wynik modelu, mnożymy otrzymaną macierz przez wektor

$v = (1, 0, \dots, 0)^T$. Prosty przykład, który ilustruje ten proces, został przedstawiony na rysunku 26.

Zakłada się, że w modelu SONG, wychodząc z korzenia, można dotrzeć do któregoś z liści po przynajmniej n krokach (przejsiach między węzłami). Udowodniono, że jeśli model SONG jest uczony na podstawie entropii krzyżowej (ang. cross-entropy, CE), to żadna ścieżka o długości n w grafie G nie odwiedza dwukrotnie tego samego węzła wewnętrznego.

Twierdzenie 4. *Niech będzie dany klasyfikator SONG z n przejściami oraz punkt x ze zbioru danych, gdzie x należy do klasy y . Załóżmy, że wartość entropii krzyżowej dla dowolnej ścieżki grafu G o długości n , która kończy się w liściu przypisanym do klasy y , wynosi zero.*

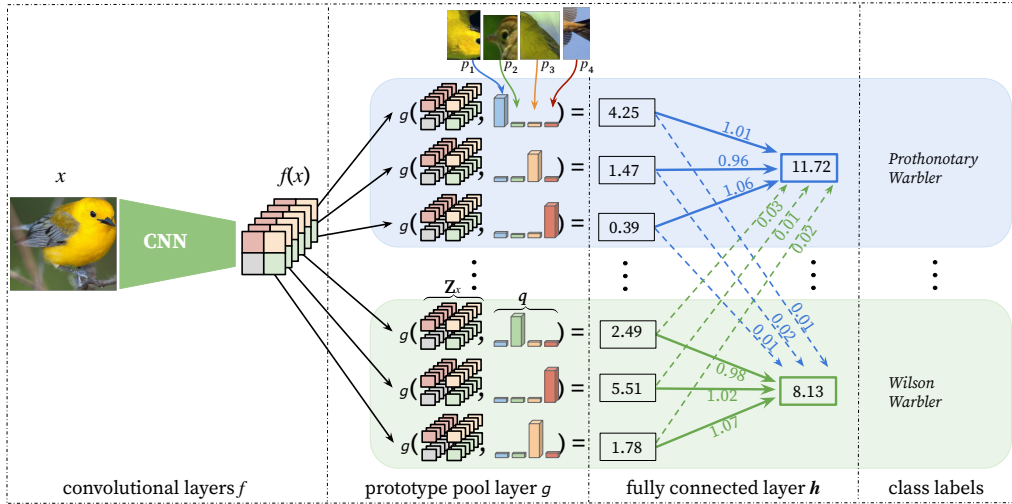
Wtedy żadna ścieżka o długości n w grafie G nie odwiedza dwukrotnie tego samego węzła wewnętrznego z niezerowym prawdopodobieństwem.

Powyższe twierdzenie w pełni dowodzi, że idealnie nauczony model SONG, czyli model, dla którego funkcja kosztu osiąga globalne minimum, nie będzie posiadał żadnych cykli zawierających wewnętrzne węzły. Jest to kluczowy wniosek, który potwierdza stabilność i spójność struktury modelu. Rysunek 25 przedstawia zarówno początkowy stan połączeń modelu SONG, jak i jego stan po zakończeniu procesu uczenia. Warto zauważyć, że macierze przejść M^0 i M^1 po zakończeniu uczenia modelu są zbinaryzowane, co oznacza, że połączenia między węzłami są ustalone i jednoznaczne. To jest istotna obserwacja, ponieważ zapewnia klarowność i interpretowalność działania modelu.

Przeprowadzone eksperymenty jednoznacznie wskazują, że model SONG osiąga lepsze wyniki niż SDT [34] na analizowanych zbiorach danych, co można zaobserwować w tabeli 7. Ponadto, SONG wymaga mniejszej liczby węzłów w porównaniu do SDT, przy jednoczesnym osiągnięciu porównywalnych wyników. Obserwacja ta zgadza się z wynikami przedstawionymi w pracy [94], które dowodzą, że grafy decyzyjne znacznie redukują zapotrzebowanie na pamięć i jednocześnie poprawiają zdolność generalizacji sieci.

Praca [A9] Szerokie zastosowanie uczenia głębokiego w takich dziedzinach, jak diagnostyka medyczna [2] i jazda autonomiczna [106] wymaga, aby modele były w stanie uzasadniać swoje decyzje. W ostatnim czasie wprowadzono wyjaśnialną sieć części prototypowych (ang. Prototypical Part Network, ProtoPNet) [21], która skupia się na kluczowych częściach obrazu i porównuje je z referencyjnymi wzorcami (prototypami) przypisanymi do poszczególnych klas. Proces porównywania opiera się na metryce podobieństwa między aktywacją obrazu, a reprezentacjami części prototypowych. Dzięki temu ProtoPNet jest w stanie wyjaśnić każdą prognozę poprzez przedstawienie listy wzorców referencyjnych i ich podobieństwa do obrazu wejściowego.

