ROZPRAWA DOKTORSKA

mgr inż. Gracjan Kątek

Zastosowanie hybrydowych metod ekstrakcji cech w problemie detekcji dezinformacji

The use of hybrid feature extraction methods in the problem of disinformation detection

DZIEDZINA: NAUKI INŻYNIERYJNO-TECHNICZNE
DYSCYPLINA: INFORMATYKA TECHNICZNA I TELEKOMUNIKACJA

PROMOTOR

dr hab. inż. Rafał Kozik, prof. PBŚ

Politechnika Bydgoska im. Jana i Jędrzeja Śniadeckich

PROMOTOR POMOCNICZY

dr inż. Adam Marchewka, prof. PBŚ

Politechnika Bydgoska im. Jana i Jędrzeja Śniadeckich

Bydgoszcz, 2026

Spis treści

1 Wstęp 7

1.1 Aktualność tematu 7

1.2 Cel, hipoteza i zakres pracy 8

1.3 Struktura pracy 9

1.4 Główne wyniki uzyskane w pracy 10

1.5 Publikacje autora związane z tematyką rozprawy 10

2 Analiza stanu wiedzy w zakresie metod detekcji dezinformacji 12

2.1 Problematyka dezinformacji 12

2.2 Wykrywanie dezinformacji 14

2.3 Ograniczenia dotychczasowych podejść 24

3 Opracowane hybrydowe metody ekstrakcji cech tekstu . . 25

3.1 Metoda LFM (Learned Fusion Method) 25

3.1.1 Etapy generowania wektorów osadzeń 27

3.1.2 Mechanizm fuzji wektorów osadzeń (Fusion Encoder) 28

3.1.3 Funkcja straty wykorzystana w procesie uczenia 29

3.2 Metoda GEM (Graph Embedding Method) 31
3.2.1 Graf spójności tekstu 31
3.2.2 Graf wynikania logicznego 33
3.2.3 Graf przyczynowo-skutkowy 34
3.2.4 Graf wiedzy zewnętrznej 35
3.2.5 Integracja i łączenie danych z wielu grafów 36

4 Metodyka badań i procedura eksperymentalna 38

4.1 Procedura eksperymentalna 38
4.2 Zbiory danych 39
4.2.1 Zbiory anglojęzyczne 40
4.2.2 Polskie zbiory binarne 42
4.2.3 Polski zbiór wieloetykietowy 43
4.3 Wstępne przetwarzanie danych 45
4.4 Ocena skuteczności klasyfikacji 46

5 Wyniki przeprowadzonych badań 49

5.1 Binarne zbiory anglojęzyczne 50
5.1.1 Word Embedding over Linguistic Features for Fake News Detection (WELFake) 50
5.1.2 ISOT Fake News detection dataset 53
5.2 Polskie binarne zbiory danych 55
5.2.1 Infotester 55

5.2.2 OpenFact 58

5.3 Polski zbiór wieloetykietowy 60

5.3.1 Wyniki dla metody referencyjnej 60

5.3.2 Wyniki dla metody LFM 68

5.3.3 Wyniki dla metody GEM 75

5.4 Analiza porównawcza z wynikami przedstawionymi w innych pracach naukowych 82

5.5 Analiza wpływu komponentów modelu na otrzymane wyniki . 84

6 Podsumowanie 86

Bibliografia 89

Spis tabel 98

Spis rysunków 102

Streszczenie

W niniejszej rozprawie skupiono się na opracowaniu hybrydowych metod ekstrakcji cech w celu poprawy skuteczności wykrywania dezinformacji w tekstach pochodzących z mediów cyfrowych. Problem dezinformacji, definiowany jako celowe rozpowszechnianie fałszywych lub wprowadzających w błąd treści, jest narastającym wyzwaniem, które może prowadzić do poważnych konsekwencji społecznych i politycznych. Tradycyjne metody ekstrakcji i klasyfikacji treści często okazują się niewystarczające do uchwycenia złożonych relacji semantycznych i kontekstualnych w tekstach, co utrudnia skuteczną detekcję dezinformacji. Rozprawa ta proponuje połączenie różnych podejść, aby lepiej uchwycić zarówno lokalne, jak i globalne wzorce obecne w tekście. W ramach rozprawy opracowano i zbadano dwie autorskie, innowacyjne metody:

- LFM (Learned Fusion Method) – która znacząco poprawiła skuteczność detekcji dezinformacji, dodatkowo rozwiązując problem niezbalansowanych danych w stosunku do metod referencyjnych. Architektura rozwiązania opiera się na ekstrakcji cech semantycznych za pomocą modelu DistilBERT oraz statystycznych przy użyciu metody TF-IDF. Integracja obu wektorów odbywa się w ramach sieci neuronowej typu enkoder.
- GEM (Graph Embedding Method) – również znacząco poprawiająca skuteczność detekcji w stosunku do metod referencyjnych. Model generuje reprezentacje wektorowe na podstawie czterech wyspecjalizowanych grafów, z których każdy odpowiada za analizę określonego aspektu tekstu: spójności, wynikania logicznego, relacji przyczynowo-skutkowych oraz wiedzy zewnętrznej. Wynikowe osadzenia są integrowane za pomocą konkatenacji, tworząc wspólny wektor cech wykorzystywany w końcowej detekcji.

Eksperymenty przeprowadzone na pięciu zbiorach danych, zarówno polskojęzycznych, jak i anglojęzycznych, o zróżnicowanej strukturze i stopniu zbalansowania, konsekwentnie potwierdziły hipotezę badawczą, że możliwe jest opracowanie hybrydowych metod ekstrakcji cech w celu podniesienia skuteczności detekcji dezinformacji w tekstach względem metod klasycznych. Uzyskane wyniki mają istotny potencjał praktyczny, gdyż mogą zostać wykorzystane w zaawansowanych narzędziach do automatycznej detekcji dezinformacji i przeciwdziałania jej rozprzestrzenianiu się.

Abstract

In this thesis, focuses on the development of hybrid feature extraction methods to improve the detection of disinformation in texts sourced from digital media. The problem of disinformation, defined as the intentional dissemination of false or misleading content, is a growing challenge that can lead to serious social and political consequences. Traditional extraction and classification methods often prove insufficient to capture the complex semantic and contextual relationships in texts, making effective disinformation detection difficult. This thesis proposes a combination of different approaches to better capture both local and global patterns present in text.

Two innovative methods were developed and tested as part of this thesis:

- LFM (Learned Fusion Method) – which significantly improved the detection of disinformation, further addressing the issue of data imbalance. The solution's architecture is based on the extraction of semantic features using the DistilBERT model and statistical features using the TF-IDF method. Integration of both vectors takes place within an encoder-type neural network.
- GEM (Graph Embedding Method) – which also significantly improved the detection performance. The model generates vector representations based on four specialized graphs, each responsible for analyzing a specific aspect of the text: coherence, logical entailment, cause-and-effect relationships, and external knowledge. The resulting embeddings are integrated using concatenation, creating a common feature vector used in the final detection.

Experiments conducted on five datasets, both Polish and English, with varying structures and levels of balance, consistently confirmed the research hypothesis that it is possible to develop hybrid feature extraction methods to improve the detection of disinformation in texts compared to traditional methods. The obtained results have significant practical implications, offering advanced tools for automatic disinformation detection and counteracting its spread.

1. Wstęp

1.1 Aktualność tematu

Współczesna rzeczywistość informacyjna charakteryzuje się rosnącą ilością danych generowanych w mediach cyfrowych, zwłaszcza w mediach społecznościowych, portalach informacyjnych i blogach. Zjawisko to prowadzi do łatwej dystrybucji nieprawdziwych informacji, co stanowi poważne wyzwanie dla społeczeństwa, rządów oraz firm technologicznych. Dezinformacja to złożone zagadnienie, które może przyjmować różne formy. Jednym z głównych celów dezinformacji jest celowe rozpowszechnianie fałszywych lub wprowadzających w błąd treści. Staje się ona coraz bardziej powszechnym problemem i może prowadzić do negatywnych skutków społecznych, takich jak destabilizacja polityczna, podziały społeczne (polaryzacja), a nawet zagrożenia dla zdrowia publicznego, jak miało to miejsce podczas pandemii COVID-19. Współczesna dezinformacja często wykorzystuje różne modalności przekazu takie jak tekst, obraz, dźwięk, a coraz częściej także format wideo. Wielomodalne treści zwiększają siłę oddziaływania przekazu, ponieważ angażują odbiorcę na wielu poziomach percepcji jednocześnie, co może wzmacniać jego podatność na manipulację.

Problem detekcji dezinformacji stawia przed badaczami liczne wyzwania, które wynikają m.in. z natury danych tekstowych, czy też kontekstu. W szczególności trudność polega na identyfikacji subtelnych różnic między prawdziwymi a fałszywymi wiadomościami, które mogą być trudne do odróżnienia dla tradycyjnych algorytmów klasyfikacyjnych. W tym kontekście kluczowym aspektem jest efektywna ekstrakcja cech, czyli proces przekształcania surowych danych tekstowych w reprezentacje numeryczne, które mogą być wykorzystane przez algorytmy uczenia maszynowego. Dotychczas stosowane metody, takie jak analiza częstotliwości występowania terminów (np. TF-IDF), modele wektorowe (np. Word2Vec, GloVe) czy metody oparte o transformery (np. BERT, GPT), choć skuteczne w wielu przypadkach, mogą być niewystarczające do wychwytywania złożonych relacji semantycznych i kontekstowych.

Podjęcie tematu zastosowania hybrydowych metod ekstrakcji cech w detekcji dezinformacji pozwala na zbadanie możliwości połączenia różnych podejść w celu zwiększenia efektywności modeli detekcji. Dzięki temu możliwe jest lepsze

uchwycenie zarówno lokalnych, jak i globalnych wzorców obecnych w tekście, co może prowadzić do wyższej skuteczności i precyzji w detekcji dezinformacji.

Ponadto hybrydowe metody ekstrakcji cech mogą zminimalizować problemy związane z przetwarzaniem i interpretacją stosunkowo małych zbiorów danych oraz ograniczyć moce obliczeniowe. Oprócz optymalizacji procesów obliczeniowych ekstrakcja istotnych cech zmniejsza ryzyko przeuczenia modeli, co jest szczególnie istotne w zadaniach, gdzie dostępne są ograniczone zasoby danych. Metody te pozwalają także na uchwycenie złożonych relacji semantycznych, subtelnych różnic w znaczeniu oraz adaptacji do dynamicznie zmieniających się wzorców informacji.

Warto również nadmienić, że odpowiednia reprezentacja cech wpływa na to, jak model interpretuje dane i uczy się na ich podstawie. Jeśli cechy są dobrze zdefiniowane, model jest w stanie zidentyfikować istotne wzorce, co znacząco poprawia jego wydajność i zdolność do rozwiązywania problemu, szczególnie w kontekście detekcji dezinformacji.

Dodatkowym aspektem motywującym podjęcie tego tematu jest dynamiczny rozwój technologii przetwarzania języka naturalnego (ang. *natural language processing*, NLP) oraz rosnące zainteresowanie sztuczną inteligencją w zadaniach związanych z analizą tekstu. Wprowadzenie hybrydowych metod ekstrakcji cech, obejmujących zarówno klasyczne techniki lingwistyczne, jak i nowoczesne modele uczenia maszynowego, może przynieść istotne korzyści dla dziedziny detekcji dezinformacji. Potencjalne implikacje tej pracy obejmują poprawę skuteczności narzędzi do analizy danych tekstowych oraz przeciwdziałanie szerzeniu się dezinformacji w sposób bardziej efektywny i skalowalny.

1.2 Cel, hipoteza i zakres pracy

Cel pracy

Celem niniejszej rozprawy jest opracowanie i empiryczna weryfikacja autorskich, hybrydowych metod ekstrakcji cech tekstu, łączących podejścia semantyczne, kontekstowe i strukturalne, w celu zwiększenia skuteczności detekcji dezinformacji.

Hipoteza i zakres pracy

Uwzględniając wyniki analizy literatury oraz potrzebę zwiększenia skuteczności metod detekcji dezinformacji, postawiono następującą hipotezę badawczą:

> Zastosowanie hybrydowych metod ekstrakcji cech, łączących reprezentacje semantyczne, logiczne, kontekstowe, przyczynowo-skutkowe oraz mechanizmy adaptacyjnego łączenia wektorów osadzeń, pozwala na istotne zwiększenie skuteczności detekcji dezinformacji w porównaniu z podejściami opartymi wyłącznie na modelach transformerowych.

W celu weryfikacji postawionej hipotezy oraz realizacji głównego celu pracy wyznaczono następujące zadania badawcze:

1. Opracowanie hybrydowej metody ekstrakcji cech uwzględniającej strukturę leksykalną oraz kontekst semantyczny.
2. Opracowanie metody łączącej informacje semantyczne, logiczne, kontekstowe oraz przyczynowo-skutkowe w celu poprawy jakości ekstrakcji cech.
3. Zaprojektowanie mechanizmu adaptacji wag, którego celem jest dostosowanie znaczenia poszczególnych typów reprezentacji (semantycznej oraz leksykalnej).
4. Badanie skuteczności metod hybrydowych w detekcji dezinformacji w porównaniu z referencyjnym podejściem opartym wyłącznie na modelach transformerowych.
5. Przeprowadzenie eksperymentalnej ewaluacji proponowanych metod hybrydowych w zestawieniu z innymi podejściami detekcji dezinformacji opisanymi w literaturze.

1.3 Struktura pracy

Niniejsza praca składa się z 6 rozdziałów. Pierwszy rozdział stanowi wstęp do pracy, gdzie przedstawiono motywację, a także hipotezę i cel pracy. W rozdziale 2 znajduje się przegląd literatury bezpośrednio związany z tematem niniejszej rozprawy. Omówione zostały w nim podejścia dotyczące ekstrakcji cech w problemach dezinformacji. W rozdziale 3 zaprezentowano własne propozycje metod

ekstrakcji cech. Z kolei rozdział 4 zawiera omówienie metodyki badań oraz charakterystyką zbiorów danych. W rozdziale 5 zaprezentowano wyniki przeprowadzonych badań dla polskich i angielskich zbiorów danych. Rozprawa zakończona jest rozdziałem 6, w którym podsumowano uzyskane wyniki oraz opisano wnioski.

1.4 Główne wyniki uzyskane w pracy

Do najważniejszych wyników zawartych w rozprawie zaliczam:

1. Opracowanie hybrydowej metody ekstrakcji cech uwzględniającej strukturę leksykalną oraz kontekst semantyczny - metoda LFM.
2. Opracowanie metody łączącej informacje semantyczne, logiczne, kontekstowe oraz przyczynowo-skutkowe w celu poprawy jakości ekstrakcji cech - metoda GEM.
3. Zaprojektowanie mechanizmu adaptacji wag, którego celem jest dostosowanie znaczenia poszczególnych typów reprezentacji - FusionEncoder w metodzie LFM.
4. Uzyskanie zbalansowanej dokładności wyższej o około 14% - 18% w odniesieniu do metody opartej wyłącznie na modelach transformerowych.

1.5 Publikacje autora związane z tematyką rozprawy

Poniższe publikacje, w których autor rozprawy miał istotny udział, stanowią spójny dorobek badawczy rozwijający zagadnienia przedstawione w rozprawie. Są one bezpośrednio powiązane z zagadnieniami przedstawionymi w rozdziałach 3 (Opracowane hybrydowe metody ekstrakcji cech tekstu) oraz 4 (Metodyka badań i procedura eksperymentalna) niniejszej rozprawy doktorskiej. Przedstawione w nich badania dokumentują kolejne etapy ewolucji metod detekcji dezinformacji.

Lista publikacji powiązanych z tematem rozprawy

Publikacje wykazane w niniejszej sekcji stanowią podstawy teoretyczne i empiryczne, na których oparto niniejsze badania. Przyczyniły się one także do powstania autorskich metod opisanych szczegółowo w tej pracy.

1. Kozik Rafał, **Kątek Gracjan**, Gackowska Marta, Kula Sebastian, Komorniczak Joanna, Ksieniewicz Paweł, Pawlicka Aleksandra, Pawlicki Marek, Choraś Michał. (2024). ***Towards explainable fake news detection and automated content credibility assessment: Polish internet and digital media use-case.*** Neurocomputing.
MNiSW: 140, IF: 6.5

2. **Kątek Gracjan**, Gackowska Marta, Komorniczak Joanna, Ksieniewicz Paweł, Kozik Rafał, Pawlicki Marek, Choraś Michał. (2024). ***Involving society to protect society from fake news and disinformation: Crowdsourced datasets and text reliability assessment***. (ACIIDS 2024) In Lecture Notes in Computer Science (pp. 384–395). Springer Nature Singapore.
MNiSW: 70, CORE: B

3. Gackowska Marta, **Kątek Gracjan**, Śrutek Mścisław, Kozik Rafał, Choraś Michał. (2023). ***Document Annotation Tool for News Content Analysis.*** Lecture Notes in Networks and Systems, vol 766. Springer, Cham.
MNiSW: 20

4. **Kątek Gracjan**, Kozik Rafał, Pawlicka Aleksandra, Pawlicki Marek, Choraś Michał. (2025). ***In Depth Analysis for Securing The Truth: Addressing The Fake News Challenge with Graph Neural Networks.*** Neurocomputing.
MNiSW: 140, IF: 6.5

5. **Kątek Gracjan**, Gackowska-Kątek Marta, Kozik Rafał, Pawlicka Aleksandra, Pawlicki Marek, Choraś Michał, Choraś Ryszard. (2026). ***Ensemble-based Fake News and Disinformation Detection using Crowdsourced Dataset.*** Logic Journal of the IGPL. (w trakcie wydania)
MNiSW: 100, IF: 0.8

Udział w projektach badawczych

Niniejsza rozprawa powstała równolegle z pracami prowadzonymi w ramach projektu badawczego SWAROG (System Wykrywania Dezinformacji Metodami Sztucznej Inteligencji), realizowanego w programie INFOSTRATEG finansowanym przez Narodowe Centrum Badań i Rozwoju. Część koncepcji oraz rozwiązań przedstawionych w pracy była rozwijana i weryfikowana w wymienionym projekcie.

2. Analiza stanu wiedzy w zakresie metod detekcji dezinformacji

Problematyka badawcza dotycząca wykrywania dezinformacji stanowi złożone i wieloaspektowe zagadnienie obejmujące szereg wyzwań związanych z detekcją, analizą oraz identyfikacją fałszywych treści. Należy przy tym pamiętać, że fałszywe wiadomości mogą powstawać w sposób zamierzony, tak aby wprowadzić odbiorcę w błąd [1], jak i niezamierzony wynikający z niestarannego doboru i sposobu przekazu treści [2]. Istotny w tym aspekcie jest również fakt, że fałszywe wiadomości rozprzestrzeniają się zdecydowanie szybciej niż wiadomości zawierające prawdziwe i rzetelne informacje [3]. Problem ten jest szczególnie widoczny w wiadomościach o treściach politycznych [4] oraz ważnych aspektach zdrowia, co miało miejsce podczas pandemii COVID-19 [5]. Dezinformacja dotyczy również aspektów społeczno-gospodarczych i może z powodzeniem negatywnie wpływać na kluczowe aspekty gospodarki, między innymi poprzez zaburzanie łańcucha dostaw [6].

Nie należy zapominać jak istotną rolę w zakresie rozprzestrzeniania fałszywych informacji, odgrywają media społecznościowe [7], gdyż dla wielu milionów użytkowników na całym świecie stanowią one źródło różnego rodzaju wiadomości o zasięgu zarówno globalnym, jak i lokalnym [8]. Fałszywe wiadomości mogą mieć przy tym różnorodną postać (tekst, obraz, nagrania audio i wideo). W niniejszej pracy skupiono się wyłącznie na tekście. Oczywiście w świetle powyższych faktów związanych z fake newsami podstawą jest ich prawidłowa detekcja.

W niniejszym rozdziale przedstawione zostaną zagadnienia związane z problematyką wykrywania dezinformacji, ze szczególnym uwzględnieniem technik wstępnego przetwarzania tekstu oraz metod klasyfikacji tekstu stosowanych w celu detekcji fałszywych informacji.

2.1 Problematyka dezinformacji

Fałszywe informacje (ang. *fake news*), dezinformacja (ang. *dissinformation*) i informacje wprowadzające w błąd (ang. *misinformation*) to pojęcia wskazujące na

wspólne zjawisko zakłócenia rzetelnego obiegu informacji w społeczeństwie. Jest ono narastającym problemem [9] szczególnie w dobie rozwoju usług cyfrowych, popularności platform społecznościowych i portali internetowych, które sprzyjają możliwości ich szybkiego rozprzestrzeniania się. Jest to nie tylko problem społeczny, ale co należy również mieć na uwadze, przyczynia się do zagrożeń związanych z cyberbezpieczeństwem.

Problem jest istotny, co znajduje swoje potwierdzenie w pracach i opracowaniach Komisji Parlamentu Europejskiego.

> Według dokumentu: „Zwalczanie dezinformacji w internecie: podejście europejskie" [10] dezinformację definiuje się jako informacje, które można zweryfikować jako fałszywe lub wprowadzające w błąd, a które są celowo tworzone, rozpowszechniane i prezentowane w sposób mający na celu osiągnięcie zysków finansowych lub świadome wprowadzenie społeczeństwa w błąd, co może prowadzić do szkody dla interesu publicznego.

W pracy poświęconej definicji fałszywych informacji [11] określa się je jako celowe przedstawianie (zwykle) fałszywych lub wprowadzających w błąd twierdzeń jako wiadomości, które są mylące, a jednocześnie manipulują procesami poznawczymi odbiorców. Jak słusznie zauważono w pracy [12] fałszywe informacje przybierają postać fałszywych stwierdzeń, które mogą być lub nie być powiązane z prawdziwymi wydarzeniami, przybierając często formę sensacyjnych nagłówków, zmanipulowanych obrazów lub celowo zniekształconej treści. Ich głównym celem jest przyciągnięcie uwagi odbiorcy, skłonienie go do kliknięcia, udostępnienia lub dalszego rozpowszechnienia przekazu. To zjawisko z kolei określane jest mianem clickbaitu [13].

Ponadto, w przypadku rozprzestrzeniania się dezinformacji, istotne jest to, że gdy fałszywa informacja zostaje opublikowana na platformie internetowej, często dochodzi do jej szybkiego rozpowszechnienia za pośrednictwem mechanizmów udostępniania i rekomendacji treści, co prowadzi do powstania gęstej sieci powiązań między użytkownikami, artykułami, a także źródłami [14]. W takiej sieci można zaobserwować powtarzalność określonych narracji, obecność tych samych autorów lub domen internetowych, a także korelacje tematyczne między różnymi wpisami, co może świadczyć o zamierzonym działaniu mającym na celu manipulację informacyjną. Wykorzystując narzędzia sztucznej inteligencji, w tym algorytmy przetwarzania języka naturalnego oraz analizę grafów, możliwe jest identyfikowanie struktur rozpowszechniania dezinformacji [15].

Zjawisko dezinformacji jest przedmiotem badań wielu dyscyplin naukowych, jednak istotny wkład w jego wykrywanie wnoszą badania z zakresu informatyki [16]. Szczególnie widoczne jest to w liczbie badań prowadzonych, które koncentrują się na wykrywaniu i analizie cech dezinformacji – aż 168 z 351 analizowanych empirycznych tematów badawczych, zidentyfikowanych przez autorów pochodzi z tej dziedziny [16]. Co więcej, informatyka zdecydowanie dominuje pod względem stosowania metod obliczeniowych (w 180 z 232 analizowanych badań [16]), co pokazuje, jak istotny wkład mają metody komputerowe w identyfikację dezinformacji.

W ostatnich latach obserwuje się dynamiczny rozwój dużych modeli językowych (ang. *large language model*, LLM) oraz ich szerokie zastosowanie. Modele te będące zaawansowanymi modelami głębokiego uczenia maszynowego pozwalają na generowanie oraz interpretację ludzkiego języka. Poprzez architekturę opartą na transformerach modele te (w zadaniach analizy i detekcji fałszywych informacji) pozwalają rozpoznawać wzorce językowe i semantyczne (wskazujące na manipulację [17]) oraz weryfikować fakty poprzez porównanie ich z szeroką bazą dokumentów tekstowych [18]. Jednocześnie autorzy pracy [19] wskazują, że w zadaniach klasyfikacji tekstu modele z rodziny BERT mogą osiągać lepsze wyniki niż duże modele językowe w zadaniach opartych na analizie wzorców. Ich badania sugerują jednak, że w sytuacjach, gdy potrzebna jest dodatkowa wiedza lub głębsza semantyka, modele LLM charakteryzują się większą skutecznością.

2.2 Wykrywanie dezinformacji

W celu uporządkowania obszaru badań w zakresie dezinformacji, autorzy licznych publikacji podejmują próby systematyzacji istniejących podejść, proponując różnorodne klasyfikacje metod wykrywania dezinformacji. Opracowania te różnią się stopniem szczegółowości oraz przyjętymi kryteriami podziału, jednak ich wspólnym celem pozostaje pogłębione zrozumienie mechanizmów determinujących skuteczną identyfikację treści nieprawdziwych.

Detekcja fałszywych informacji w tekście może być rozpatrywana jako szczególny przypadek problemu klasyfikacji tekstu [20]. W pracy [21] zaproponowany został podział na 3 główne metody:

- oparte na wiedzy (ang. *knowledge-based*), które wykorzystują zewnętrzne źródła wiedzy, takie jak bazy danych faktów, ontologie czy repozytoria informacji, wiedzę ekspertów w celu porównania treści analizowanego tekstu z ustalonymi faktami;

- oparte na cechach (ang. *feature-based*), które koncentrują się na analizie wewnętrznych cech tekstu (np. statystyk lingwistycznych, składniowych lub semantycznych), bez odniesienia do wiedzy zewnętrznej;
- oparte na modalności (ang. *modality-based*), które integrują informacje pochodzące z różnych źródeł danych (np. tekst, obraz, dźwięk).

W kolejnych artykułach zastał dostarczony bardziej szczegółowy podział, jak na przykład w pracy [22], gdzie zostały wyróżnione następujące metody:

- strategia oparta na wiedzy (ang. *knowledge-based strateg*y), odnoszące się do wykorzystania istniejących faktów oraz technik weryfikacji faktów;
- podejście lingwistyczne (ang. *language approach*), skupiające się na analizie struktury językowej, stylu wypowiedzi, używanego słownictwa czy wykrywania charakterystycznych wzorców retorycznych;
- podejście niezależne od tematu (ang. *topic-agnostic approach*), które koncentrują się na wykrywaniu dezinformacji bez względu na konkretną tematykę tekstu, dzięki czemu mogą być stosowane uniwersalnie;
- metoda hybrydowa (ang. *hybrid method*), łączące różne strategie w celu zwiększenia skuteczności detekcji dezinformacji;
- podejście wykorzystujące uczenie maszynowe (ang. *machine learning approach*), w którym modele uczone na dużych zbiorach danych uczą się rozróżniać pomiędzy treściami prawdziwymi a fałszywymi na podstawie wzorców zawartych w danych treningowych.

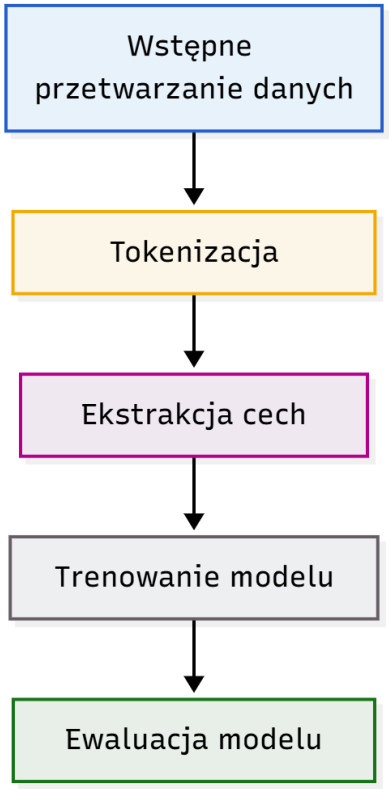
W związku ze znaczną dynamiką powstawania dezinformacji niezbędne staje się zastosowanie wieloaspektowego podejścia do ich wykrywania. Podejście bazujące na wiedzy dotyczy sytuacji, w których wiarygodność informacji oceniana jest na podstawie wiedzy eksperckiej oraz analiz, jak również porównywania wiadomości z wiarygodnymi źródłami [23]. Podejście językowe oraz metody bazujące na cechach opierają się na analizie stylu pisania, gramatyki często połączonej z analizą emocji zawartych w tekście, jak i poszukiwaniu nieścisłości lub sprzeczności w sformułowaniach. Często również w przypadku tych metod stosowana jest analiza struktury zdań, częstość używanych słów czy analiza statystyk tekstu [24, 25].

W odniesieniu do wcześniej opisanych metod wykrywania dezinformacji istotną rolę odgrywają techniki przetwarzania języka naturalnego, wspierane przez

rozwiązania z zakresu sztucznej inteligencji, a w szczególności uczenia maszynowego [26].

Techniki przetwarzania języka naturalnego wykorzystywane są m.in. w zadaniach klasyfikacji tekstu, polegających na przypisywaniu dokumentom jednej lub wielu kategorii tematycznych na podstawie ich treści. Takie podejście pozwala na systematyczne porządkowanie informacji oraz ocenę ich zgodności z zaufanymi źródłami, co stanowi istotne wsparcie dla metod bazujących na wiedzy, analizie językowej oraz cechach tekstu [27].

Na rysunku 2.1 przedstawiono schemat procesu klasyfikacji tekstu w celu wykrywania dezinformacji. Obejmuje on kilka etapów, mających na celu przypisanie dokumentowi jednej lub wielu klas na podstawie jego treści. Do kluczowych faz tego procesu należą: wstępne przetwarzanie danych, tokenizacja, ekstrakcja cech, trenowanie modelu oraz jego ewaluacja.



Rysunek 2.1: Etapy procesu tworzenia modelu służącego do klasyfikacji tekstu. [Opracowanie własne]

Wstępne przetwarzanie danych (ang. *preprocessing*) ma na celu oczyszczenie i normalizację danych tekstowych w celu redukcji szumu informacyjnego oraz ujednolicenia reprezentacji tekstu [28]. Typowe operacje obejmują m.in.: usuwanie znaków specjalnych, liczb i słów przystankowych (ang. *stopwords*), sprowadzanie słów do formy podstawowej - lematyzacja lub stemming [29], konwersję do małych liter (ang. *lowercasing*), usuwanie nadmiarowych białych znaków.

Tokenizacja polega na podziale ciągu znaków na mniejsze jednostki zwane tokenami, którymi najczęściej są słowa, podsłowa lub znaki. W zależności od modelu i języka stosuje się różne strategie tokenizacji [30], takie jak na przykład: whitespace tokenization, Byte-Pair Encoding (BPE) [31], WordPiece [32], SentencePiece [33] czy SaGe [34]. Tokeny stanowią podstawową jednostkę dalszego przetwarzania.

Ekstrakcja cech tekstu jest kluczowym etapem w wykrywaniu dezinformacji oraz szerzej, w zadaniach przetwarzania języka naturalnego. W przypadku detekcji dezinformacji ekstrakcja cech polega na wydobyciu reprezentatywnych informacji z tekstu, które mogą być użyte do oceny prawdziwości, intencji lub stylu zawartego w nim przekazu. To z kolei jest istotnym wyzwaniem w celu zapewnienia autentyczności i szeroko rozumianego cyberbezpieczeństwa oraz bezpieczeństwa społeczeństwa.

Wśród technik ekstrakcji cech [35] wyróżnia się metody podstawowe, takie jak:

- n-gram [36] do modelowania sekwencji słów lub znaków w tekście;
- bag-o-words [37] gdzie tekst jest reprezentowany przez zbiór słów;
- metoda TF służąca do obliczenia częstotliwości występowania słowa w tekście [38];
- TF-IDF [39] jako rozbudowana metoda gdzie oceniane jest również znaczenie i ważność słowa w tekście.

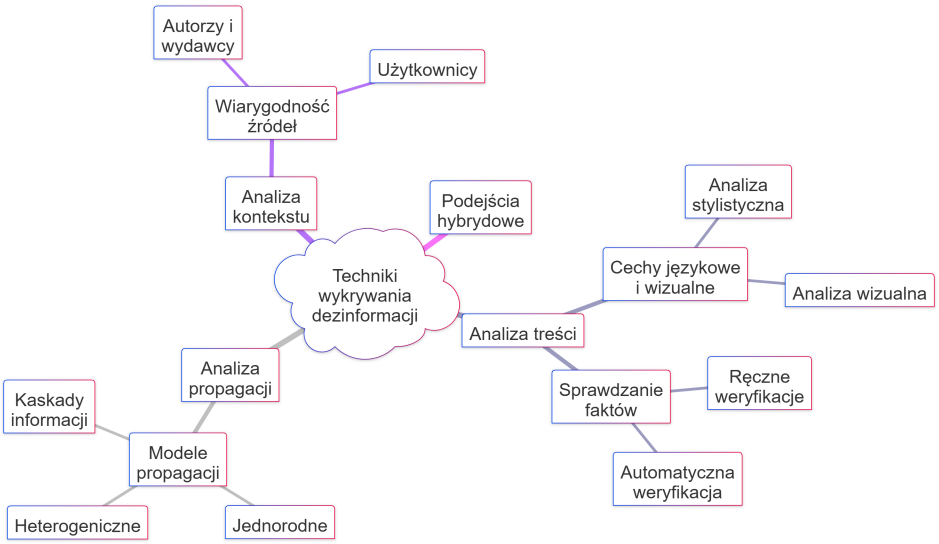
Z kolei do bardziej zaawansowanych metod zalicza się metody obejmujące transformery [40] takie jak BERT [41], RoBERTa [42], DistilBERT [43]. Dodatkowo warto wskazać, że w literaturze znajdują się prace wskazujące, że metody ekstrakcji cech oparte na modelach transformerowych, takich jak BERT, przewyższają pod względem dokładności klasyczne podejścia bazujące na TF-IDF [44]. Umożliwiają one dwukierunkową reprezentację tekstu, gdzie znaczenie poszczególnych

jednostek językowych jest modelowane z uwzględnieniem zarówno kontekstu poprzedzającego, jak i następującego w sekwencji. Dzięki tym właściwościom modele transformerowe znajdują zastosowanie nie tylko w zadaniach klasyfikacji tekstu, lecz także w innych zagadnieniach NLP, między innymi takich jak:

- analiza emocji w tekście (ang. *text sentiment analysis*) [45, 46];
- udzielanie odpowiedzi na pytania (ang. *question answering*) [47].

W zadaniach przetwarzania języka naturalnego oraz klasyfikacji tekstu istotne są hybrydowe techniki ekstrakcji cech, które integrując różnorodne podejścia analityczne, jednocześnie wykazują się zwiększoną skutecznością klasyfikacyjną, co zostało potwierdzone m.in. w pracy [48]. Szczegółowego omówienia istniejących technik ekstrakcji cech dokonano w [49]. Autorzy pracy wskazali na połączenie technik takich jak: TF, TF-IDF, Word2Vec, BERT, które następnie są wykorzystywane w połączeniu z klasyfikatorami, takimi jak SVM, KNN, Random Forest, AdaBoost, XGBoost, MLP czy splotowe sieci neuronowe (ang. *convolutional neural networks*, CNN) [50]. Literatura przedmiotu dostarcza również przykładów metod łączących różne techniki ekstrakcji cech, np. TF-IDF, Word2Vec oraz sieci LSTM (ang. *Long Short-Term Memory*) [51]. Bardziej zaawansowane podejścia wykorzystują osadzanie węzłów kontekstowych (ang. *contextual node embedding*), grafy hierarchiczne oraz dynamiczną fuzję z wykorzystaniem BERT [52].

Na Rysunku 2.2 graficznie przedstawiono omówione metody wykrywania dezinformacji. Zobrazowano na nim podział technik na cztery główne kategorie: analizę treści, analizę propagacji, analizę kontekstu oraz podejścia hybrydowe, które łączą w sobie elementy z pozostałych grup. Każda z tych kategorii obejmuje szereg specyficznych technik, od analizy cech językowych i wizualnych, przez ręczne i automatyczne sprawdzanie faktów, po modelowanie dynamiki rozprzestrzeniania się informacji w sieciach oraz ocenę wiarygodności źródeł.