ProtoPNet, mimo swoich zalet, wprowadza pewne ograniczenia. Jednym z nich jest założenie, że każda klasa posiada swój unikalny zestaw prototypów. Jednakże, w praktyce wiele cech wizualnych może występować w różnych klasach, co powoduje pewne trudności. Takie podejście ogranicza skalowalność ProtoPNet, ponieważ liczba prototypów rośnie w sposób liniowy wraz z liczbą klas. Ponadto, duża liczba prototypów może utrudniać interpretację wyni-



Rysunek 27: Architektura modelu ProtoPool z prototypową warstwą g , która zawiera pulę prototypów $p_1 - p_4$ i trzy sloty na klasę. Każdy slot jest zaimplementowany jako rozkład $q \in \mathbb{R}^4$ prototypów z puli, gdzie kolejne wartości q odpowiadają prawdopodobieństwu przypisania kolejnych prototypów do slotu. W tym przykładzie p_1 i p_2 są przypisane odpowiednio do pierwszego slotu *Prothonotary Warbler* i *Wilson Warbler*. Wspólne prototypy p_3 i p_4 są przypisane do drugiego i trzeciego slotu obu klas. Rysunek pochodzi z pracy [A9].

ków przez użytkowników i prowadzić do wyboru wielu prototypów tła, co wpływa negatywnie na klarowność i użyteczność modelu [E14].

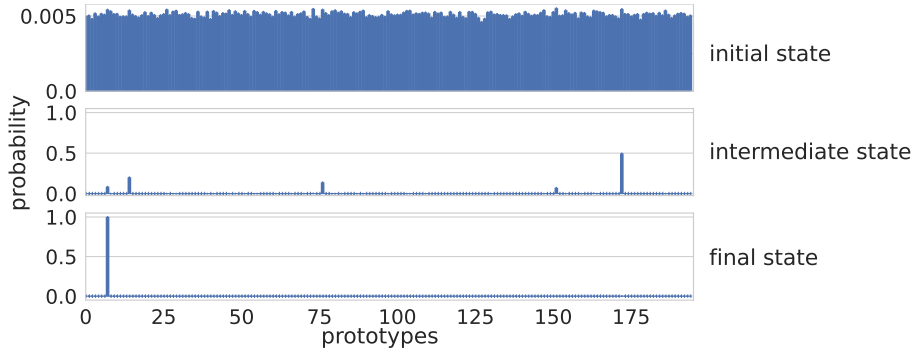
Aby pokonać powyższe ograniczenia, w pracy [A9] wprowadzono nowy model ProtoPool, który wykorzystuje innowacyjne mechanizmy, znacznie zmniejszające liczbę prototypów i ułatwiające proces uczenia. W przeciwieństwie do wcześniejszych podejść, takich jak sztywne przypisanie prototypów do klas [E14, 21, 101] lub węzłów drzewa [73], model ProtoPool wprowadza miękkie przypisanie reprezentowane przez rozkład prawdopodobieństwa dla zbioru prototypów (zob. rysunek 27). Dzięki temu model ten optymalnie wykorzystuje prototypy z dostępnej puli.

Do zdefiniowania rozkładów dla prototypów można zastosować funkcję *softmax*. Jednakże zwykłe użycie funkcji softmax prowadzi do przypisania wielu prototypów z niskim prawdopodobieństwem, co może obniżyć interpretowalność modelu. Dlatego w celu uzyskania rozkładów, w których tylko jeden prototyp w pojedynczym slotcie posiada prawdopodobieństwo bliskie 1, potrzebna jest różniczkowalna funkcja *argmax*. Idealnym rozwiązaniem w tym przypadku jest estymator Gumbel-Softmax [50]. Dla wektora $q = (q^1, \dots, q^n) \in \mathbb{R}^n$ i parametru $\tau \in (0, \infty)$, *Gumbel-Softmax* jest zdefiniowany jako

$$\text{Gumbel-softmax}(q, \tau) = (y^1, \dots, y^n) \in \mathbb{R}^n,$$

gdzie $y^i = \frac{\exp((q^i + \eta_i)/\tau)}{\sum_{m=1}^n \exp((q^m + \eta_m)/\tau)}$, zaś η_m są próbkami pobranymi ze standardowego rozkładu Gumbela dla $m \in \{1, \dots, n\}$.

Dzięki zastosowaniu estymatora Gumbel-Softmax, możliwe jest generowanie dyskretnych próbek w sposób gładki i różniczkowalny (zmniejszając parametr τ), co jest istotne w procesie uczenia. Przykładowe działanie funkcji Gumbel-Softmax można zaobserwować na rysunku 28.



Rysunek 28: Rozkład próbek na początkowym, środkowym i końcowym etapie uczenia modelu ProtoPool. Początkowo wszystkim prototypom przypisywane jest prawdopodobieństwo bliskie 0,005. Następnie dokonuje się binaryzacji rozkładu, a ostatecznie jeden prototyp jest wybrany i przypisywany z prawdopodobieństwem bliskim 1. Rysunek pochodzi z pracy [A9].