Rysunek 2.2: Przegląd technik wykrywania dezinformacji (na podstawie [53])

W celu usystematyzowania informacji na temat metod służących wykrywaniu fałszywych wiadomości w różnych językach oraz zbiorów danych (na których trenowano modele) opracowano tabelę 2.1. Przedstawione w niej prace koncentrują się na wykorzystywaniu różnych modeli uczenia maszynowego i głębokiego uczenia do poprawy dokładności wykrywania dezinformacji głównie w wiadomościach i postach oraz mediach społecznościowych.

Tabela 2.1: Podsumowanie zbiorów danych i technologii do wykrywania fałszywych wiadomości w różnych językach

Autorzy	**Język**	**Dane**	**Technologia**	**Rok**
Martínez-Gallego et al. [54]	Hiszpański	Spanish Fake News Corpus (971 wiadomości) + Fake news in Spanish (1600 wiadomości), łącznie 2571 próbek	BETO + LSTM	2021
Blanco-Fernández et al. [55]	Hiszpański	Syntetyczny zbiór 57 231 artykułów politycznych – dane uzyskane przez web scraping i modele generatywne	Fine-tuned BERT / RoBERTa	2024

Autorzy	Język	Dane	Technologia	Rok
Ibañez-Lissen et al. [56]	Hiszpański	Zbiory danych z wiadomości i mediów społecznościowych (fałszywe wiadomości polityczne)	GCN + BERT	2024
Moreno-Vallejo et al. [57]	Hiszpański	Dane z mediów społecznościowych i artykułów prasowych	Porównanie MLP, CNN, LSTM	2023
Catelli et al. [58]	Włoski	Recenzje dotyczące dziedzictwa kulturowego we Włoszech	BERT + ELECTRA + analiza sentymentu	2023
Buzea et al. [59]	Rumuński	Wiadomości online po rumuńsku	SVM, NB, LSTM, CNN, GRU, Ro-BERT	2022
Bucos et al. [60]	Rumuński	Dane z Factual.ro	Back Translation, Easy Data Augmentation	2023
Dinu et al. [61]	Rumuński	Rumuńskie wiadomości online	LR, SVM, RF, SD Classifier	2022
Valeanu et al. [62]	Rumuński	Rumuńskie tweety nt. szczepień	SVM, MLP, RF, RCNN, BERT	2023
Daria-Mihaela et al. [**?**]	Rumuński	RoCliCo: 8313 artykułów prasowych (clickbait)	RF, SVM, BiLSTM, Ro-BERT	2023
Moisi et al. [63]	Rumuński	FakeRom – 1000+ artykułów z Veridica	NB, LR, SVM, BERT	2024
Farooq et al. [64]	Urdu	4097 wiadomości z 9 dziedzin	TF-IDF, BoW, SVM, k-NN	2023
Iqbal et al. [65]	Urdu	12 047 tweetów	TF, IDF, SVM, CNN, RNN	2024
Munir et al. [66]	Urdu	Tekst + obrazy	-	2024
Al Ghamdi et al. [67]	Ang., Arb., Urdu	Twitter, artykuły online	TF-IDF, LR, BERT, RF	2023
Harris et al. [68]	Urdu	UrduFake@FIRE2020: 1300 wiadomości	ELECTRA, mBERT, XLM-R	2025

Autorzy	Język	Dane	Technologia	Rok
Sorour et al. [69]	Arabski	1475 prawdziwych, 3152 fałszywych wiadomości	CNN + LSTM	2022
Azzeh et al. [70]	Arabski	Dane z sieci i Twittera	AraBERT, MARBERT, SVM, NB	2025
Almandouh et al. [71]	Arabski	228 461 rekordów z arabskich zbiorów	FastText, LR, XGB, AdaBoost	2024
Mohawesh et al. [72]	Wielojęzyczny	ENG-HIN, ENG-IDN, ENG-SWA, ENG-VIE	Capsule NN, mBERT, XLM-R	2023
Keya et al. [73]	Angielski	WelFake, LIAR	BERT, FakeStack	2023
Han et al. [74]	Angielski	FakeNewsNet (politifact, gossipcop)	Grafowe sieci neuronowe	2020
Lu et al. [75]	Angielski	Twitter15, Twitter16	Sieć ko-uwagi + grafy	2020
Roy et al. [76]	Bengalski	BanFakeNews	Bi-GRU	2024
Malla et al. [77]	Angielski	FakeNewsNet, Gossip, tweet'y	GAT	2024
Frisli [78]	Norweski	426 262 tweet'y COVID-19 (5,11% dezinformacja)	Semi-supervised LR + class weights	2025
Wanda et al. [79]	Indonezyjski	Fałszywe wiadomości po indonezyjsku	GRN (Generative Round Networks)	2024
Canhasi et al. [80]	Albański	Albańskie oznaczone wiadomości	LR, NB, SVM, XGBoost	2022
Isa et al. [81]	Indonezyjski	COVID-19 news (Indonezja)	IndoBERT	2022

Dinu, Fusu i Gifu [61] zaprezentowali w swojej pracy podejście wykorzystujące klasyfikatory SVM i regresję logistyczną (ang. *logistic regression*, LR) oraz techniki przetwarzania tekstu takie jak TF-IDF i 300-wymiarowe osadzenia wektorowe CoRoLa. Autorzy w pracy skupili się na języku rumuńskim, a najlepsze wyniki uzyskali przy użyciu SVM i LR. Valeanu et al. [62] w podobny sposób zaprezentowali metodę wykrywania dezinformacji na podstawie postów z platformy Twitter o szczepieniach z wykorzystaniem metod takich jak SVM, MLP i las losowy (ang. *random forest*, RF). Zaproponowane metody osiągnęły wyniki AUC zakresie od 0.744 do 0.858, co wskazuje na wysoką wydajność algorytmów

klasyfikacyjnych.

Bucos i Țucudean [60] użyli zbioru danych Veridica z rumuńskimi artykułami prasowymi do analizy fałszywych wiadomości. W swoim badaniu wykorzystali klasyfikatory, takie jak ekstremalnie losowe drzewa (ang. *extremely randomized trees*, Extra Trees), RF i SVM, uzyskując najlepsze wyniki przy użyciu techniki Back Translation. Moisi, Țucudean i Ionescu [63] zaprezentowali podejście wykorzystujące BiLSTM i RoBERTa-large do wykrywania fałszywych wiadomości w rumuńskim, osiągając dokładność 96.5% dla modelu opartego na BERT.

W zakresie języka arabskiego, Al Ghamdi et al. [67] zebrali dane z Twittera i artykułów, tworząc korpus, który zawiera zarówno fałszywe, jak i prawdziwe wiadomości. Analizowali różne modele klasyfikacyjne, takie jak Naive Bayes, Logistic Regression, SVM i BERT, osiągając dokładność 90% przy użyciu modelu BERT. Sorour i Abdelkader[69] zastosowali podejście hybrydowe, wykorzystując CNN i LSTM do wykrywania fałszywych wiadomości w języku arabskim, osiągając dokładność 81%.

Harris, Hadi, Ahmad et al. [68] opracowali model wykrywania fałszywych wiadomości dla języka urdu, wykorzystując zbiory danych UrduFake@FIRE2020, które zawierały zarówno prawdziwe, jak i fałszywe wiadomości z pięciu dziedzin: zdrowie, sport, rozrywka, technologia oraz biznes. Użyli modeli, takich jak ELECTRA, mBERT i XLM-RoBERTa, osiągając dokładność 91,4% przy użyciu metody ensemble.

W pracy Azzeh, Qusef i Alabboushi [70], autorzy użyli sześciu technik reprezentacji tekstu oraz pięciu głębokich modeli osadzenia słów, takich jak AraBERT, AraELECTRA, ARBERT, MARBERT i CAMeLBERT, do wykrywania fałszywych wiadomości w języku arabskim. Najlepsze wyniki uzyskano przy użyciu CAMeLBERT połączonego z głęboką siecią neuronową (DNN), osiągając F_1-score na poziomie 71.3% i AUC 79.1%. Chociaż wiele badań koncentruje się na pojedynczych językach, inne prace (np. [72]), uwzględniają wiele języków, takich jak: angielski, hindi, suahili, wietnamski i indonezyjski. Ich podejście oparte było na sieciach neuronowych kapsułkowych, które wykrywały fałszywe wiadomości z poprawą o około 3.97% w porównaniu do modeli bazowych.

W kontekście języka angielskiego, Keya et al. [73] użyli osadzeń BERT w połączeniu z głębokim CNN i LSTM w modelu FakeStack, który osiągnął dokładność powyżej 97% w wykrywaniu fałszywych wiadomości. Han, Karunasekera i Leckie [74] wykorzystali sieci neuronowe oparte na grafach (GNN) do analizy wzorców rozprzestrzeniania się fałszywych wiadomości w sieci społecznościo-

wej. Ich podejście, które uwzględniało dane z Twittera i cechy użytkowników, osiągnęło dokładność 83%.

Wśród najnowszych prac w tej dziedzinie warto wspomnieć badania Roy et al. [76], którzy stworzyli nową metodę wykrywania fałszywych wiadomości w języku bengalskim, osiągając dokładność 99% przy użyciu modelu Bidirectional Gated Recurrent Unit (GRU). Malla i Banka [77] zaprezentowali podejście wykorzystujące Graph Attention Networks (GAT), które uwzględniało preferencje użytkowników i kontekst społeczny, osiągając dokładność 98%.

Inne innowacyjne podejścia obejmują pracę Frisli [78], gdzie zastosowano półnadzorowane podejście do samodzielnego treningu w klasyfikacji dezinformacji, osiągając dokładność powyżej 98%. Z kolei Wanda i Diqi [79] w swoim badaniu nad językiem indonezyjskim wykorzystali nową architekturę Generative Round Networks (GRN), osiągając dokładność 94,33%.

W przypadku języka albańskiego Canhasi et al. [80] użyli klasycznych technik klasyfikacyjnych, takich jak KNN, XGBoost oraz osadzeń BERT i FastText do wykrywania fałszywych wiadomości. Ich wyniki pokazały skuteczność metod klasyfikacyjnych w tym języku. Z kolei dla języka indonezyjskiego, Isa, Nico i Permana [81] użyli modelu IndoBERT, który osiągnął dokładność 94,66% w wykrywaniu fałszywych wiadomości, szczególnie tych związanych z COVID-19.

Popularność w kontekście detekcji dezinformacji zdobywają heterogeniczne grafy, które umożliwiają rozszerzenie możliwości dużych modeli językowych (LLM) poprzez integrację różnych typów encji i relacji. Xie i in. [82] w swojej pracy wykorzystali heterogeniczny graf zawierający węzły wiadomości, encji i tematów do modelowania treści wiadomości. Wiedza z trzech grafów wiedzy jest następnie łączona w celu wzbogacenia procesu decyzyjnego. Z kolei Kang et al. [83] zastosowali podejście oparte na heterogenicznych grafach, aby wykorzystać różnorodne powiązania między wiadomościami, takie jak ich kontekst czasowy, treść, tematykę i źródło, w celu identyfikacji fałszywych informacji. W artykule zaproponowano budowę heterogenicznego grafu, nazwanego News Detection Graph (NDG), zawierającego różne typy węzłów i krawędzi, co pozwala na integrację wieloaspektowych danych pochodzących z wielu wiadomości. Sun et al. [84] wprowadza model wykrywania fałszywych wiadomości oparty na percepcji tematu, gdzie określona wiadomość jest dzielona na zdania, a graf heterogeniczny tworzony jest z węzłów reprezentujących zdania, tematy i encje. W dalszym etapie następuje ekstrakcja cech i porównania encji w celu oceny spójności semantycznej. Równie popularne jest stosowanie grafowych sieci neuronowych [85]. Karnyoto et al. [86] w swojej pracy wykorzystali heterogeniczną grafową sieć neuronową do

wykrywania dezinformacji na temat pandemii COVID19 [86]. Aby utworzyć sieć neuronową grafu, zbudowali węzły i krawędzie w grafie, a także węzły słowo-słowo i słowo-dokument.

2.3 Ograniczenia dotychczasowych podejść

Analiza literatury wskazuje na różnorodność metod i podejść skutecznych w wykrywaniu fałszywych wiadomości, obejmujących zarówno klasyczne techniki, jak i nowoczesne modele głębokiego uczenia, które osiągają zadowalające wyniki w różnych językach i kontekstach. Przeprowadzony przegląd pozwala zidentyfikować kilka istotnych obszarów wymagających dalszych badań.

Po pierwsze, zauważalne jest niewystarczające wykorzystanie grafów w językach innych niż angielski. Opisane w literaturze podejścia grafowe koncentrują się głównie na anglojęzycznych źródłach informacji, podczas gdy potencjał zastosowania grafów w językach mniej zasobnych pozostaje w dużej mierze niezbadany.

Po drugie, zauważalny jest niedostatek badań nad efektywnością hybrydowych metod ekstrakcji cech. Choć wskazuje się na możliwość łączenia technik takich jak TF-IDF, Word2Vec, BERT, CNN czy LSTM, brakuje systematycznych analiz porównawczych, które oceniałyby skuteczność różnych konfiguracji w odmiennych domenach i językach.

Po trzecie, istotnym ograniczeniem pozostaje silna zależność od niewielkich, najczęściej binarnych zbiorów danych. Ogranicza to nie tylko możliwość porównań między modelami, lecz także obniża potencjał uogólniania wyników na bardziej zróżnicowane i rzeczywiste scenariusze.

3. Opracowane hybrydowe metody ekstrakcji cech tekstu

W niniejszym rozdziale przedstawiono dwie autorskie, hybrydowe metody ekstrakcji cech, opracowane w ramach pracy. Pierwszą z nich jest metoda LFM, szczegółowo opisana w podrozdziale 3.1, natomiast drugą – metoda GEM, której charakterystyka znajduje się w podrozdziale 3.2. Podrozdziały te zawierają również szczegółową charakterystykę metod składowych, wchodzących w skład opracowanych podejść hybrydowych.

3.1 Metoda LFM (Learned Fusion Method)

Pierwszą zaproponowaną w rozprawie doktorskiej hybrydową metodą ekstrakcji cech jest metoda LFM (Learned Fusion Method). Metoda ta wykorzystuje model DistilBERT (opisany w podrozdziale 3.1.1.1) oraz statystyczną metodę TF-IDF (opisaną w podrozdziale 3.1.1.2) w celu uzyskania reprezentacji tekstu z uwzględnieniem zarówno struktury leksykalnej jak i kontekstu semantycznego. Dodatkowo, w celu połączenia dwóch różnych reprezentacji wektorowych w jeden spójny wektor, została opracowana sieć neuronowa typu encoder (określona jako FusionEncoder). Metoda ta składa się z kilku etapów, które zostały przedstawione na rysunku 3.1.

W pierwszym kroku reprezentacja wejściowa (każdy dokument tekstowy T) jest przekształcana na dwa oddzielne wektory:

- Reprezentacja semantyczna za pomocą modelu DistilBERT:

$$x_{bert} = DistilBERT(T) \in \mathbb{R}^{d_{bert}} \tag{3.1}$$

  gdzie $d_{\text{bert}} = 768$

- Reprezentacja statystyczna za pomocą metody TF-IDF:

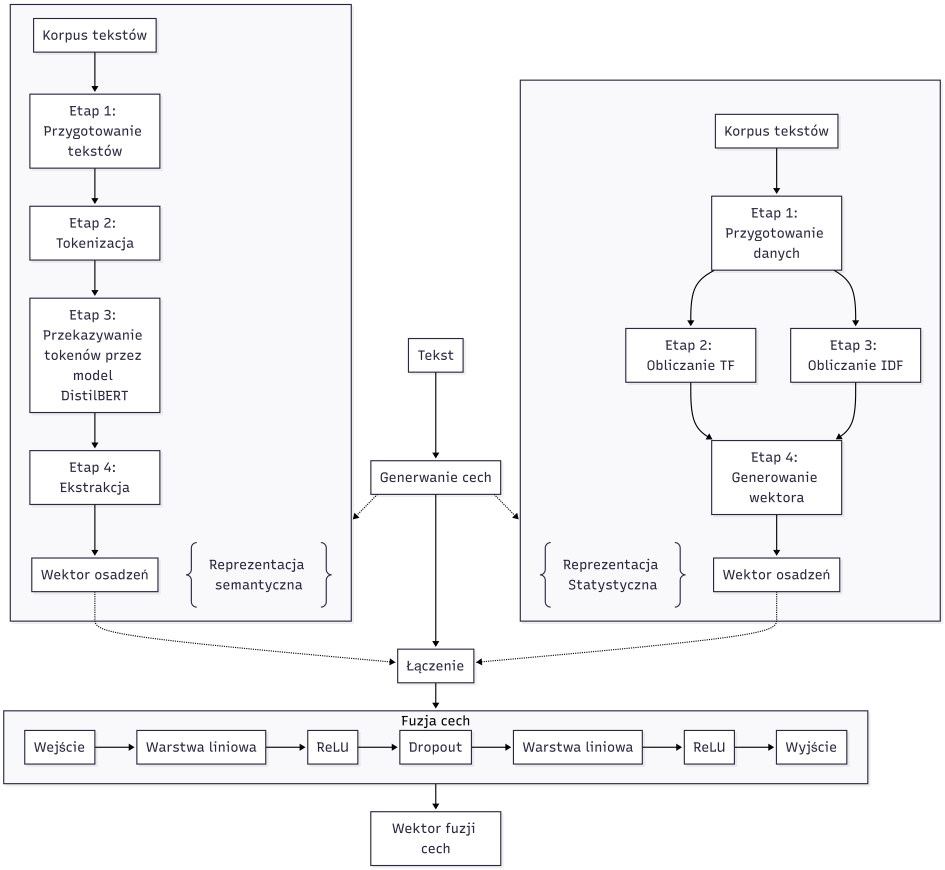
$$x_{\text{tfidf}} = \text{TFIDF}(T) \in \mathbb{R}^{d_{\text{tfidf}}} \tag{3.2}$$

gdzie wartość parametru d_{tfidf} została empirycznie ustalona na poziomie 500.

Końcowo oba wektory są łączone w jeden wspólny wektor zgodnie ze wzorem 3.3

$$x = [x_{\text{bert}}; x_{\text{tfidf}}] \in \mathbb{R}^d \tag{3.3}$$

gdzie $d = d_{\text{bert}} + d_{\text{tfidf}}$.



Rysunek 3.1: Architektura metody LFM

Tak utworzony wektor x jest następnie przekazywany do sieci neuronowej, której zadaniem jest utworzenie wektora fuzji cech.

3.1.1 Etapy generowania wektorów osadzeń

3.1.1.1 Reprezentacja uzyskana modelem DistilBERT

Model DistilBERT stanowiący zoptymalizowaną wersję modelu BERT cechuje się mniejszą liczbą parametrów oraz zwiększoną efektywnością obliczeniową przy jednoczesnym zachowaniu wysokiej jakości generowanych reprezentacji semantycznych [87].

Metoda ta posiada dwa kluczowe etapy przetwarzania tekstu. W etapie pierwszym wstępnie przetworzony tekst poddawany jest tokenizacji. W ramach niej na początku i na końcu tekstu dodawane są specjalne tokeny (`[CLS]` oraz `[SEP]`). Następnie poszczególne słowa przekształcane są w odpowiadające im identyfikatory z wykorzystaniem słownika osadzeń wyuczonych w fazie wstępnego uczenia modelu.

Tak przygotowany ciąg tokenów przekazywany jest do modelu DistilBERT odpowiadającego za wygenerowanie jego wektorowej reprezentacji. Reprezentacja długości wektora ma stałą wartość 768. Warto również wspomnieć, że pomimo możliwości przetwarzania do 512 tokenów wejściowych autor niniejszej rozprawy zdecydował się na ograniczenie ich liczby do 256. Rozwiązanie to podyktowane było potrzebą redukcji zużycia zasobów pamięciowych oraz skrócenia czasu trenowania modeli.

Kolejnym istotnym zagadnieniem jest konieczność zastosowania tokenizera kompatybilnego z modelem DistilBERT, co wynika z potrzeby zapewnienia spójności między procesem tokenizacji a strukturą modelu.

Proces ekstrakcji cech w tej metodzie realizowany jest z wykorzystaniem wstępnie wytrenowanego modelu DistilBERT, w dwóch wersjach językowych:

- angielskiej [88] — dla zbiorów danych w języku angielskim;
- polskiej [89] — dla zbiorów danych w języku polskim.

3.1.1.2 Reprezentacja uzyskana metodą TF-IDF

Reprezentacja TF-IDF ma na celu ocenę istotności terminów w kontekście całego zbioru danych. Wartość wskaźnika składa się z 2 elementów częstotliwości występowania słowa w tekście (ang. *Term Frequency*, TF) oraz rzadkości w całym zbiorze dokumentów (ang. *Inverse Document Frequency*, IDF).

Częstotliwość występowania słowa w tekście można wyliczyć ze wzoru 3.4. Przy czym otrzymany wynik otrzymuje wartości z przedziału [0,1], gdzie 0 oznacza brak wystąpień danego słowa w tekście, a 1 gdy występuje tylko dane słowo.

$$TF(x, y) = \frac{\text{liczba wystąpień słowa } x \text{ w tekście } y}{\text{całkowita liczba słów w dokumencie } y} \tag{3.4}$$

Rzadkość występowania słowa w dokumencie polega na ocenie jak niepowtarzalne, jest dane słowo, biorąc pod uwagę cały zbiór zdań w artykule. Wzór na obliczenie tej wartości został przedstawiony poniżej (wzór 3.5). Wysoka wartość tego wskaźnika oznacza, że dane słowo jest unikatowe w odniesieniu do całości artykułu, przez co powinno być istotniejsze pod kątem analizy tekstu.

$$IDF(x, y) = \log\left(\frac{\text{liczba zdań w artykule } y}{\text{liczba zdań zawierających słowo } x + 1}\right) \tag{3.5}$$

Końcowa wartość współczynnika TF-IDF jest wynikiem mnożenia wartości obu współczynników. Należy przy tym zauważyć, że im częściej dane słowo będzie pojawiało się w zdaniach, zachowując jednocześnie unikalność w innych fragmentach tekstu to wartość tego współczynnika, będzie wyższa (osiągane wartości są z przedziału [0,1]). Dzięki temu mamy możliwość koncentracji modelu na tylko istotnych słowach z pominięciem tych mało istotnych.

3.1.2 Mechanizm fuzji wektorów osadzeń (Fusion Encoder)

Zastosowana jako enkoder sieć neuronowa jest architekturą opartą na dwóch warstwach liniowych, funkcjach aktywacji ReLU (ang. *Rectified Linear Activation Function*), a także mechanizmie Droput. Schemat został przedstawiony w części FusionEncoder na rysunku 3.1

Pierwsza warstwa liniowa przekształca dane wejściowe x o wymiarze 1268 (768 wymiar z BERT i 500 z TF-IDF) na przestrzeń o wymiarze 256 określona wzorem:

$$h_1 = W_1 x + b_1 \tag{3.6}$$

gdzie:

- $W_1 \in \mathbb{R}^{256\times1268}$ - macierz wag,

- $b_1 \in \mathbb{R}^{256}$ - wektor przesunięć (bias),
- $x \in \mathbb{R}^{1268}$ - wektor wejściowy.

Kolejnym elementem jest funkcja aktywacji ReLU, której zadaniem jest wprowadzenie nieliniowości do modelu zastępując wartości ujemnie zerami. Wyrażona jest ona wzorem:

$$h_1^{\text{ReLU}} = \max(0, h_1) \tag{3.7}$$

W zaproponowanej sieci funkcja ta występuje dwukrotnie po pierwszej oraz po drugiej warstwie liniowej gdzie zamiast h_1 i h_1^{ReLU} jest odpowiednio h_2 i h_2^{ReLU}.

Kolejnym elementem wykorzystywanym wyłącznie na etapie uczenia sieci jest mechanizm Dropout, będący techniką regularyzacji polegającą na losowym wyłączaniu niektórych neuronów. Wartość po funkcji aktywacji h_1^{ReLU} jest mnożona przez losową maskę r, przyjmującą wartości 0 dla neuronów nieaktywnych i 1 dla neuronów aktywnych. W zaproponowanej metodzie współczynnik Dropout został ustalony na wartość 0.3.

Następnie zastosowano kolejną warstwę liniową, która przekształca dane z przestrzeni o wymiarze 256 na przestrzeń o wymiarze 64 określona wzorem:

$$h_2 = W_2 h_1^{\text{Dropout}} + b_2 \tag{3.8}$$

gdzie:

- $W_2 \in \mathbb{R}^{64\times256}$ - macierz wag,
- $b_2 \in \mathbb{R}^{64}$ - wektor przesunięć (bias).

3.1.3 Funkcja straty wykorzystana w procesie uczenia

Ze względu na częste występowanie problemu niezbalansowania klas w detekcji dezinformacji, w zaproponowanej metodzie (na etapie enkodera) została wykorzystana funkcja Focal Loss [90]. Funkcja ta pozwala na zwiększenie wagi próbek klasy mniejszościowej i równocześnie ogranicza wpływ klasy dominującej. Warto podkreślić, że wartości domyślne parametrów γ oraz α, występujące we wzorze, zostały zaproponowane i empirycznie zweryfikowane w tej samej publikacji [90].

$$FL(p_t) = -\alpha(1 - p_t)^{\gamma} \log(p_t) \tag{3.9}$$

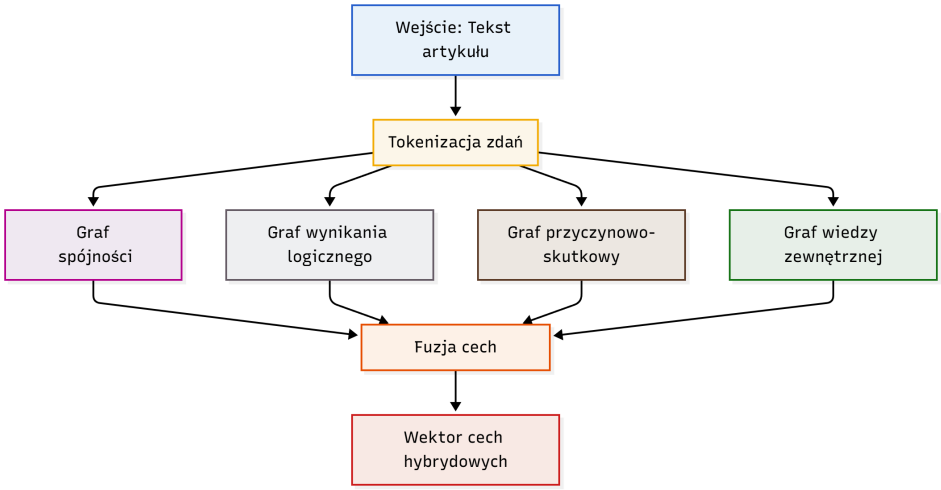
Gdzie:

- $p_t = \begin{cases} \hat{y}, & \text{jeśli } y = 1 \\ 1 - \hat{y}, & \text{jeśli } y = 0 \end{cases}$
- $\gamma > 0$ — parametr skupienia (domyślnie $\gamma = 2$),
- $\alpha \in (0, 1)$ — współczynnik ważenia klasy mniejszości (domyślnie $\alpha = 0.25$).

Podsumowując, zaproponowana metoda opiera się na dwuwarstwowej sieci neuronowej, zaprojektowanej do pracy z heterogenicznymi wektorami cech. Łączy ona zaawansowaną analizę semantyczną, realizowaną za pomocą modelu DistilBERT, z informacją statystyczną o częstości terminów, dostarczaną przez TF-IDF. Ponadto, dzięki zastosowaniu funkcji straty Focal Loss, model skutecznie radzi sobie zarówno z danymi zbalansowanymi, jak i silnie niezbalansowanymi, co zostało wykazane w dalszej części pracy.

3.2 Metoda GEM (Graph Embedding Method)

W ramach hybrydowego podejścia do ekstrakcji cech opracowano także metodę GEM (Graph Embedding Method). Metoda ta została zaprezentowana na rysunku 3.2.



Rysunek 3.2: Schemat działania metody GEM

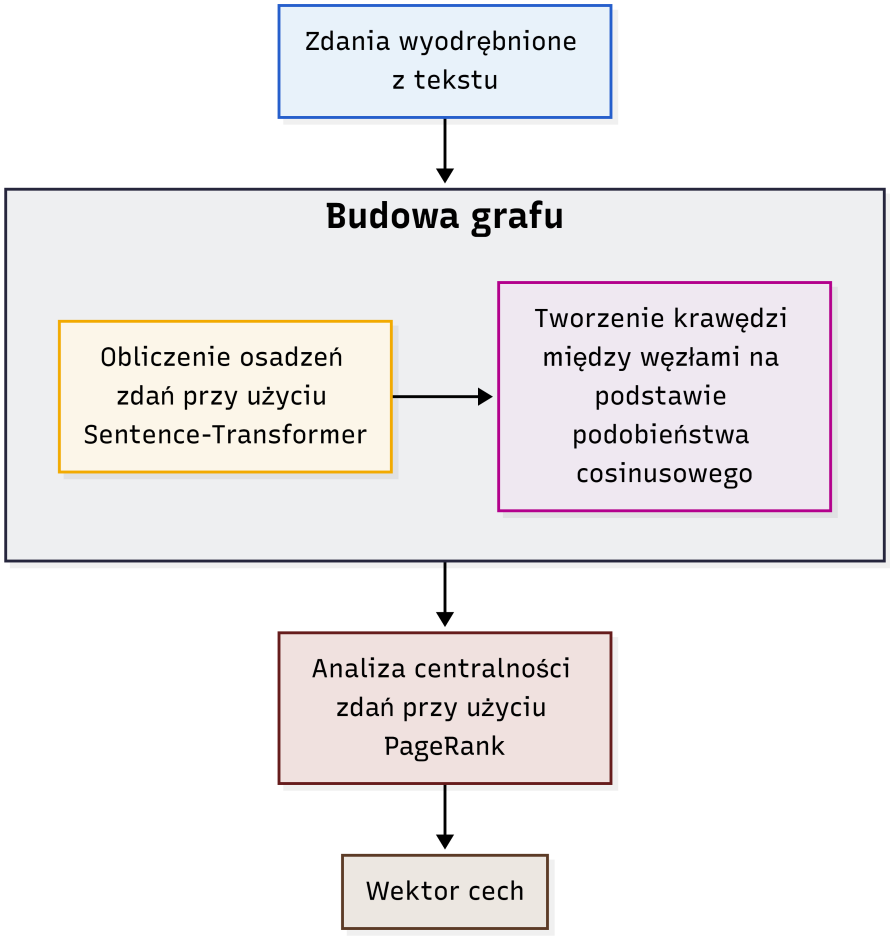
Schemat (rysunek 3.2) prezentuje kompleksowe rozwiązanie hybrydowego ekstraktora cech składającego się z czterech głównych grafów, a każdy z grafów jest odpowiedzialny za inną perspektywę analizy tekstu:

- graf spójności odpowiadający za spójność semantyczną tekstu;
- graf wynikania logicznego odpowiadający za logiczne zależności;
- graf przyczynowo-skutkowego odpowiadający za relacje przyczynowo-skutkowe;
- graf wiedzy zewnętrznej odpowiadający za powiązanie z wiedzą zewnętrzną.

Następnie wektory cech z poszczególnych grafów są łączone, tworząc hybrydowy wektor wykorzystywany do dalszej analizy.

3.2.1 Graf spójności tekstu

Pierwszy z zaprojektowanych w zaproponowanej metodzie grafów został przedstawiony na rysunku 3.3

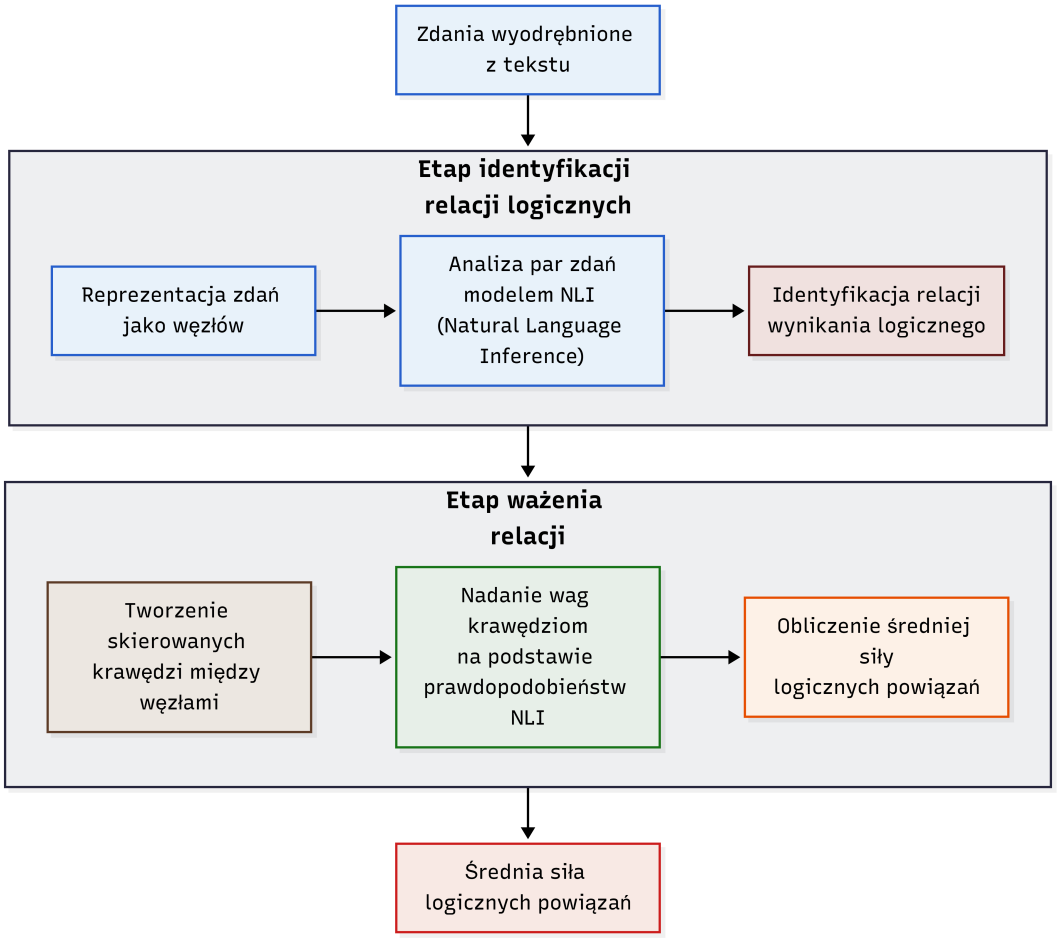


Rysunek 3.3: Schemat działania grafu spójności tekstu

W zaproponowanym module (grafie) każde zdanie występujące w tekście zostaje reprezentowane jako węzeł grafu, a krawędzie łączące węzły odzwierciedlają podobieństwo semantyczne między zdaniami. Obliczenia podobieństwa są dokonywane poprzez wyliczenie wektorów osadzeń przy użyciu modelu Sentence-Transformer [91]. Z kolei waga krawędzi w grafie odpowiada wartości podobieństwa cosinusowego pomiędzy osadzeniami. Dodatkowo w kolejnym etapie dokonywana jest analiza centralności zdań z wykorzystaniem algorytmu PageRank. Ocena ta pozwala na określenie, które fragmenty tekstu są najbardziej reprezentatywne w stosunku do całej treści. Metoda ta zwraca dwie wartości (cechy spójności oraz centralności).