	CUB-200-2011	Stanford Cars
Model	Acc [%]	Acc [%]
ProtoPool	85.5	88.9
bez \mathcal{L}_{orth}	82.4	86.8
bez Gumbel-Softmax	80.3	64.5
bez Gumbel-Softmax i \mathcal{L}_{orth}	65.1	30.8
bez podobieństwa ogniskowego	85.3	88.8

Tabela 8: Wpływ mechanizmów wprowadzonych w modelu ProtoPool na jego wydajność, przedstawiony dla zestawów danych CUB-200-2011 [99] i Stanford Cars [55]. Rozważamy trening modelu bez \mathcal{L}_{orth} , z softmax zamiast Gumbel-Softmax i z podobieństwem z ProtoPNet zamiast podobieństwa ogniskowego. Można zaobserwować, że połączenie proponowanych mechanizmów uzyskuje najlepszą dokładność. Tabela pochodzi z pracy [A9].

W ramach pracy [A9] wprowadzono również innowacyjny mechanizm, mający na celu skuteczne rozwiązanie problemu, w którym ten sam prototyp mógł być przypisany do wielu slotów jednej klasy. Taka sytuacja negatywnie wpływała na pojemność warstwy puli prototypów, co prowadziło do osiągania słabych wyników. Aby przeciwdziałać temu problemowi, zastosowano funkcję kosztu

$$\mathcal{L}_{orth} = \sum_{i < j}^K \frac{\langle q_i, q_j \rangle}{\|q_i\|_2 \cdot \|q_j\|_2},$$

gdzie q_1, \dots, q_K reprezentowały rozkłady dla danej klasy. Dzięki temu, model efektywnie zarządzał przypisaniem prototypów do odpowiednich slotów, co w rezultacie przyczyniło się do znaczącej poprawy wydajności i skuteczności modelu.

Kolejnym istotnym wyzwaniem, które pojawiło się w kontekście wyboru prototypów, było występowanie wysokiej aktywacji, gdy wszystkie elementy w zbiorze $Z_x = \{z \in f(x)\}$ przypominały prototypy związane z tłem. Aby skutecznie temu zapobiec, wprowadzono innowacyjną

funkcję podobieństwa ogniskowego

$$g = \max_{z \in Z_x} g(z) - \text{mean}_{z \in Z_x} g(z),$$

której celem było poszerzenie różnicy między maksymalną, a średnią aktywacją, eliminując tym samym skupienie prototypów na tle. Dodatkowo, wprowadzenie tego mechanizmu poprawiło proces uczenia modelu, ponieważ wcześniej gradient był ograniczony tylko do najbardziej aktywnej części obrazu. W tabeli 8 przedstawiono wpływ poszczególnych mechanizmów na wydajność modelu ProtoPool.

4.3 Podsumowanie i dalsze kierunki badań

W ramach przeprowadzonych badań, dokonano istotnego postępu w rozwijaniu nowych metod eksploracji i interpretacji danych przy pomocy płytkich i głębokich modeli uczenia maszynowego. Badania skoncentrowały się na eksploracji przestrzeni ukrytej oraz interpretacji wyników głębokich modeli klasyfikacyjnych. Przyczyniły się one nie tylko do lepszego zrozumienia istoty samej natury danych, ale także w wielu przypadkach doprowadziły do znaczącej poprawy wydajności głębokich sieci neuronowych, jednocześnie optymalizując ich struktury i doskonaląc ich proces uczenia.

W przeprowadzonych badaniach uwzględniono aktualne problemy związane z przetwarzaniem dużych ilości danych, a także korzystano z różnorodnych narzędzi z dziedziny uczenia maszynowego. Dużo uwagi poświęcono teoretycznemu uzasadnieniu opracowanych metod oraz praktycznej analizie proponowanych rozwiązań. Rezultaty tych badań zostały zaprezentowane na renomowanych konferencjach naukowych oraz opublikowane w prestiżowych czasopismach, co potwierdza ich wartość i znaczenie dla społeczności naukowej.

Powstałe prace otwierają możliwość przeprowadzania zaawansowanej analizy wysokowymiarowych danych oraz skutecznego przetwarzania dużej ilości informacji. Pierwsza grupa prac koncentruje się na badaniu płytkich metod, które reprezentują dane za pomocą skupisk podprzestrzeni zawartych w przestrzeni danych lub zajmują się analizą niekompletnych danych. To podejście umożliwia skuteczną ekstrakcję istotnych informacji z danych o dużych rozmiarach, pokonując trudności związane z ich niekompletnością. Druga grupa badań skupia się na eksploracji i analizie głębokich sieci neuronowych, które stanowią aktualnie dynamicznie rozwijającą się dziedzinę. Poprzez dogłębne badanie struktury tych sieci oraz analizę osiąganych przez nie wyników, opracowano skuteczne metody optymalizacji procesu uczenia, konstrukcji sieci i zwiększania ich wydajności.

Wyniki przeprowadzonych badań przyczyniły się do doskonalenia technik trenowania głębokich sieci. Dodatkowo, badania skupiły się na poprawie zdolności sieci do generalizacji poprzez analizę przestrzeni ukrytej. Dzięki temu możliwe stało się lepsze zrozumienie danych generowanych przez głębokie modele oraz wyjaśnienie ich predykcji. Ponadto badania te miały na celu doskonalenie procesu predykcyjnego, identyfikując kluczowe cechy i ich zależności, które wpływały na decyzje podejmowane przez modele.