3.2.2 Graf wynikania logicznego

Kolejnym modułem zaproponowanej metody jest graf wynikania logicznego. Graf ten modeluje logiczne zależności między zdaniami, umożliwiając ocenę, w jakim stopniu jedno zdanie wynika z innego. Schemat działania tego modułu przedstawiony został na rysunku 3.4



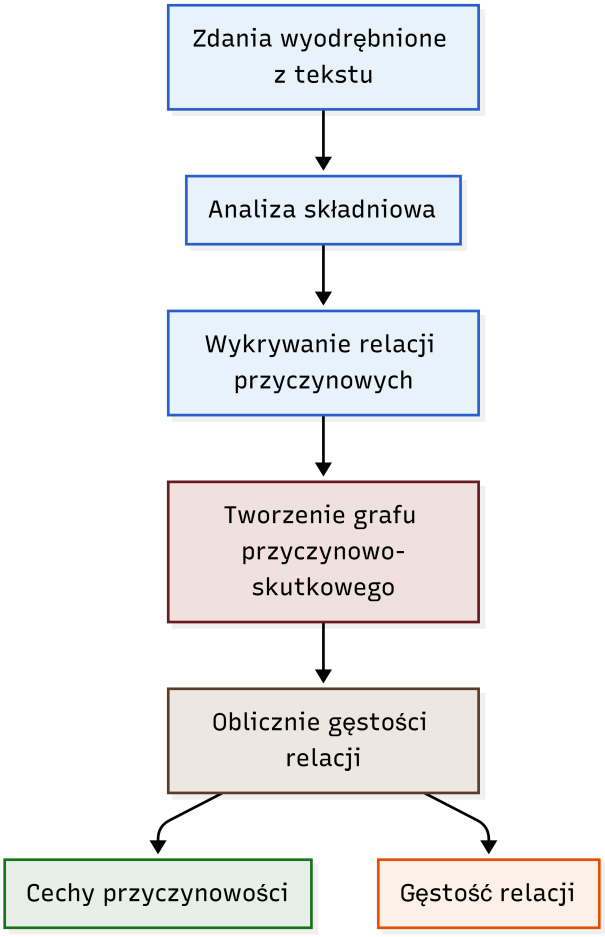
Rysunek 3.4: Schemat działania modułu wynikania logicznego

Każde zdanie jest reprezentowane przez węzeł, a skierowane krawędzie wskazują relacje potwierdzające logiczne wnioskowanie pomiędzy zdaniami. W celu identyfikacji tych relacji wykorzystany został model NLI (ang. *Natural Language Inference*). Dodatkowo w zaproponowanej metodzie został wykorzystany me-

chanizm wag krawędzi, który odpowiada wartości prawdopodobieństwa, z jakim model przewiduje klasę wynikania dla danej pary węzłów. Metoda ta zwraca informację o średniej sile logicznych powiązań między zdaniami wykorzystywanej jako jeden z parametrów na późniejszym etapie badań.

3.2.3 Graf przyczynowo-skutkowy

Kolejnym modułem (przedstawionym na rysunku 3.5) jest graf przyczynowo-skutkowy. Został on opracowany w celu wykrywania zależności przyczynowo-skutkowych w analizowanym tekście.

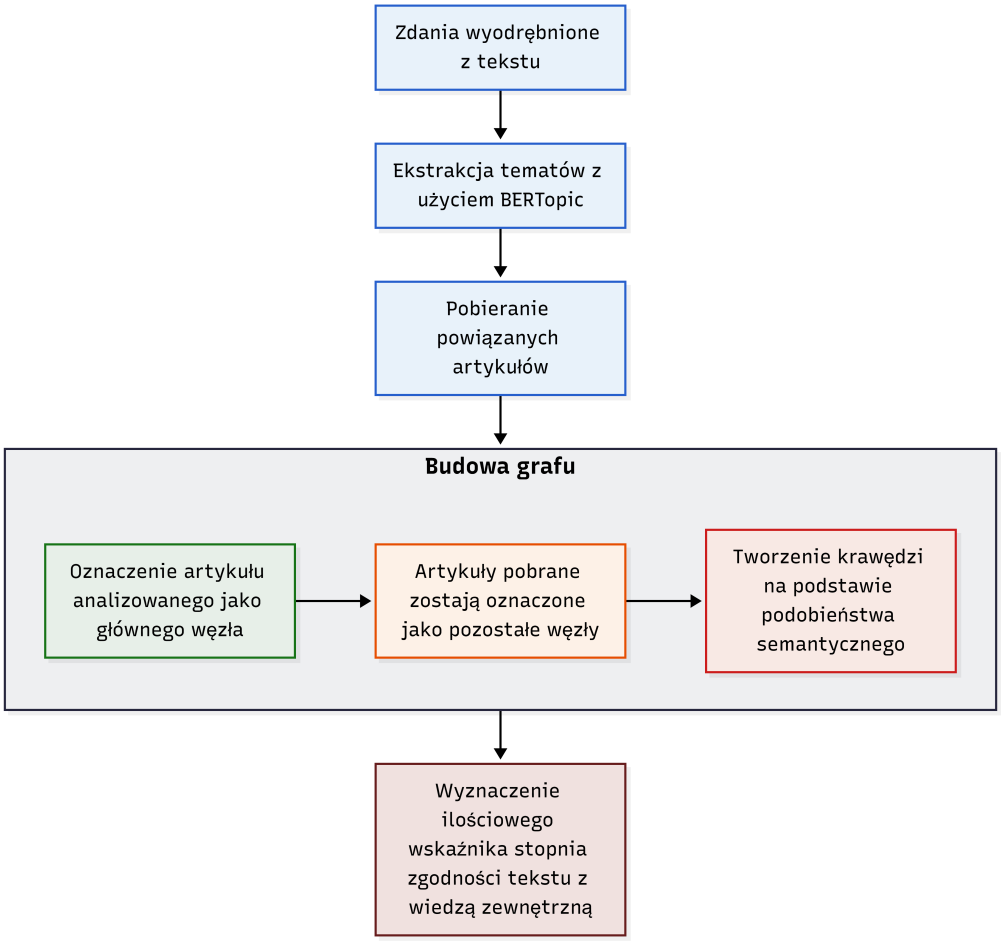


Rysunek 3.5: Schemat działania modułu przyczynowo-skutkowego

Skierowane krawędzie grafu odwzorowują relacje przyczynowo-skutkowe wykryte w tekście. Analiza grafu pozwala określić liczbę i gęstość relacji przyczynowo-skutkowych. Gęstość relacji oraz cechy przyczynowości są następnie zwracane jako parametry wykorzystywane w późniejszym etapie badań.

3.2.4 Graf wiedzy zewnętrznej

Ostatnim modułem w autorskiej metodzie jest graf wiedzy zewnętrznej. Został on przedstawiony na rysunku 3.6.



Rysunek 3.6: Schemat działania modułu wiedzy zewnętrznej

Graf wiedzy zewnętrznej umożliwia powiązanie analizowanego tekstu z in-

formacjami pochodzącymi z zewnętrznych źródeł np. wiadomości pobrane z kanału RSS Google News. Metoda ta dla zdań w artykule (przy pomocy modelu BERTopic [92] wspierany przez wielojęzyczny model językowy [91]) wykrywa automatycznie tematy, a następnie wybierany jest temat występujący najczęściej.

W kolejnym kroku dokonywane jest wyszukiwanie i pobieranie artykułów powiązanych tematycznie. Po pobraniu artykułów analizowany artykuł zostaje oznaczony jako główny węzeł grafu, natomiast znalezione artykuły stanowią pozostałe węzły w grafie. Krawędzie w grafie stanowią stopień podobieństwa semantycznego (obliczony na podstawie porównania wektorów osadzeń Sentence-BERT miarą podobieństwa cosinusowego). Na wyjściu zaproponowanego modułu otrzymujemy ilościowy wskaźnik stopnia zgodności tekstu z wiedzą zewnętrzną.

W celu uniknięcia ryzyka wycieku danych (ang. *data leakage*) zadbano o pełną separację danych między etapem uczenia a testowania. Moduł wiedzy zewnętrznej korzysta wyłącznie z informacji pozyskiwanych po momencie podziału zbioru danych, a graf wiedzy budowany jest niezależnie dla każdego artykułu z wykorzystaniem wyłącznie zewnętrznych źródeł (np. wiadomości z kanału RSS). Oznacza to, że w procesie ewaluacji nie występują żadne połączenia ani zależności pomiędzy dokumentami ze zbiorów treningowych i testowych.

3.2.5 Integracja i łączenie danych z wielu grafów

Proces integracji wyników analiz przeprowadzonych na podstawie opisanych wcześniej grafów stanowi ważny etap opracowanej autorskiej metody ekstrakcji cech. Polega on na syntezie cech wydobytych z czterech niezależnych struktur grafowych, z których każda odpowiada za odrębny poziom interpretacji warstwy tekstowej. Proces ten realizowany jest przez operację konkatenacji wektorów cząstkowych, co pozwala na zachowanie pełnej informacji strukturalnej pochodzącej z każdego z grafów, bez ryzyka utraty istotnych składowych.

Formalnie proces tworzenia hybrydowego wektora cech V_H można zapisać jako:

$$V_H = V_{sem} \oplus V_{log} \oplus V_{caus} \oplus V_{ctx} \tag{3.10}$$

gdzie:

- V_{sem} - wektor cech reprezentujący aspekty semantyczne (z grafu spójności),

- V_{log} - wektor cech reprezentujący strukturę logiczną (z grafu wynikania logicznego),
- V_{caus} - wektor cech reprezentujący relacje przyczynowo-skutkowe (z grafu przyczynowo-skutkowego),
- V_{ctx} - wektor cech reprezentujący powiązanie z wiedzą zewnętrzną (z grafu wiedzy zewnętrznej).

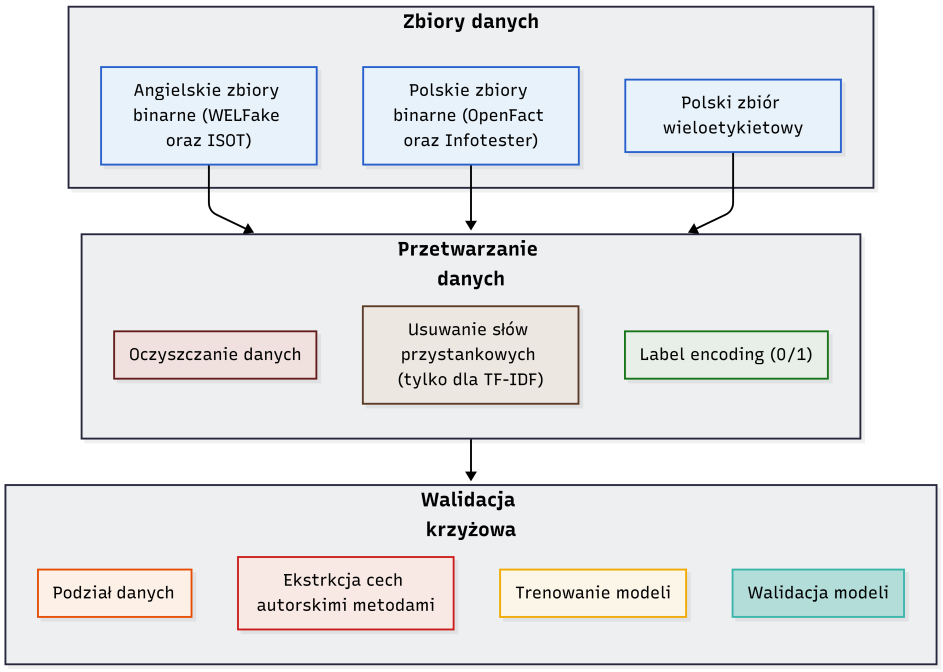
Wynikowy wektor (V_H) stanowi wielowymiarową, heterogeniczną reprezentację analizowanego tekstu. Takie podejście pozwala na uchwycenie w procesie detekcji różnorodnych zależności wewnątrz tekstu, zapewniając modelowi komplementarny zestaw danych wejściowych.

4. Metodyka badań i procedura eksperymentalna

W niniejszym rozdziale przedstawiono szczegółową metodykę przeprowadzonych badań oraz procedurę eksperymentalną.

4.1 Procedura eksperymentalna

W ramach rozprawy doktorskiej przeprowadzono badania zaproponowanych metod LFM, GEM oraz metody referencyjnej, pełniącej rolę standardu porównawczego dla oceny skuteczności zaproponowanych metod. Schemat przeprowadzonych badań przedstawiono na rysunku 4.1.



Rysunek 4.1: Schemat procedury przeprowadzonych eksperymentów

38:14606610

W pierwszym kroku ze zbiorów danych opisanych w podrozdziale 4.2 zostały pobrane dane tekstowe oraz etykiety. Dane te (w drugim kroku) przeszły proces wstępnego przetwarzania, który szczegółowo został opisany w podrozdziale 4.3.

W kolejnym etapie przetwarzania zbiór danych został podzielony na zbiory treningowe i testowe z wykorzystaniem procedury walidacji krzyżowej (ang. cross-validation), umożliwiającej obiektywną ocenę zdolności generalizacyjnych modelu poprzez wielokrotne uczenie i testowanie na różnych podzbiorach danych.

Następnym etapem przeprowadzonych badań była ekstrakcja cech tekstu, realizowana zarówno metodą referencyjną, jak i z wykorzystaniem zaproponowanych metod LFM oraz GEM. W ramach niniejszej rozprawy, w celu odniesienia uzyskanych wyników metod hybrydowych (szczegółowo opisanych w rozdziale 3) do wyników metody referencyjnej, wybrano podejście bazujące na modelu DistilBERT. Metoda ta, opisana w podrozdziale 3.1.1.1, pełniła rolę ekstraktora cech, natomiast proces klasyfikacji realizowano z użyciem algorytmów takich jak drzewo decyzyjne, maszyna wektorów nośnych (ang. support vector machine, svm), perceptron wielowarstwowy (ang. multi-layer perceptron, mlp) i ekstremalne wzmacnianie gradientowe (ang. extreme gradient boosting, xgboost). Uzyskane w ten sposób reprezentacje wektorowe tekstów zostały następnie przekazane do wspomnianych klasyfikatorów w celu dalszej analizy.

Końcowym etapem prac badawczych była ewaluacja uzyskanych modeli przy użyciu miar jakości opisanych w rozdziale 4.4. Dla każdego ze zbiorów danych przeprowadzono łącznie dwanaście eksperymentów, z wyjątkiem zbioru pochodzącego z projektu SWAROG, w przypadku którego liczba eksperymentów była większa ze względu na jego specyficzną strukturę.

Spośród zaproponowanych autorskich metod ekstrakcji cech najbardziej skuteczna okazała się metoda GEM (Graph Embedding Method). Uzyskała ona wyniki o około 14% wyższe niż metoda referencyjna. Wyniki wszystkich przeprowadzonych eksperymentów zostały szczegółowo zaprezentowane w rozdziale 5.

4.2 Zbiory danych

W ramach rozprawy doktorskiej wykorzystano pięć zbiorów danych do klasyfikacji dezinformacji, z czego dwa zbiory obejmowały dane w języku angielskim, a trzy w języku polskim. Dodatkowo, ze względu na uczestnictwo autora w projekcie SWAROG (System Wykrywania Dezinformacji Metodami Sztucznej Inteligencji w ramach programu Infostrateg) wykorzystano zbiór opracowany w ramach tego

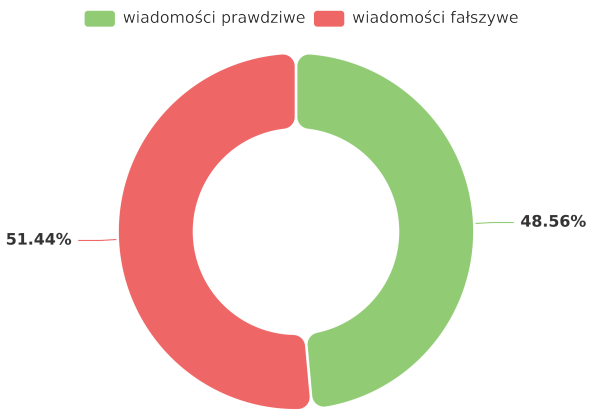
projektu.

4.2.1 Zbiory anglojęzyczne

4.2.1.1 Word Embedding over Linguistic Features for Fake News Detection (WELFake)

Pierwszy zbiór danych wykorzystany w ramach rozprawy doktorskiej - Word Embedding over Linguistic Features for Fake News Detection (WELFake) [93] stanowi korpus tekstowy w języku angielskim. Obejmuje 72 134 artykuły sklasyfikowane binarnie i został opracowany poprzez integrację informacji pochodzących z różnych źródeł.

W zbiorze tym każdy z artykułów został opatrzony odpowiednią etykietą (ang. *label*) jednoznacznie wskazującą jego charakter. Wartość 1 przypisano artykułom prawdziwym (ang. *real news*), natomiast 0 przypisano artykułom zawierającym treści dezinformacyjne, mijające się z prawdą (ang. *fake news*).



Rysunek 4.2: Rozkład liczności etykiet w zbiorze danych WELFake

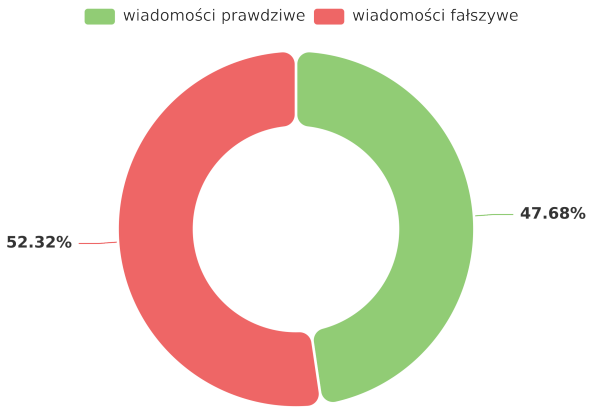
Rozkład klas w zbiorze danych został przedstawiony na wykresie 4.2. Jak można zauważyć, 35 028 artykułów zostało oznaczonych jako artykuły zawierające prawdziwe informacje oraz 37 106 jako artykuły zawierające fałszywe informacje. Jak można zauważyć, zbiór ten posiada względnie zrównoważony rozkład klas, a jego szczegółowa struktura zbioru danych została zaprezentowana w tabeli 4.1.

Tabela 4.1: Struktura i format zbioru danych WELFake

id	Pole zawierające indywidualny identyfikator artykułu. Numeracja zaczyna się od 0.
title	Jest to kolumna zawierająca tytuł artykułu. Średnia długość tekstu w tej kolumnie wynosi 76 znaków, a średnia ilość słów wynosi 12. Dodatkowo ta kolumna zawiera 558 pustych wierszy.
text	Jest to kolumna zawierająca treść artykułu. Średnia długość tekstu w tej kolumnie wynosi 3 268 znaków, a średnia ilość słów wynosi 540. Dodatkowo ta kolumna zawiera 39 pustych wierszy.
label	Jest to kolumna zawierająca etykietę binarną artykułu - 0 dla wartości fałszywych i 1 dla prawdziwych. kolumna ta nie zawiera pustych wierszy.

4.2.1.2 ISOT Fake News detection dataset

Kolejny wykorzystany w rozprawie doktorskiej zbiór danych - ISOT Fake News detection dataset [94] stanowi binarny korpus tekstów w języku angielskim. Zbiór ten zawiera łącznie 44 919 artykułów, a jego szczegółowy rozkład przedstawiono na wykresie 4.3.



Rysunek 4.3: Rozkład liczności etykiet w zbiorze danych ISOT

Tabela 4.2: Struktura i format zbioru danych ISOT

title	Jest to kolumna zawierająca tytuł artykułu. Średnia długość tekstu w tej kolumnie wynosi 80 znaków, a średnia ilość słów wynosi 12.
text	Jest to kolumna zawierająca treść artykułu. Średnia długość tekstu w tej kolumnie wynosi 2 469 znaków, a średnia ilość słów wynosi 405.
subject	Jest to kolumna zawierająca tematykę artykułu. Artykuły zostały podzielone na 8 kategorii - politics news, world news, news, politics, left-news, government news, us news, middle-east.
date	Jest to kolumna zawierająca dane o czasie publikacji artykułu. Dane prawdziwe były zbierane w przedziale od 13 stycznia 2016 do 31 grudnia 2017, a dane fałszywe od 31 marca 2015 do 19 lutego 2018.

Jak można zauważyć zbiór ten podobnie jak wcześniejszy opisany w rozdziale 4.2.1.1 posiada względnie zrównoważony rozkład klas, gdzie 21 417 artykułów zostało oznaczonych jako prawdziwe informacje oraz 23 502 jako artykuły zawierające fałszywe informacje. Szczegółowa struktura zbioru danych została zaprezentowana w tabeli 4.2. W zaprezentowanej strukturze nie występuje pole etykiety (ang. *label*), gdyż treści fałszywe i prawdziwe znajdowały się w dwóch oddzielnych plikach. Natomiast autor pracy zdecydował się na dodanie takiego pola, w celu utworzenia spójnego zbioru danych do dalszych analiz.

4.2.2 Polskie zbiory binarne

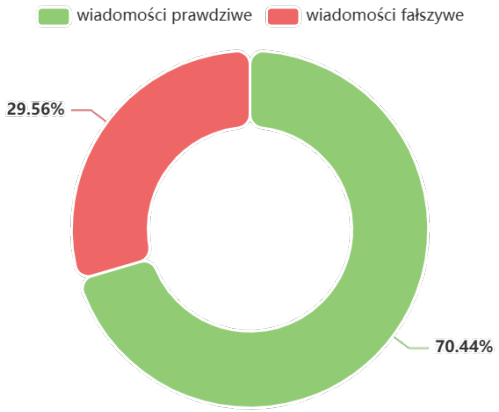
4.2.2.1 OpenFact

Zbiór OpenFact [95] jest binarnym polskim zbiorem danych tekstowych utworzonym przez Uniwersytet Ekonomiczny w Poznaniu w ramach programu Infostrateg Narodowego Centrum Badań i Rozwoju. Zbiór ten zawiera 38 068 artykułów (a po usunięciu duplikatów 36 770) i posiada dwie kolumny url oraz label. Pierwsza z nich wskazuje na adresy stron internetowych, skąd pobrane były informacje, z kolei druga kolumna zawiera etykiety tak, lub nie, w zależności od tego, czy artykuł jest prawdziwy, czy też stanowi dezinformację. Warto również wspomnieć, że zbiór ten jest najbardziej niezbalansowanym zbiorem wykorzystywanym w niniejszej rozprawie. Liczy on 32 405 artykułów oznaczonych jako „Nie” i tylko 3 472 artykuły z etykietą „Tak".

42:10369360

4.2.2.2 Infotester

Zbiór danych Infotester [96] stanowi polskojęzyczny korpus opracowany w ramach programu INFOSTRATEG, realizowanego przy wsparciu Narodowego Centrum Badań i Rozwoju. Celem projektu było stworzenie narzędzi wspierających wykrywanie dezinformacji oraz ocenę wiarygodności treści publikowanych w internecie.



Rysunek 4.4: Rozkład liczności etykiet w zbiorze danych Infotester

Zbiór obejmuje 15 608 artykułów tekstowych, które zostały poddane ręcznej ocenie przez wykwalifikowanych annotatorów. Dane mają charakter binarny, co oznacza, że każdemu artykułowi przypisano etykietę wskazującą na jego prawdziwość lub fałszywość. Warto jednak zaznaczyć, że w zbiorze występują 4 etykiety, ale dwie z nich oznaczone jako trudno stwierdzić (ang. *hard to say*) - 105 próbek i mylna informacja (ang.*misinformation*) - 38 próbek zostały usunięte ze zbioru. Struktura udostępnionych danych jest szczegółowa i zawiera m.in. następujące pola: unikalny identyfikator rekordu, identyfikator artykułu, kategoria tematyczna, adres URL źródłowej publikacji, oceny dwóch niezależnych recenzentów wraz z czasem ich dokonania, link umożliwiający pobranie treści artykułu, a także ocenę końcową wraz z czasem jej nadania.

4.2.3 Polski zbiór wieloetykietowy

Zbiór SWAROG [97] jest wieloetykietowym polskim zbiorem danych tekstowych. W odróżnieniu od poprzednich zbiorów danych nie zawiera jednoznacznej

etykiety, a odpowiedzi na 13 pytań. Pytania były podzielone na 3 grupy: weryfikacyjne, manipulacyjne i metafizyczne.

Pierwsza grupa pytań dotyczyła czynników weryfikacyjnych, obejmowała pytania, które wymagały dodatkowej aktywności ze strony annotatora w formie weryfikacji treści przedstawionych w artykule w innych źródłach. Pytania z tej grupy:

- Czy istnieje co najmniej jedno wiarygodne źródło, które potwierdza wszystkie informacje zawarte w treści? (Q1)
- Czy większość podanych informacji jest potwierdzona przez wiarygodne źródła? (Q2)
- Czy żadna z informacji nie jest potwierdzona przez wiarygodne źródła? (Q3)
- Czy stwierdzenie odnosi się do aktualnych danych? (Q4)

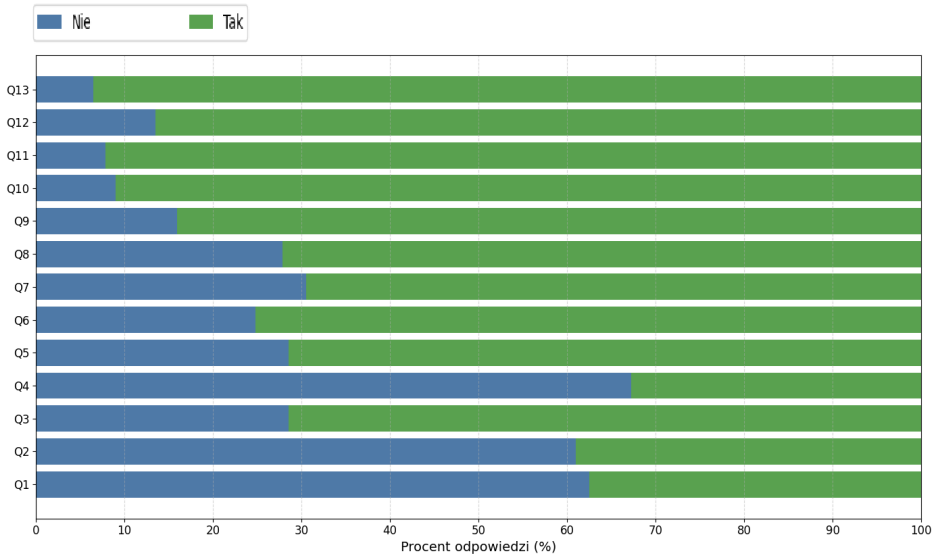
Druga grupa pytań dotyczyła czynników manipulacyjnych, miała na celu zbadanie, czy autor opublikowanej treści celowo narzucił czytelnikowi określony punkt widzenia lub wprowadził odbiorcę w błąd. Pytania z tej grupy:

- Czy do właściwego zrozumienia treści wymagane są dodatkowe informacje? (Q5)
- Czy treść zawiera nieścisłości? (Q6)
- Czy stwierdzenie zawiera fragmenty wyrwane z kontekstu? (Q7)
- Czy autor stwierdzenia stosuje wybiórcze przedstawianie faktów? (Q8)
- Czy autor stwierdzenia próbuje wprowadzić czytelnika w błąd? (Q9)

Ostatnią grupę pytań stanowią czynniki metafizyczne. Celem ich stosowania jest wykrycie czy w badanym tekście oddziaływano na emocje czytelnika, uwzględniając jego poglądy. Pytania z tej grupy:

- Czy treść ma charakter satyryczny? (Q10)
- Czy autor przyznaje, że przedstawione fakty są zmyślone? (Q11)

- Czy stwierdzenie zawiera obietnice polityczne? (Q12)
- Czy stwierdzenie zawiera treści religijne? (Q13)



Rysunek 4.5: Rozkład etykiet w zbiorze danych SWAROG

Na wykresie 4.5 przedstawiony został rozkład klas dla poszczególnych pytań opisanych powyżej. Jak można zauważyć, zbiór ten charakteryzuje się znaczną nierównowagą klas. W kategorii pytań dotyczących czynników weryfikacyjnych większość odpowiedzi została oznaczona jako negatywna, a w pozostałych dwóch grupach klasę większościową stanowią odpowiedzi twierdzące.

4.3 Wstępne przetwarzanie danych

Proces oczyszczania danych stanowił przygotowanie wstępne dla wszystkich metod ekstrakcji cech opisanych w rozprawie. Etap ten był podzielony na dwie części. Pierwsza z nich odpowiadała za eliminację pustych rekordów. Dodatkowo w drugim wariancie tego etapu zdecydowano się usunąć znaczniki HTML, a w przypadku stosowania metody statystycznej TF-IDF dodatkowo znaki specjalne, nawiasy oraz liczby tak, aby pozostały jedynie słowa, oraz interpunkcja w postaci przecinków i kropek. Działania te miały na celu przygotowanie tekstu do dalszej analizy.

Kolejny krok polegał na usunięciu słów przystankowych (ang. *stop words*), gdyż ich występowanie nie wpływa na zrozumienie tekstu, a jednocześnie ogranicza koncentrację na słowach mających większe znaczenie i będących bardziej istotnymi dla zrozumienia sensu artykułu. Co istotne na tym etapie dokonano również zamiany wartości true i false na wartości 0 i 1, wykorzystując label encoding.

4.4 Ocena skuteczności klasyfikacji

Ocena skuteczności klasyfikacji jest jednym z istotnych zagadnień w kontekście określenia działania modelu. Macierz pomyłek (ang. *confusion matrix*) jest jednym z podstawowych narzędzi do oceny jakości klasyfikatorów. Macierz ta umożliwia szczegółową analizę wyników modelu, poprzez zestawienie przewidywanych etykiet z rzeczywistymi etykietami przypisanymi do danych testowych. Dla problemu klasyfikacji binarnej ma ona postać macierzy o wymiarze 2x2, gdzie wyróżniamy cztery kategorie:

- True Positive (TP) jest to liczba przypadków poprawnie zaklasyfikowanych pozytywnie;
- True Negative (TN) jest to liczba przypadków poprawnie zaklasyfikowanych jako negatywne;
- False Positive (FP) jest to liczba przypadków błędnie zaklasyfikowanych jako pozytywne;
- False Negative (FN) jest to liczba przypadków błędnie zaklasyfikowanych jako negatywne.

Na podstawie wartości z macierzy pomyłek wyznacza się różne miary skuteczności klasyfikatora takie jak dokładność, precyzja, czułość czy miara F_1. Na etapie walidacji krzyżowej w pracy wykorzystano kilka z nich.

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

Precyzja (ang. *Precision*) mierzy, jaki odsetek przypadków zaklasyfikowanych jako pozytywne jest rzeczywiście pozytywny. Innymi słowy, z wszystkich przypadków, które model uznał za pozytywne, precyzja wskazuje, ile z nich faktycznie było poprawnie zaklasyfikowanych. Jest to ważna miara, gdy chcemy zminimalizować błąd pierwszego rodzaju.

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

Czułość (ang. *Recall*) mierzy, jaki odsetek rzeczywiście pozytywnych przypadków został poprawnie zaklasyfikowany przez model. Określa, jak dobrze model wychwytuje prawdziwe pozytywne przypadki. Jest to istotna metryka, gdy zależy nam na tym, by nie przeoczyć żadnych pozytywnych przypadków, czyli zminimalizować błąd drugiego rodzaju.

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4.3}$$

Metryka F_1 mierzy harmoniczną średnią między precyzją a czułością, łącząc obie te miary w jedną wartość. Określa, jak dobrze model balansuje między wykrywaniem prawdziwych pozytywnych przypadków (czułość) a minimalizowaniem liczby błędnych alarmów (precyzja). Jest to istotna metryka, gdy chcemy uzyskać równowagę między tym, jak dobrze model wykrywa pozytywne przypadki, a tym, jak rzadko popełnia błędy w postaci fałszywych alarmów.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.4}$$

Dokładność (ang. *Accuracy*) to odsetek wszystkich poprawnie zaklasyfikowanych przypadków (zarówno pozytywnych, jak i negatywnych) względem wszystkich przypadków w zbiorze danych.

$$Balanced\ Accuracy = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right) \tag{4.5}$$

Zbalansowana dokładność (ang. *Balanced Accuracy*) jest definiowana jako średnia arytmetyczna czułości oraz specyficzności (ang. *specificity*). Miara ta znajduje zastosowanie w analizie zbiorów o niezbalansowanym rozkładzie klas, gdyż w przeciwieństwie do standardowej dokładności (Accuracy), niweluje wpływ dominacji klasy liczniejszej na wynik końcowy.

5. Wyniki przeprowadzonych badań

W niniejszym rozdziale przedstawiono wyniki badań dotyczących oceny zaproponowanych przez autora hybrydowych metod ekstrakcji cech. Eksperymenty skupiają się na trzech komplementarnych obszarach:

- porównanie proponowanych metod względem metod referencyjnych (sekcja 5.1, 5.2, 5.3);
- porównanie proponowanych rozwiązań względem innych metod opisanych w literaturze (sekcja 5.4);
- ocena wpływu komponentów proponowanych metod na otrzymywane wyniki (sekcja 5.5).

Przy porównaniu względem metod referencyjnych (obszar pierwszy) zastosowano cztery wybrane modele klasyfikacyjne uczenia maszynowego (XGBoost, SVM, MLP oraz drzewa decyzyjne). Wyniki uzyskane przez modele zostały ocenione przy użyciu różnych metryk, takich jak dokładność (ang. *accuracy*), zbalansowana dokładność (ang. *balanced accuracy*), miara F_1-*score*, precyzja (ang. *precision*) oraz czułość (ang. *recall*). Ponadto dla zapewnienia przejrzystości prezentacji, eksperymenty podzielono na trzy grupy odpowiadające różnym typom zbiorów danych oraz zadaniom klasyfikacyjnym. W każdej grupie dokonano porównania trzech podejść: metody bazowej oraz dwóch autorskich metod hybrydowych. Eksperymenty zostały zorganizowane w następujący sposób:

- Zbiory angielskie binarne – proces detekcji binarnej przeprowadzony na danych w języku angielskim, stanowiący punkt odniesienia dla dalszych analiz.
- Zbiory polskie binarne – analogiczne eksperymenty wykonane na danych w języku polskim, pozwalające ocenić skuteczność metod w zadaniach dwuklasowych w warunkach języka o mniejszych zasobach niż w przypadku języka angielskiego.
- Zbiór polski wieloetykietowy – scenariusz bardziej złożony, w którym dane opatrzone są wieloma etykietami. W celu zachowania spójności analizy, dla każdej etykiety zdefiniowano osobny problem detekcji binarnej.

Przy porównaniu proponowanych rozwiązań z metodami opisanymi w literaturze (obszar drugi), skupiono się na ocenie ich konkurencyjności względem podejść opartych na dużych modelach językowych (np. GPT-4).

W obszarze trzecim przeprowadzono analizę ablacyjną (ang. *ablation study*), mającą na celu ocenę wpływu poszczególnych komponentów modelu na końcową skuteczność detekcji dezinformacji.

5.1 Binarne zbiory anglojęzyczne

W poniższym podrozdziale zostały przedstawione i szerzej omówione wyniki uzyskane w trakcie przeprowadzonych badań dla zaproponowanych rozwiązań.

5.1.1 Word Embedding over Linguistic Features for Fake News Detection (WELFake)

Jednym ze zbiorów wykorzystanych w niniejszej rozprawie jest zbiór WELFake. Zbiór ten, jak wspomniano we wcześniejszej części pracy, jest wyraźnie zbalansowany. Rezultaty uzyskane dla metody referencyjnej zostały przedstawione w Tabeli 5.1. Jedną z metryk jest dokładność (ang. *accuracy*). Miara ta przedstawia stosunek poprawnie sklasyfikowanych przykładów do ogólnej liczby przykładów. W tej metryce najwyższe rezultaty otrzymał model MLP z wynikiem 85.25%. Dla metody referencyjnej znacząco od wyników odbiega metoda wykorzystująca drzewa decyzyjne, która z wynikiem tylko 72.04% wykazuje najgorszą efektywność spośród zaproponowanych klasyfikatorów.

Tabela 5.1: Tabela z wynikami eksperymentów metody referencyjnej na zbiorze WELFake

Metryka	**XGBOOST**	**SVM**	**DECISION TREE**	**MLP**
Accuracy	0.8139	0.8086	0.7204	0.8525
Balanced Accuracy	0.8072	0.8160	0.7133	0.8517
F_1 (0)	0.8158	0.8235	0.7148	0.8532
F_1 (1)	0.8371	0.8149	0.7195	0.8589
Precision (0)	0.8375	0.8205	0.7453	0.8608
Precision (1)	0.8210	0.8325	0.7020	0.8353
Recall (0)	0.8104	0.8140	0.6965	0.8282
Recall (1)	0.8333	0.8240	0.7557	0.8497

Kolejną miarą przedstawioną w Tabeli 5.1 jest zbalansowana dokładność (ang. *balanced accuracy*), która jest szczególnie istotną miarą w przypadku nierównomiernie rozłożonych klas (co w przypadku tego zbioru danych nie było kluczowe ze względu na dobre zbalansowanie zbioru). W metryce tej podobnie jak w dokładności najlepsze rezultaty otrzymał model MLP, a najgorsze wyniki uzyskano dla modelu wykorzystującego drzewa decyzyjne.