W przyszłości wnioskodawca planuje kontynuować niektóre z przedstawionych badań (np. dalszą eksplorację modeli generatywnych). Badania te mogą rozwijać się równolegle w dwóch

istotnych kierunkach. Po pierwsze, interesujące jest opracowanie nowych technik transferu uczenia, które wykorzystają ukrytą przestrzeń wstępnie wytrenowanych modeli generatywnych. Celem tego jest poprawa wydajności w dalszych zadaniach, takich jak klasyfikacja, co może prowadzić do szybszej konwergencji i lepszych wyników. Po drugie, nadal istnieje potrzeba opracowania odpowiednich miar generatywności do uczenia i oceny modeli, ponieważ obecnie dostępne rozwiązania nie są optymalne [14].

Kolejnym istotnym obszarem rozwoju sztucznej inteligencji jest połączenie głębokich sieci neuronowych z klasycznymi metodami interpretowalnymi, takimi jak drzewa decyzyjne i grafy. Przykładem takiego połączenia może być wykorzystanie badań dotyczących samorganizujących się grafów w kontekście wyjaśnialności predykcji modeli. W pracy [73] zastosowano drzewa decyzyjne jako przykład klasycznej metody interpretowalnej.

4.4 Literatura

- [1] J. Aczél and Z. Daróczy. On measures of information and their characterizations. *New York*, 1975.
- [2] Michael Anis Mihdi Afnan, Yanhe Liu, Vincent Conitzer, Cynthia Rudin, Abhishek Mishra, Julian Savulescu, and Masoud Afnan. Interpretable, not black-box, artificial intelligence should be used for embryo selection. *Human Reproduction Open*, 2021.
- [3] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park. Fast algorithms for projected clustering. In *ACM SIGMOD Record*, volume 28, pages 61–72. ACM, 1999.
- [4] C. C. Aggarwal and P. S. Yu. *Finding generalized projected clusters in high dimensional spaces*, volume 29. ACM, 2000.
- [5] Eirikur Agustsson, Alexander Sage, Radu Timofte, and Luc Van Gool. Optimal transport maps for distribution preserving operations on latent spaces of generative models. *arXiv preprint arXiv:1711.01970*, 2017.
- [6] Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauber. Latent space oddity: on the curvature of deep generative models. *arXiv:1711.11379*, 2017.
- [7] W Bär, M Alber, and F Nüsslin. A variable fluence step clustering and segmentation algorithm for step and shoot imrt. *Physics in Medicine & Biology*, 46(7):1997, 2001.
- [8] Alexander C. Berg, Tamara L. Berg, and Jitendra Malik. Shape matching and object recognition using low distortion correspondences. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 26–33. IEEE, 2005.
- [9] Urs Bergmann, Nikolay Jetchev, and Roland Vollgraf. Learning texture manifolds with the periodic spatial gan. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 469–477. JMLR. org, 2017.

- [10] M. J. Berry and G. Linoff. *Data mining techniques: for marketing, sales, and customer support*. John Wiley & Sons, Inc., 1997.
- [11] David Berthelot, Colin Raffel, Aurko Roy, and Ian J. Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer. *arXiv:1807.07543*, 2018.
- [12] C. Böhm, K. Kailing, P. Kröger, and A. Zimek. Computing clusters of correlation connected objects. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 455–466. ACM, 2004.
- [13] Christian Bohm, K Railing, H-P Kriegel, and Peer Kroger. Density connected clustering with local subspace preferences. In *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*, pages 27–34. IEEE, 2004.
- [14] Ali Borji. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 179:41–65, 2019.
- [15] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv:1511.06349*, 2015.
- [16] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.
- [17] Lora E Burke, Jacqueline M Dunbar-Jacob, and Martha N Hill. Compliance with cardiovascular disease prevention strategies: a review of the research. *Annals of Behavioral Medicine*, 19(3):239–263, 1997.
- [18] Stef Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of statistical software*, 45(3), 2011.
- [19] Gal Chechik, Jeremy Heitz, Gal Elidan, Pieter Abbeel, and Daphne Koller. Max-margin classification of incomplete data. In *Advances in Neural Information Processing Systems*, pages 233–240, 2007.
- [20] Gal Chechik, Jeremy Heitz, Gal Elidan, Pieter Abbeel, and Daphne Koller. Max-margin classification of data with absent features. *Journal of Machine Learning Research*, 9:1–21, 2008.
- [21] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. In *NeurIPS*, pages 8930–8941, 2019.
- [22] J. Chen and J. Yang. Robust subspace segmentation via low-rank representation. *IEEE transactions on cybernetics*, 44(8):1432–1445, 2014.
- [23] Claudio Conversano and Roberta Siciliano. Incremental tree-based missing data imputation with lexicographic ordering. *Journal of Classification*, 26(3):361–379, 2009.