Miara F_1 umożliwia ocenę modelu dla poszczególnych klas. Dla klasy 0 (oznaczonej jako wiadomości fałszywe) model MLP uzyskał wynik 85.32%, a dla klasy 1 (oznaczonej jako wiadomości prawdziwe) osiągnął 85.89%. Model drzew decyzyjnych osiągnął gorsze wyniki, szczególnie dla klasy 1, gdzie miara ta wyniosła jedynie 71.95%.

Tabela 5.2: Tabela z wynikami eksperymentów metody LFM na zbiorze WELFake

Metryka	**XGBOOST**	**SVM**	**DECISION TREE**	**MLP**
Accuracy	0.9137	0.9381	0.8254	0.9304
Balanced Accuracy	0.9379	0.9247	0.9302	0.9302
F_1 (0)	0.9101	0.9359	0.8165	0.9280
F_1 (1)	0.9170	0.9401	0.8335	0.9326
Precision (0)	0.9210	0.9414	0.8343	0.9333
Precision (1)	0.9071	0.9351	0.8177	0.9279
Recall (0)	0.8995	0.9305	0.7994	0.9228
Recall (1)	0.9271	0.9453	0.8499	0.9375

Analizując wyniki uzyskane przez pierwszą z zaproponowanych metod hybrydowych (LFM) przedstawione w Tabeli 5.2 można zauważyć, że dla miary dokładności najwyższe rezultaty otrzymał model SVM z wynikiem 93.81%. Dla metody tej znacząco odbiega metoda bazująca na drzewach decyzyjnych, która z wynikiem 82.54% wykazuje najgorszą efektywność spośród zaproponowanych klasyfikatorów. Kolejną miarą przedstawioną w Tabeli 5.2 jest zbalansowana dokładność. W metryce tej podobnie jak w dokładności, najwyższe rezultaty otrzymał model SVM, a najniższe model wykorzystujący drzewa decyzyjne.

Miara F_1 pozwala uzyskać ocenę modelu dla poszczególnych klas. Dla klasy 0 (oznaczonej jako wiadomości fałszywe) model SVM uzyskał wynik 93.59%, a dla klasy 1 (oznaczonej jako wiadomości prawdziwe) osiągnął 94.01%, Model drzew decyzyjnych osiągnął gorsze wyniki, szczególnie dla klasy 0, gdzie miara ta wyniosła jedynie 81.65%.

Analizując wyniki uzyskane przez drugą z zaproponowanych metod hybry-

Tabela 5.3: Tabela z wynikami eksperymentów dla metody GEM na zbiorze WELFake

Metryka	XGBOOST	SVM	DECISION TREE	MLP
Accuracy	0.9189	0.9387	0.7869	0.9454
Balanced Accuracy	0.9187	0.9386	0.7861	0.9453
F_1 (0)	0.9162	0.9368	0.7749	0.9436
F_1 (1)	0.9215	0.9404	0.7978	0.9472
Precision (0)	0.9208	0.9375	0.7962	0.9486
Precision (1)	0.9173	0.9399	0.7791	0.9426
Recall (0)	0.9116	0.9363	0.7547	0.9386
Recall (1)	0.9259	0.9410	0.8175	0.9519

dowych (GEM) przedstawione w Tabeli 5.3 można zauważyć, że dla mary dokładności najwyższe rezultaty otrzymał model MLP z wynikiem 94.54%. Dla metody tej gorszymi wynikami charakteryzuje się metoda wykorzystująca drzewa decyzyjne, która z wynikiem tylko 78.69% wykazuje najgorszą efektywność spośród zaproponowanych klasyfikatorów. Kolejną miarą przedstawioną w Tabeli 5.3 jest zbalansowana dokładność. W metryce tej podobnie jak w dokładności, najlepsze rezultaty otrzymał model MLP, a najgorsze drzewa decyzyjne.

Miara F_1 pozwala uzyskać ocenę modelu dla poszczególnych klas. Dla klasy 0 (oznaczonej jako wiadomości fałszywe) model MLP uzyskał wynik 94.36%, a dla klasy 1 (oznaczonej jako wiadomości prawdziwe) osiągnął 94.72%. Model drzew decyzyjnych osiągnął znacznie gorsze wyniki, szczególnie dla klasy 0, gdzie miara ta wyniosła jedynie 77.49%.

Dodatkowo przeprowadzone zostały testy istotności statystycznej na podstawie wartości zbalansowanej dokładności uzyskanej w 10-krotnej walidacji krzyżowej i klasyfikatora XGBoost. Test Friedmana wykazał istotne statystycznie różnice pomiędzy analizowanymi metodami ($\chi^2 = 18.73$, $p = 0.0001$). W celu identyfikacji par metod różniących się istotnie zastosowano testy post-hoc Wilcoxona z korekcją Holma-Bonferroniego. Analiza wykazała istotne statystycznie różnice pomiędzy wszystkimi parami metod ($p < 0{,}05$). Najwyższą średnią wartość zbalansowanej dokładności uzyskała metoda LFM (0.9379 ± 0.0241), która okazała się istotnie lepsza od metody GEM (0.9187 ± 0.0076) oraz metody referencyjnej ($0{,}8072 \pm 0.0369$). Dodatkowo, na podstawie rang Friedmana, metoda LFM uzyskała najniższą (najlepszą) średnią rangę równą 1.18, metoda GEM rangę 1.82, natomiast metoda referencyjna rangę 3.00.

Na podstawie uzyskanych wyników można stwierdzić, że zastosowanie metod hybrydowych w przypadku tego zbioru danych przyniosło rzeczywistą oraz istotną statystycznie poprawę wyników detekcji dezinformacji.

5.1.2 ISOT Fake News detection dataset

Kolejnym anglojęzycznym zbiorem binarnym wykorzystanym w niniejszej pracy jest zbiór ISOT (ISOT Fake News detection dataset). Zbiór ten, zgodnie z wcześniejszymi opisami, jest zbiorem zbalansowanym, podobnie jak zbiór WELFake. Wyniki eksperymentów uzyskane przez metodę referencyjną zostały przedstawione w Tabeli 5.4.

Tabela 5.4: Tabela z wynikami eksperymentów dla metody referencyjnej na zbiorze ISOT

Metryka	**XGBOOST**	**SVM**	**DECISION TREE**	**MLP**
Accuracy	0.8455	0.8448	0.7826	0.8452
Balanced Accuracy	0.8501	0.8497	0.7803	0.8608
F_1 (0)	0.8391	0.8347	0.7795	0.8455
F_1 (1)	0.8491	0.8440	0.7670	0.8526
Precision (0)	0.8482	0.8441	0.7570	0.8556
Precision (1)	0.8257	0.8286	0.7819	0.8456
Recall (0)	0.8384	0.8351	0.8098	0.8566
Recall (1)	0.8449	0.8442	0.7332	0.8451

Jak można zauważyć dla miary dokładności najwyższe rezultaty otrzymał model XGBoost z wynikiem 84.55% oraz model MLP, którego dokładność wynosiła 84.52%. Dla metody referencyjnej najgorsze wyniki uzyskano dla klasyfikatora drzew decyzyjnych, który przy dokładności 78,26% wykazuje najniższą efektywność spośród analizowanych metod. W metryce zbalansowanej dokładności podobnie, jak w metryce dokładności, najlepsze rezultaty otrzymał model MLP, a najgorsze bazujący na drzewach decyzyjnych.

Wyniki uzyskane przy użyciu miary F_1 wskazują następujące wartości dla poszczególnych klas. Dla klasy 0 model MLP uzyskał wynik 84.55%, a dla klasy 1 osiągnął 85.26%, co wskazuje na jego skuteczność w klasyfikacji obu klas. Model drzew decyzyjnych osiągnął gorsze wyniki, szczególnie dla klasy 1, gdzie miara ta wyniosła 76.70%.

Analiza wyników metody LFM (Tabela 5.5) wskazuje, że najwyższą dokład-

Tabela 5.5: Tabela z wynikami eksperymentów dla metody LFM na zbiorze ISOT

Metryka	XGBOOST	SVM	DECISION TREE	MLP
Accuracy	0.9505	0.9461	0.8736	0.9582
Balanced Accuracy	0.9505	0.9464	0.8725	0.9581
F_1 (0)	0.9527	0.9481	0.8810	0.9601
F_1 (1)	0.9482	0.9440	0.8651	0.9562
Precision (0)	0.9533	0.9562	0.8675	0.9593
Precision (1)	0.9475	0.9355	0.8808	0.9573
Recall (0)	0.9520	0.9400	0.8950	0.9610
Recall (1)	0.9489	0.9528	0.8501	0.9552

ność (93.82%) uzyskał model MLP. Dla metody tej od wyników odbiega metoda wykorzystująca drzewa decyzyjne, która z wynikiem 87.36% wykazuje najgorszą efektywność spośród zaproponowanych klasyfikatorów. Kolejną miarą przedstawioną w Tabeli jest zbalansowana dokładność. W metryce tej podobnie jak w dokładności najlepszymi rezultatami charakteryzuje się model MLP, a najgorszymi drzewa decyzyjne.

Analiza wyników miary F_1 pokazuje, że model MLP osiągnął 96.01% dla klasy 0 oraz 95.62% dla klasy 1. Model drzew decyzyjnych uzyskał gorsze wyniki, w szczególności dla klasy 1, gdzie miara F_1 wyniosła 86.51%.

Tabela 5.6: Tabela z wynikami eksperymentów metody GEM na zbiorze ISOT

Metryka	XGBOOST	SVM	DECISION TREE	MLP
Accuracy	0.9418	0.9531	0.8175	0.9561
Balanced Accuracy	0.9417	0.9532	0.8161	0.9559
F_1 (0)	0.9444	0.9550	0.8291	0.9582
F_1 (1)	0.9390	0.9511	0.8042	0.9539
Precision (0)	0.9445	0.9588	0.8124	0.9557
Precision (1)	0.9389	0.9470	0.8236	0.9566
Recall (0)	0.9443	0.9513	0.8465	0.9606
Recall (1)	0.9391	0.9551	0.7857	0.9512

Analizując wyniki uzyskane przez drugą z zaproponowanych metod GEM przedstawione w Tabeli 5.6 można zauważyć, że dla mary dokładności najwyższe rezultaty otrzymał model MLP z wynikiem 95.61% oraz SVM 95.31%. W przypadku tej metody znacząco od pozostałych wyników odbiega klasyfikator wykorzystujący drzewa decyzyjne, który przy dokładności 81,75% charakteryzuje się

najniższą efektywnością spośród zaproponowanych modeli. Kolejną miarą przedstawioną w Tabeli 5.6 jest zbalansowana dokładność. W metryce tej podobnie jak w dokładności najlepsze rezultaty otrzymał model MLP oraz SVM, a najgorsze wykorzystujący drzewa decyzyjne.

Dla klasy 0 w mierze F_1 model MLP uzyskał wynik 95.82%, a dla klasy 1 (oznaczonej jako wiadomości prawdziwe) osiągnął 95.39%, co wskazuje na wysoką skuteczność w klasyfikacji obu klas. Model Drzew decyzyjnych osiągnął znacznie gorsze wyniki, szczególnie dla klasy 1, gdzie miara ta wyniosła 80.42%.

Test Friedmana wykazał istotne statystycznie różnice pomiędzy analizowanymi metodami (χ^2 = 17.64, p = 0.0001). W celu identyfikacji par metod różniących się istotnie zastosowano testy post-hoc Wilcoxona z korekcją Holma-Bonferroniego. Analiza wykazała istotne statystycznie różnice pomiędzy wszystkimi parami metod (p < 0.05). Najwyższą średnią wartość zbalansowanej dokładności uzyskała metoda LFM (0.9505 ± 0.0206), która okazała się istotnie lepsza od metody GEM (0.9417 ± 0.01) oraz metody referencyjnej (0.8501 ± 0.0387). Dodatkowo, na podstawie rang Friedmana, metoda LFM uzyskała najniższą średnią rangę równą 1.2727, metoda GEM rangę 1.7273, natomiast metoda referencyjna rangę 3.00.

Na podstawie uzyskanych wyników można stwierdzić, że zastosowanie metod hybrydowych w przypadku tego zbioru danych przyniosło rzeczywistą poprawę wyników detekcji dezinformacji. Dodatkowo porównując wyniki uzyskane na obu zbiorach, można zauważyć, że połączenie metod takich jak MLP oraz SVM z hybrydowymi metodami ekstrakcji cech pozwala uzyskać wysoką skuteczność poprawnej detekcji dezinformacji w tekście.

5.2 Polskie binarne zbiory danych

W poniższym podrozdziale zostały szerzej omówione wyniki uzyskane podczas badań zaproponowanych metod ekstrakcji cech na polskich binarnych zbiorach danych.

5.2.1 Infotester

Pierwszym polskim zbiorem danych wykorzystanym w niniejszej rozprawie jest zbiór Infotester. Zbiór ten jest zbiorem niezbalansowanym.

W Tabeli 5.7 zostały przedstawione wyniki uzyskane przez metodę referencyjną. Analiza wyników dla wszystkich modeli przedstawionych w Tabeli 5.7 obrazuje ich istotne różnice w skuteczności, zwłaszcza w kontekście klasyfikacji danych o niezbalansowanym rozkładzie klas. Podstawowa miara ogólnej poprawności klasyfikacji, którą jest dokładność (ang. Accuracy) dla przedstawionych klasyfikatorów wynosi odpowiednio 86.22%, 85.46%, 82.30% oraz 86.26%. Pomimo relatywnie wysokich wartości, miara ta nie pozwala w pełni ocenić skuteczności modeli.

Tabela 5.7: Tabela z wynikami eksperymentów dla metody referencyjnej na zbiorze Infotester

Metryka	**XGBOOST**	**SVM**	**DECISION TREE**	**MLP**
Accuracy	0.8622	0.8546	0.8230	0.8626
Balanced Accuracy	0.7672	0.6693	0.7379	0.7964
F_1 (0)	0.8872	0.8771	0.8738	0.8865
F_1 (1)	0.6741	0.5485	0.5715	0.6830
Precision (0)	0.8855	0.8645	0.8767	0.8895
Precision (1)	0.6948	0.7148	0.5585	0.6736
Recall (0)	0.8942	0.9029	0.8690	0.8850
Recall (1)	0.6501	0.4415	0.6052	0.6829

Analizując znacznie bardziej miarodajną metrykę, którą jest zbalansowana dokładność można zauważyć, że najwyższy wynik uzyskał model MLP, a najniższy model SVM. Biorąc pod uwagę powyższe zauważalna jest tendencja do faworyzowania klasy większościowej przez model SVM. Wskaźnikiem potwierdzającym tę hipotezę jest wartość miary F_1, która w przypadku klasy oznaczonej jako 1 dla modelu SVM osiąga wartość na poziomie 54.85% przy 87.71% dla klasy oznaczonej jako 0. Ponadto wartość tej metryki potwierdza wynik uzyskany w zbalansowanej dokładności gdzie ponownie najlepsze rezultaty uzyskał model MLP.

Wyniki przedstawione w Tabeli 5.8 wskazują na najwyższą skuteczność modelu XGBoost, który osiągnął 86.24% dokładności oraz 84.69% zbalansowanej dokładności. MLP również uzyskał wysokie wyniki, zwłaszcza w klasyfikacji klasy pozytywnej, gdzie osiągnął najlepszy wynik miary F_1 (88.64%), precyzji (89.17%) i czułości (88.02%).

W kontekście miary F_1, zarówno dla klasy 0, jak i 1, MLP i XGBoost wyróżniały się wysoką skutecznością, zwłaszcza w rozpoznawaniu klasy pozytywnej, gdzie oba modele osiągnęły wyniki powyżej 87%. Dla metody SVM uzyskano

Tabela 5.8: Tabela z wynikami eksperymentów dla metody LFM na zbiorze Infotester

Metryka	XGBOOST	SVM	DECISION TREE	MLP
Accuracy	0.8624	0.8231	0.7639	0.8614
Balanced Accuracy	0.8469	0.7990	0.7298	0.8491
F_1 (0)	0.8080	0.7433	0.6477	0.8101
F_1 (1)	0.8788	0.8521	0.8100	0.8864
Precision (0)	0.8015	0.7352	0.6444	0.8017
Precision (1)	0.8862	0.8560	0.8170	0.8917
Recall (0)	0.8133	0.7517	0.6489	0.8168
Recall (1)	0.8807	0.8528	0.8116	0.8802

wyniki na poziomie 82% dla klasy 0 i 85% dla klasy 1.

W Tabeli 5.9 zostały przedstawione wyniki z przeprowadzonego eksperymentów dla metody GEM. Analizując uzyskane wyniki, można zauważyć, że druga (GEM) z zaproponowanych metod osiąga lepsze wyniki od metody LFM.

Tabela 5.9: Tabela z wynikami eksperymentów dla metody GEM na zbiorze Infotester

Metryka	XGBOOST	SVM	DECISION TREE	MLP
Accuracy	0.9509	0.9248	0.8202	0.9491
Balanced Accuracy	0.9360	0.9115	0.7805	0.9444
F_1 (0)	0.8986	0.8646	0.6855	0.9057
F_1 (1)	0.9729	0.9506	0.8765	0.9681
Precision (0)	0.8978	0.8510	0.6877	0.8807
Precision (1)	0.9734	0.9574	0.8754	0.9807
Recall (0)	0.8995	0.8790	0.6834	0.9328
Recall (1)	0.9725	0.9440	0.8777	0.9560

Porównując wartości miary dokładności, można zauważyć, że modele XGBoost, SVM oraz MLP osiągneły bardzo wysokie wyniki przekraczające 92%. Wskazuje to na ich skuteczność w klasyfikacji dezinformacji. Zbalansowana dokładność, będąca średnią z czułości dla każdej klasy, dostarcza dodatkowych informacji przy analizie niezbalansowanych zbiorów danych. Również w tym przypadku wymienione trzy modele wykazały wysokie wartości (powyżej 91%), natomiast model drzew decyzyjnych osiągnął znacznie niższy wynik (78.05%).

Przeprowadzony test Friedmana potwierdził istnienie statystycznie istotnych różnic między analizowanymi rozwiązaniami ($\chi^2 = 22.00$, $p = 0.00002$). Szczegółowa analiza post-hoc z wykorzystaniem testu Wilcoxona (z korekcją Holma-Bonferroniego) wykazała, że różnice te występują dla wszystkich par metod ($p < 0.05$). Najwyższą średnią zbalansowaną dokładność odnotowano dla metody GEM (0.9360 ± 0.0124), która uzyskała również najniższą średnią rangę Friedmana (1.00), wyprzedzając metodę LFM (2.00) oraz metodę referencyjną (3.00)

5.2.2 OpenFact

Kolejnym zbiorem danych wykorzystanym w niniejszej rozprawie jest zbiór OpenFact. Zbiór ten, jak opisano we wcześniejszej części pracy, jest zbiorem niezbalansowanym.