- [24] Inderjit S Dhillon, Subramanyam Mallela, and Dharmendra S Modha. Information-theoretic co-clustering. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 89–98, 2003.
- [25] Adji Bousso Dieng, Dustin Tran, Rajesh Ranganath, John Paisley, and David Blei. Variational inference via χ upper bound minimization. *NIPS*, 2017.
- [26] Manfredo Perdigao do Carmo. *Riemannian geometry*. Birkäuser, 1992.
- [27] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv:1605.09782*, 2016.
- [28] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially Learned Inference. *arXiv:1606.00704*, 2016.
- [29] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2017.
- [30] Antonio D’Ambrosio, Massimo Aria, and Roberta Siciliano. Accurate tree-based missing data imputation and data fusion within the statistical learning paradigm. *Journal of classification*, pages 1–32, 2012.
- [31] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2765–2781, 2013.
- [32] Ad Feelders. Data mining in economic science. *Dealing with the data flood*, pages 166–175, 2002.
- [33] Roman Filipovych, Susan M Resnick, and Christos Davatzikos. Semi-supervised cluster analysis of imaging data. *NeuroImage*, 54(3):2185–2197, 2011.
- [34] Nicholas Frosst and Geoffrey Hinton. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*, 2017.
- [35] Xiang Gao, Meera Sitharam, and Adrian E Roitberg. Bounds on the Jensen gap, and implications for mean-concentrated distributions. *arXiv preprint arXiv:1712.05267*, 2017.
- [36] Zoubin Ghahramani, Geoffrey E Hinton, et al. The em algorithm for mixtures of factor analyzers. Technical report, Technical Report CRG-TR-96-1, University of Toronto, 1996.
- [37] Zoubin Ghahramani and Michael I Jordan. Supervised learning from incomplete data via an em approach. In *Advances in neural information processing systems*, pages 120–127, 1994.
- [38] Milena A Gianfrancesco, Suzanne Tamang, Jinoos Yazdany, and Gabriela Schmajuk. Potential biases in machine learning algorithms using electronic health record data. *JAMA internal medicine*, 178(11):1544–1547, 2018.

- [39] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [40] Gérard Govaert and Mohamed Nadif. *Co-clustering: models, algorithms and applications*. John Wiley & Sons, 2013.
- [41] P. D. Grünwald. *The minimum description length principle*. MIT press, 2007.
- [42] Jenny A Harding, Muhammad Shahbaz, and A Kusiak. Data mining in manufacturing: a review. 2006.
- [43] J. A. Hartigan. *Clustering algorithms*. John Wiley & Sons, Inc., 1975.
- [44] Elad Hazan, Roi Livni, and Yishay Mansour. Classification with low rank and missing data. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 257–266, 2015.
- [45] R. Heckel and H. Bölcskei. Robust subspace clustering via thresholding. *IEEE Transactions on Information Theory*, 61(11):6320–6342, 2015.
- [46] Ari Heljakka, Arno Solin, and Juho Kannala. Pioneer Networks: Progressively Growing Generative Autoencoder. *arXiv:1807.03026*, 2018.
- [47] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. *arXiv:1706.08500*, 2017.
- [48] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.
- [49] R. A. Irizarry, B. Hobbs, and other. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–264, 2003.
- [50] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv:1611.01144*, 2016.
- [51] Neesha Jothi, Wahidah Husain, et al. Data mining in healthcare—a review. *Procedia computer science*, 72:306–313, 2015.
- [52] Agnan Kessy, Alex Lewin, and Korbinian Strimmer. Optimal whitening and decorrelation. *The American Statistician*, 72(4):309–314, 2018.
- [53] Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018.
- [54] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

- [55] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.
- [56] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.
- [57] Samuli Laine. Feature-based metrics for exploring the latent space of generative models. In *International Conference on Learning Representations*, 2018.
- [58] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International Conference on Machine Learning*, pages 1558–1566, 2016.
- [59] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [60] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database, 2010.
- [61] M. Li and P. Vitányi. *An introduction to Kolmogorov complexity and its applications*. Springer Science & Business Media, 2013.
- [62] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- [63] M. Lichman. UCI machine learning repository, 2013.
- [64] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *Advances in Neural Information Processing Systems*, pages 9605–9616, 2018.
- [65] Zhun-ga Liu, Quan Pan, Jean Dezert, and Arnaud Martin. Adaptive imputation of missing values for incomplete pattern classification. *Pattern Recognition*, 52:85–95, 2016.
- [66] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision, ICCV*, December 2015.
- [67] Can-Yi Lu, Hai Min, Zhong-Qiu Zhao, Lin Zhu, De-Shuang Huang, and Shuicheng Yan. Robust and efficient subspace segmentation via least squares regression. In *European conference on computer vision*, pages 347–360. Springer, 2012.
- [68] Vaden Masrani, Tuan Anh Le, and Frank Wood. The thermodynamic variational objective. *NIPS*, 2019.

- [69] Patrick E McKnight, Katherine M McKnight, Souraya Sidani, and Aurelio Jose Figueredo. *Missing data: A gentle introduction*. Guilford Press, 2007.
- [70] Abhilash Alexander Miranda, Yann-Aël Le Borgne, and Gianluca Bontempi. New routes from minimal approximation error to principal components. *Neural Processing Letters*, 27(3):197–207, 2008.
- [71] Tom M Mitchell. Machine learning and data mining. *Communications of the ACM*, 42(11):30–36, 1999.
- [72] Gabriela Moise and Jörg Sander. Finding non-redundant, statistically significant regions in high dimensional data: a novel approach to projected and subspace clustering. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 533–541. ACM, 2008.
- [73] Meike Nauta et al. Neural prototype trees for interpretable fine-grained image recognition. In *CVPR*, pages 14933–14943, 2021.
- [74] Meike Nauta, Ron van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. *arXiv preprint arXiv:2012.02046*, 2020.
- [75] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *Workshop on Deep Learning and Unsupervised Feature Learning*, NeurIPS, 2011.
- [76] James R Norris. *Markov chains*. Number 2. Cambridge university press, 1998.
- [77] Costas Panagiotakis, Ilias Grinias, and Georgios Tziritas. Natural image segmentation based on tree equipartition, bayesian flooding and region merging. *IEEE Transactions on Image Processing*, 20(8):2276–2287, 2011.
- [78] Costas Panagiotakis, Harris Papadakis, Elias Grinias, Nikos Komodakis, Paraskevi Frago-poulou, and Georgios Tziritas. Interactive image segmentation based on synthetic graph coordinates. *Pattern Recognition*, 46(11):2940–2952, 2013.
- [79] D. Park, C. Caramanis, and S. Sanghavi. Greedy subspace clustering. In *Advances in Neural Information Processing Systems*, pages 2753–2761, 2014.
- [80] Girish Punj and David W Stewart. Cluster analysis in marketing research: Review and suggestions for application. *Journal of marketing research*, pages 134–148, 1983.
- [81] Sheng Qian, Guanyue Li, Wen-Ming Cao, Cheng Liu, Si Wu, and Hau San Wong. Improving representation learning in autoencoders via multidimensional interpolation and dual regularizations. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3268–3274. AAAI Press, 2019.

- [82] L. Qiao, B. Zhang, Y. Sun, and J. Su. Rigidly self-expressive sparse subspace clustering. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 101–114. Springer, 2016.
- [83] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
- [84] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv:1511.06434*, 2015.
- [85] Emmanuel Ramasso, Vincent Placet, and Mohamed Lamine Boubakar. Unsupervised consensus clustering of acoustic emission time-series for robust damage sequence estimation in composites. *IEEE Transactions on Instrumentation and Measurement*, 64(12):3297–3307, 2015.
- [86] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *ICML*, 2014.
- [87] Eitan Richardson and Yair Weiss. On gans and gmms. In *Advances in Neural Information Processing Systems*, pages 5847–5858, 2018.
- [88] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *arXiv:1606.03498*, 2016.
- [89] Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.
- [90] C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.
- [91] Michael J Shaw, Chandrasekar Subramaniam, Gek Woo Tan, and Michael E Welge. Knowledge management and data mining for marketing. *Decision support systems*, 31(1):127–137, 2001.
- [92] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. How good is my gan? In *Proceedings of the European conference on computer vision (ECCV)*, pages 213–229, 2018.
- [93] Ken Shoemake. Animating rotation with quaternion curves. In *ACM SIGGRAPH computer graphics*, volume 19, pages 245–254. ACM, 1985.
- [94] Jamie Shotton, Toby Sharp, Pushmeet Kohli, Sebastian Nowozin, John Winn, and Antonio Criminisi. Decision jungles: Compact and rich models for classification. 2016.
- [95] Florence L Stahura and Jurgen Bajorath. Virtual screening methods that complement HTS. *Combinatorial Chemistry & High Throughput Screening*, 7(4):259–269, 2004.

- [96] Alberto Suárez and James F Lutsko. Globally optimal fuzzy decision trees for classification and regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(12):1297–1311, 1999.
- [97] A. L. Tarca, V. J. Carey, X. Chen, R. Romero, and S. Drăghici. Machine learning and its applications to biology. *PLoS computational biology*, 3(6):e116, 2007.
- [98] Chad Vidden, Marco Vriens, and Song Chen. Comparing clustering methods for market segmentation: A simulation study. *Applied Marketing Analytics*, 2(3):225–238, 2016.
- [99] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.
- [100] Bao Wang, Xiyang Luo, Zhen Li, Wei Zhu, Zuoqiang Shi, and Stanley Osher. Deep neural nets with interpolating function as output activation. In *Advances in Neural Information Processing Systems*, pages 743–753, 2018.
- [101] Jiaqi Wang et al. Interpretable image recognition by constructing transparent embedding space. In *ICCV*, pages 895–904, 2021.
- [102] Y.-X. Wang, H. Xu, and C. Leng. Provable subspace clustering: When lrr meets ssc. In *Advances in Neural Information Processing Systems*, pages 64–72, 2013.
- [103] D. Warszycki, S. Mordalski, K. Kristiansen, R. Kafel, I. Sylte, Z. Chilmonczyk, and A. J. Bojarski. A linear combination of pharmacophore hypotheses as a new tool in search of new active compounds—an application for 5-HT_{1A} receptor ligands. *PloS ONE*, 8(12):e84510, 2013.
- [104] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.
- [105] Tom White. Sampling generative networks: Notes on a few effective techniques. *arXiv:1609.04468*, 2016.
- [106] Gesa Wiegand, Matthias Schmidmaier, Thomas Weber, Yuanting Liu, and Heinrich Hussmann. I drive-you trust: Explaining driving behavior of autonomous cars. In *Extended abstracts of the 2019 chi conference on human factors in computing systems*, pages 1–6, 2019.
- [107] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.
- [108] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in neural information processing systems*, pages 82–90, 2016.

- [109] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.
- [110] C. You, D. Robinson, and R. Vidal. Scalable sparse subspace clustering by orthogonal matching pursuit. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3918–3927, 2016.
- [111] Chong You, Chun-Guang Li, Daniel P. Robinson, and René Vidal. Oracle based active set algorithm for scalable elastic net subspace clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3928–3937, 2016.
- [112] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.
- [113] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiao lei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5907–5915, 2017.
- [114] Quanshi Zhang, Yu Yang, Haotian Ma, and Ying Nian Wu. Interpreting cnns via decision trees. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6261–6270, 2019.

5 Informacja o wykazywaniu się istotną aktywnością naukową realizowaną w więcej niż jednej uczelni lub instytucji naukowej

Na aktywność naukową wnioskodawcy, realizowaną w więcej niż jednej jednostce badawczej, składają się staże i wizyty naukowe, jak również granty badawcze. Poza taką sformalizowaną współpracą, powstały również publikacje z osobami związanymi z innymi jednostkami naukowymi. Poniżej znajduje się wykaz wszystkich działań dokumentujących aktywność naukową.

5.1 Staże i wizyty naukowe

- 2023.03.01 – 2023.03.16: staż podoktorski na Wydziale Informatyki i Telekomunikacji Politechnika Wroclawska, Polska, współpraca z drem hab. Maciejem Ziębą.

W czasie pobytu współpracowano w zakresie badań nad brakującymi danymi, zjawiskiem zaszumionych etykiet oraz pojęciem meta-uczenia.

- 2023.01.21 – 2023.01.28: wizyta w Department of Media and Digital Technologies, Uniwersytet w Helsinkach, Finlandia, współpraca z prof. Arto Klami.

W trakcie wizyty nawiązano współpracę z prof. Arto Klami oraz członkami z jego grupy badawczej wnioskowania probabilistycznego w Katedrze Informatyki w Uniwersytet w Helsinkach.

- 2022.07.05 – 2022.07.31: staż podoktorski w Instituto Superior Técnico Uniwersytet Autonomiczny w Barcelonie, Hiszpania, współpraca z prof. Joost van de Weijer.

W ramach stażu nawiązano współpracę z zespołem prof. Joost van de Weijera, w zakresie kognitywnych narzędzi wyjaśniających działanie głębokich sieci neuronowych.

5.2 Udział w grantach realizowanych we współpracy krajowej

W ramach aktywności naukowej wnioskodawca uczestniczył (bądź uczestniczy) w realizacji 9 grantów badawczych (bądź badawczo-rozwojowych), z czego w jednym pełni rolę kierownika. Poniżej wymieniono granty badawcze realizowane we współpracy z osobami spoza uczelni macierzystej.

- (1) 2019 – 2023: *Sztuczne sieci neuronowe inspirowane biologicznie,*

Team-Net (FNP), nr POIR.04.04.00-00-14DE/18-00,
funkcja: młody naukowiec,
grant zrzeszający 6 zespołów badawczych,
z czego 3 są kierowane przez osoby spoza uczelni macierzystej:
prof. dra hab. Daniela Wójcika (Instytut Biologii Doświadczalnej PAN),
dra hab. Pawła Oświecimkę (Instytut Fizyki Jądrowej PAN),
dra hab. Tomasza Trzcińskiego (Politechnika Warszawska).

- (2) 2022 – 2023: *System automatycznego wykrywania wrodzonych wad serca dziecka w badaniu ultrasonograficznym (UKG) wykorzystujący modele AI*, realizująca projekt B+R przez UES Sp. z o. o.,
funkcja: wykonawca,
grant realizowany we współpracy z drem inż. Arkadiuszem Lewickim (Wyższa Szkoła Informatyki i Zarządzania w Rzeszowie).
- (3) 2017: *Deep learning in terms of a distributed system of energy-efficient mobile devices for the optimization of the machine operation process*, Europejski Fundusz Rozwoju Regionalnego, Reliability Solutions Sp. z o.o.,
funkcja: wykonawca,
grant realizowany we współpracy z drem Pawłem Morkiszem (Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie).

Lista pozostałych projektów badawczych:

- (1) 2021 - 2024: *Rzadkie i dyskretne reprezentacje w ukrytych przestrzeniach*, Sonata (NCN), nr 2020/39/D/ST6/01332,
funkcja: kierownik.
- (2) 2022 - 2025: *Głębokie samoorganizujące się grafy neuronowe*, Opus (NCN), nr 2021/41/B/ST6/01370,
funkcja: wykonawca.
- (3) 2020 - 2023: *Generowanie rzeczywistych obrazów za pomocą modeli opartych na architekturze autoenkodera*, Opus (NCN), nr 2019/33/B/ST6/00894,
funkcja: wykonawca.
- (4) 2018 - 2021: *Efektywne metody uczenia nienadzorowanego z zastosowaniami w głębokim nauczaniu*, Opus (NCN), nr 2017/25/B/ST6/01271,
funkcja: wykonawca.
- (5) 2017 - 2020: *Dodatkowa informacja w grupowaniu danych i zagadnieniach pokrewnych*, Sonata (NCN), nr 2016/21/D/ST6/00980,
funkcja: wykonawca.
- (6) 2016 - 2019: *Teoria analizy niekompletnych danych*, Opus (NCN), nr 2015/19/B/ST6/01819,
funkcja: wykonawca.

5.3 Współpraca krajowa i międzynarodowa potwierdzona publikacjami

Poza współpracą nawiązaną w ramach wyjazdów i grantów, wnioskodawca prowadził również badania we współpracy z innymi osobami spoza jednostki macierzystej. Poniżej zaprezentowano pełną listę wszystkich współautorów publikacji, pracujących w jednostkach innych niż Uniwersytet Jagielloński:

- prof. Mario A. T. Figueiredo (Uniwersytet w Lizbonie) - 3 publikacje,
- dr Rafał Kafel (Instytut Farmakologii im. Jerzego Maja PAN) – 1 publikacja,
- dr Timoleon Kipouros (Uniwersytet Cambridge) – 1 publikacja,
- prof. Per Ola Kristensson (Uniwersytet Cambridge) – 1 publikacja,
- dr hab. Rafał Kurczab (Instytut Farmakologii im. Jerzego Maja PAN) – 1 publikacja,
- dr Mateusz Marzec (Akademia Górniczo-Hutnicza) – 1 publikacja,
- Agnieszka Słowik (Uniwersytet Cambridge) – 1 publikacja,
- dr Sławomir K. Tadeja (Uniwersytet Cambridge) – 2 publikacje,
- dr hab. Tomasz Trzcíński (Politechnika Warszawska) – 1 publikacja,
- dr Dawid Warszycycki (Instytut Farmakologii im. Jerzego Maja PAN) – 1 publikacja.

6 Informacja o osiągnięciach dydaktycznych, organizacyjnych oraz popularyzujących naukę

6.1 Osiągnięcia dydaktyczne

- (1) Prowadzenie następujących wykładów dla studentów:
 - *Języki programowania do przetwarzania danych* (2022/2023, 2021/2022, 2020/2021, 2019/2020, 2018/2019),
 - *Praktyczne wprowadzenie do programowania* (2022/2023, 2021/2022, 2019/2020, 2018/2019),
 - *Programowanie obiektowe w języku Java* (2017/2018),
 - *Przetwarzanie i wizualizacja danych w SAS* (2022/2023, 2021/2022, 2020/2021, 2019/2020, 2018/2019).
- (2) Prowadzenie następujących seminariów dla studentów:
 - *Inżynieria danych i oprogramowania* (2018/2019).
- (3) Promotor 10 magistrantów (4 osoby z informatyki, 6 z matematyki).
- (4) Prowadzenie ćwiczeń oraz zajęć laboratoryjnych z następujących przedmiotów na kierunku informatyka i matematyka (od października 2013 do teraz):
 - Algebra liniowa z geometrią 1,
 - Analiza danych statystycznych w systemie SAS,
 - Informatyka,
 - Języki programowania do przetwarzania danych,
 - Metody numeryczne,
 - Modelowanie matematyczne 1,
 - Praktyczne wprowadzenie do programowania,
 - Programowanie obiektowe w języku Java,
 - Przetwarzanie i wizualizacja danych w SAS,
 - Rachunek prawdopodobieństwa 1,
 - Rachunek prawdopodobieństwa 2,
 - Równania różniczkowe cząstkowe 1,
 - Równania różniczkowe i różnicowe,
 - Statystyka 1,
 - Statystyka 2,
 - Wprowadzenie do programowania funkcyjnego.

6.2 Osiągnięcia organizacyjne

- (1) Sekretarz komisji rekrutacyjnej oraz zespołu egzaminacyjnego na studia I i II stopnia na Wydziale Matematyki i Informatyki UJ (2017-2023).
- (2) Koordynator Festiwal Nauki i Sztuki na Wydziale Matematyki i Informatyki UJ (2016-2020).
- (3) Koordynator w projekcie „Sztuczne sieci neuronowe inspirowane biologicznie” (2019-2023).

- (4) Członek Komitetu Organizacyjnego konferencji TFML Theoretical Foundations of Machine Learning (2017 i 2019).
- (5) Współorganizator wydarzeń popularyzujących z uczenia maszynowego m.in. Machine Learning Nokia Workshop (2019).
- (6) Członek komitetu organizacyjnego Jubileuszowego Zjazdu Matematyków Polskich w stulecie PTM (2019).
- (7) Jury stypendium imienia Michała Jakuba Łyska (2018 i 2019).
- (8) Uczestnictwo w tworzeniu tzw. „ścieżki SAS” na Wydziale Matematyki i Informatyki UJ, we współpracy z firmą SAS Institute Sp. z o. o. (studenci mogą otrzymać certyfikat „Analityk danych SAS” wydawany wspólnie przez Spółkę i Wydział).
- (9) Aktywne uczestnictwo w pozyskiwaniu najnowszej aparatury badawczej (złożenie wniosku do MNiSW w sprawie przyznania środków finansowych na realizację inwestycji związanej z działalnością naukową).
- (10) Pomoc w modernizacji i rozbudowie wydziałowej serwerowni.

6.3 Osiągnięcia popularyzujące naukę

- (1) Wykłady popularyzujące naukę:
 - „*Google Maps – wyszukiwanie trasy*”, cykl warsztatów „Praca z uczniem uzdolnionym matematycznie”, 24 czerwca 2015.
- (2) Współautor książek popularno-naukowych:
 - „*Głębokie uczenie: wprowadzenie*”, Wydawnictwo: Helion, 184 stron, 2022.