Tabela 5.10: Tabela z wynikami eksperymentów dla metody referencyjnej na zbiorze OpenFact

Metryka	XGBOOST	SVM	DECISION TREE	MLP
Accuracy	0.8544	0.8498	0.8180	0.8562
Balanced Accuracy	0.7965	0.7659	0.7359	0.8174
F_1 (0)	0.8656	0.8632	0.8449	0.8666
F_1 (1)	0.7637	0.7361	0.6164	0.7761
Precision (0)	0.8595	0.8515	0.8471	0.8653
Precision (1)	0.8112	0.8340	0.6043	0.7864
Recall (0)	0.8719	0.8751	0.8427	0.8679
Recall (1)	0.7211	0.6567	0.6290	0.7669

Analizując wyniki detekcji dezinformacji uzyskane przez metodę referencyjną (Tabela 5.10) można zauważyć wyraźne zróżnicowanie skuteczności, zwłaszcza w kontekście rozpoznawania klasy mniejszościowej. Najwyższe wartości metryk ogólnych osiągnął model MLP, uzyskując dokładność na poziomie 85.62% oraz zbalansowaną dokładność na poziomie 81.74%, co wskazuje na jego największą efektywność zarówno pod względem ogólnej trafności klasyfikacji, jak i zrównoważonego rozpoznawania obu klas. Model XGBoost również osiągnął wysokie wyniki zwłaszcza w zakresie dokładności (85.44%) oraz F_1 dla klasy 0 (86.56%), zachowując jednocześnie relatywnie dobre zrównoważenie klasyfikacji. Należy jednak zauważyć, że model ten uzyskał niższe wyniki dla klasyfikacji klasy mniejszościowej. Najsłabsze wyniki podobnie jak we wcześniejszych eksperymentach uzyskał model wykorzystujący algorytm drzew decyzyjnych.

Tabela 5.11: Tabela z wynikami eksperymentów dla metody LFM na zbiorze OpenFact

Metryka	XGBOOST	SVM	DECISION TREE	MLP
Accuracy	0.8744	0.8698	0.8380	0.8762
Balanced Accuracy	0.8165	0.7859	0.7559	0.8374
F_1 (0)	0.8856	0.8832	0.8649	0.8866
F_1 (1)	0.7837	0.7561	0.6364	0.7961
Precision (0)	0.8795	0.8715	0.8671	0.8853
Precision (1)	0.8312	0.8540	0.6243	0.8064
Recall (0)	0.8919	0.8951	0.8627	0.8879
Recall (1)	0.7411	0.6767	0.6490	0.7869

W Tabeli 5.11 zostały przedstawione wyniki eksperymentów przeprowadzone dla zaproponowanej metody hybrydowej LFM. Jak można zauważyć najwyższe rezultaty uzyskuje model MLP. Jego dokładność wynosi 87.62%, a zbalansowana dokładność 83.74%, co potwierdza jego efektywność względem ogólnej skuteczności klasyfikacji oraz rozpoznawania obu klas. Najniższe wyniki uzyskał model drzew decyzyjnych, osiągając dokładność na poziomie 83.8%. Szczególnie niska była skuteczność w klasyfikacji klasy mniejszościowej, gdzie miara F_1 wyniosła 63.64%.

Tabela 5.12: Tabela z wynikami eksperymentów dla metody GEM na zbiorze OpenFact

Metryka	XGBOOST	SVM	DECISION TREE	MLP
Accuracy	0.9444	0.9398	0.9080	0.9462
Balanced Accuracy	0.8865	0.8559	0.8259	0.9074
F_1 (0)	0.9556	0.9532	0.9349	0.9566
F_1 (1)	0.8537	0.8261	0.7064	0.8661
Precision (0)	0.9495	0.9415	0.9371	0.9553
Precision (1)	0.9012	0.9240	0.6943	0.8764
Recall (0)	0.9619	0.9651	0.9327	0.9579
Recall (1)	0.8111	0.7467	0.7190	0.8569

W Tabeli 5.12 zostały przedstawione wyniki eksperymentów metody hybrydowej GEM. Jak można zauważyć najwyższe rezultaty uzyskuje model MLP. Jego dokładność wynosi 94.62%, a zbalansowana dokładność 90.74%. Co potwierdza jego efektywność względem ogólnej skuteczności klasyfikacji oraz rozpoznawania obu klas. Równie wysokie wyniki osiągnął model XGBoost. Jego rezultaty

były niewiele gorsze od modelu MLP. Najniższe wyniki osiągnął model drzew decyzyjnych, osiągając 90.8% dokładności, jednak co istotne najgorzej radzi sobie z rozpoznawaniem klasy mniejszościowej osiągając jedyne 73.64% miary F_1.

Przeprowadzony test Friedmana potwierdził istnienie statystycznie istotnych różnic między analizowanymi rozwiązaniami ($\chi^2 = 22.00$, $p = 0.00002$). Szczegółowa analiza post-hoc z wykorzystaniem testu Wilcoxona (z korekcją Holma-Bonferroniego) wykazała, że różnice te występują dla wszystkich par metod ($p < 0.05$). Najwyższą średnią zbalansowaną dokładność odnotowano dla metody GEM (0.8865 ± 0.0132), która uzyskała również najniższą średnią rangę Friedmana (1.00), wyprzedzając metodę LFM (2.00) oraz metodę referencyjną (3.00).

5.3 Polski zbiór wieloetykietowy

Podczas analizy wyników klasyfikacji poszczególnych grup pytań (weryfikacyjnych, manipulacyjnych oraz metafizycznych), zaobserwowano wyraźne różnice w skuteczności poszczególnych modeli. W każdej z grup można wyróżnić zarówno najlepsze, jak i najsłabsze podejścia modelowe, co odzwierciedlono poniżej.

5.3.1 Wyniki dla metody referencyjnej

W tabeli 5.13 przedstawione zostały wyniki metody referencyjnej dla pytania "Czy istnieje co najmniej jedno wiarygodne źródło, które potwierdza wszystkie informacje zawarte w treści?". Zauważalna jest w otrzymanych wynikach tendencja, gdzie model MLP osiąga najwyższą skuteczność metryki dokładności (70.01%). Sugeruje to zdolność modelu do poprawnego klasyfikowania większości próbek. Pozostałe modele wykazują dokładność niższą, ale porównywalną z modelem MLP.

Tabela 5.13: Wyniki metody referencyjnej – pytanie Q1

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.6878	0.6752	0.6763	0.7001
Balanced Accuracy	0.6602	0.5864	0.6532	0.6639
F_1 (0)	0.7762	0.8276	0.7589	0.7982
F_1 (1)	0.5405	0.0865	0.5461	0.5143
Precision (0)	0.7409	0.6752	0.7366	0.7419
Precision (1)	0.5843	0.1531	0.5718	0.6003
Recall (0)	0.8159	0.9564	0.7831	0.8680
Recall (1)	0.5045	0.0865	0.5233	0.4598

Analiza metryki zbalansowanej dokładności, która dokładniej odzwierciedla wydajność modeli na niezbalansowanym zbiorze danych niż wcześniej omówiona miara, wskazuje, że model MLP osiągnął wartość 66.39%, nieznacznie przewyższając wynik uzyskany przez XGBoost. Najniższe wyniki odnotowano dla modelu SVM, który osiągnął 58.64%, co wskazuje na niższą wydajność w klasyfikacji klasy mniejszościowej.

W Tabeli 5.14 przedstawione zostały wyniki dla pytania, "Czy większość podanych informacji jest potwierdzona przez wiarygodne źródła?". Podobnie jak w przypadku poprzedniego pytania najwyższe wyniki uzyskał model MLP (69.86%) oraz model SVM (68.95%) w metryce dokładności. Należy jednak wspomnieć, że uzyskane różnice pomiędzy wszystkimi modelami są niewielkie.

Tabela 5.14: Wyniki metody referencyjnej – pytanie Q2

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.6847	0.6895	0.6744	0.6886
Balanced Accuracy	0.6631	0.6117	0.6563	0.6651
F_1 (0)	0.7655	0.8432	0.7498	0.7719
F_1 (1)	0.5580	0.1190	0.5618	0.5531
Precision (0)	0.7333	0.6892	0.7289	0.7344
Precision (1)	0.5972	0.8346	0.5856	0.6019
Recall (0)	0.8013	0.9804	0.7723	0.8153
Recall (1)	0.5250	0.1147	0.5404	0.5149

W aspekcie metryki zbalansowanej dokładności podobnie jak w przypadku wcześniejszego pytania najwyższą wartość uzyskał model MLP (66.51%) oraz mo-

del XGBoost (66.31%), co potwierdza zdolność obydwu modeli do równomiernej analizy klas. Model SVM wykazuje natomiast najniższy wynik w tej metryce, co potwierdza jego tendencję do faworyzowania klasy większościowej.

Tabela 5.15: Wyniki metody referencyjnej – pytanie Q3

Metryka	**XGBoost**	**SVM**	**DECISION TREE**	**MLP**
Accuracy	0.7740	0.8000	0.7118	0.7648
Balanced Accuracy	0.6375	0.5888	0.6407	0.5908
F_1 (0)	0.3600	0.0888	0.4509	0.2282
F_1 (1)	0.8985	0.9200	0.8279	0.8877
Precision (0)	0.4363	0.0888	0.4321	0.4053
Precision (1)	0.8449	0.8000	0.8477	0.7985
Recall (0)	0.3142	0.0888	0.4722	0.1789
Recall (1)	0.9607	0.9588	0.8091	0.9543

Tabela 5.15 prezentuje wyniki dla pytania "Czy żadna z informacji nie jest potwierdzona przez wiarygodne źródła?", które wykazują odmienne charakterystyki. Najwyższą dokładność uzyskał model SVM (80%), a tuż za nim model XGBoost. Warto jednak zwrócić uwagę na wartość metryki zbalansowanej dokładności, gdzie model drzew decyzyjnych ma najwyższą wartość 64.07% oraz model XGBoost, który uzyskał wynik na poziomie 63.75%.

Tabela 5.16: Wyniki metody referencyjnej – pytanie Q4

Metryka	**XGBoost**	**SVM**	**DECISION TREE**	**MLP**
Accuracy	0.7333	0.7539	0.7101	0.7533
Balanced Accuracy	0.6542	0.5971	0.6505	0.6536
F_1 (0)	0.8500	0.8900	0.8196	0.8751
F_1 (1)	0.4501	0.0971	0.4809	0.3966
Precision (0)	0.8012	0.7539	0.8001	0.8000
Precision (1)	0.5186	0.0971	0.5037	0.5354
Recall (0)	0.9064	0.9671	0.8403	0.9685
Recall (1)	0.4021	0.0971	0.4607	0.3355

W tabeli 5.16 zostały przedstawione wyniki dla pytania "Czy stwierdzenie odnosi się do aktualnych danych?". Pod względem miary dokładności modele SVM (75.39%) i MLP (75.33%) osiągają najwyższe, bardzo zbliżone wartości. Jednakże, analiza metryki zbalansowanej dokładności wykazuje, że modele XGBoost

(65.42%) i drzewo decyzyjne (65.05%) ponownie wykazują przewagę nad SVM (59.71%).

Tabela 5.17: Tabela z wynikami eksperymentów metody referencyjnej - pytanie Q5

Metryka	**XGBoost**	**SVM**	**DECISION TREE**	**MLP**
Accuracy	0.7265	0.7605	0.6792	0.7526
Balanced Accuracy	0.6461	0.6031	0.6436	0.6453
F_1 (0)	0.4369	0.1031	0.5084	0.3669
F_1 (1)	0.8455	0.8964	0.7764	0.8786
Precision (0)	0.5030	0.1031	0.4886	0.5163
Precision (1)	0.7969	0.7605	0.7973	0.7957
Recall (0)	0.3906	0.1031	0.5304	0.3045
Recall (1)	0.9016	0.9731	0.7568	0.9755

W tabeli 5.17 przedstawione zostały wyniki dla pytania "Czy do właściwego zrozumienia treści wymagane są dodatkowe informacje?". Najwyższą dokładność dla tego pytania odnotowuje model SVM. Pozostałe modele uzyskują niższe wartości tej metryki. Analizując jednak metrykę zbalansowanej dokładności, stwierdzono, że najwyższe rezultaty osiągają modele XGBoost i MLP, podczas gdy podobnie jak we wcześniejszych pytaniach model SVM odnotowuje najniższy rezultat, co potwierdza jego właściwość faworyzowania klasy większościowej.

Tabela 5.18: Tabela z wynikami eksperymentów metody referencyjnej - pytanie Q6

Metryka	**XGBoost**	**SVM**	**DECISION TREE**	**MLP**
Accuracy	0.7413	0.7601	0.6930	0.7402
Balanced Accuracy	0.6431	0.5856	0.6480	0.6124
F_1 (0)	0.4095	0.0856	0.4969	0.3069
F_1 (1)	0.8638	0.8912	0.7964	0.8679
Precision (0)	0.4824	0.0856	0.4772	0.4608
Precision (1)	0.8118	0.7601	0.8170	0.7823
Recall (0)	0.3617	0.0856	0.5190	0.2465
Recall (1)	0.9245	0.9556	0.7770	0.9673

Kolejnym pytaniem w zbiorze danych było pytanie "Czy treść zawiera nieścisłości?", gdzie uzyskane wyniki zostały przedstawione w Tabeli 5.18. Ponownie

najlepszy rezultat jeżeli chodzi o dokładność uzyskał model SVM (76.01%). Jednakże analiza wyników w metryce zbalansowanej dokładności ponownie wskazuje, że model ten ma problemy z faworyzowaniem klasy dominującej, a najlepsze rezultaty w tej metryce osiąga model drzew decyzyjnych oraz XGBoost.

Analiza metryki F_1 pokazuje podobnie jak w przypadku wcześniejszego pytania wzorce, gdzie dla klasy mniejszościowej najsłabsze wartości ma model SVM, jednocześnie notujący najwyższe wyniki dla klasy dominującej.

W Tabeli 5.19 przedstawiono wyniki dla pytania "Czy stwierdzenie zawiera fragmenty wyrwane z kontekstu?". Model SVM (80.15%) ponownie osiąga najwyższą dokładność, a za nim plasuje się XGBoost (78.33%). Jednakże, zbalansowana dokładność pokazuje, że dla modeli drzew decyzyjnych i XGBoost uzyskany wynik jest zdecydowanie wyższy niż dla MLP i SVM. To ponownie sugeruje, że wysoka dokładność SVM i MLP może wynikać z faworyzowania jednej z klas.

Tabela 5.19: Tabela z wynikami eksperymentów metody referencyjnej - pytanie Q7

Metryka	**XGBoost**	**SVM**	**DECISION TREE**	**MLP**
Accuracy	0.7833	0.8015	0.7230	0.7597
Balanced Accuracy	0.6522	0.5907	0.6545	0.5967
F_1 (0)	0.3904	0.0907	0.4697	0.2607
F_1 (1)	0.9035	0.9216	0.8358	0.8813
Precision (0)	0.4700	0.0907	0.4500	0.4203
Precision (1)	0.8519	0.8015	0.8557	0.8008
Recall (0)	0.3411	0.0907	0.4921	0.2099
Recall (1)	0.9632	0.9607	0.8169	0.9466

Analizując metrykę F_1, obserwujemy bardzo niskie wartości klasy mniejszościowej dla wszystkich modeli, z SVM osiągającym najniższą wartość. Z drugiej strony, wartości dla klasy dominującej są bardzo wysokie dla wszystkich klasyfikatorów, z SVM na czele.

Tabela 5.20 prezentuje wyniki dla pytania "Czy autor stwierdzenia stosuje wybiórcze przedstawianie faktów?". Model SVM ponownie wyróżnia się najwyższą dokładnością (80.54%), z MLP (76.44%) i XGBoost (75.94%) tuż za nim. Pod względem zbalansowanej dokładności, drzewo decyzyjne i XGBoost wykazują bardzo zbliżone i najwyższe wartości.

64:68299575

Tabela 5.20: Tabela z wynikami eksperymentów metody referencyjnej - pytanie Q8

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.7594	0.8054	0.6995	0.7644
Balanced Accuracy	0.6385	0.6088	0.6394	0.5942
F_1 (0)	0.3775	0.1088	0.4645	0.2032
F_1 (1)	0.8841	0.9299	0.8118	0.8905
Precision (0)	0.4523	0.1088	0.4450	0.4167
Precision (1)	0.8310	0.8054	0.8326	0.7889
Recall (0)	0.3310	0.1088	0.4865	0.1611
Recall (1)	0.9460	0.9788	0.7922	0.9607

Tabela 5.21 przedstawia wyniki dla pytania "Czy autor stwierdzenia próbuje wprowadzić czytelnika w błąd?", które wydają się być najbardziej skrajnym przypadkiem niezbalansowania klas spośród analizowanych pytań. SVM osiąga wyjątkowo wysoką dokładność (95.09%), a za nim plasuje się MLP (94%) i XGBoost (89.23%). Jednakże, zbalansowana dokładność dla wszystkich modeli jest znacznie niższa i bardzo zbliżona, wahając się od 61.96% do 63.54%. Ta dysproporcja jest kluczowa i pokazuje kolejny raz, że metoda referencyjna nie jest w stanie poradzić sobie z ekstrakcją cech mogącą zniwelować dużą dysproporcję między klasami.

Tabela 5.21: Tabela z wynikami eksperymentów metody referencyjnej - pytanie Q9

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.8923	0.9509	0.8310	0.9400
Balanced Accuracy	0.6226	0.6199	0.6354	0.6196
F_1 (0)	0.2451	0.1199	0.3218	0.1499
F_1 (1)	0.9875	0.9899	0.9465	0.9882
Precision (0)	0.3020	0.1199	0.3065	0.3053
Precision (1)	0.9502	0.9509	0.9629	0.9496
Recall (0)	0.2151	0.1199	0.3400	0.1356
Recall (1)	0.9877	0.9899	0.9308	0.9882

Tabela 5.22 przedstawia wyniki dla pytania "Czy treść ma charakter satyryczny?". Obserwujemy tu bardzo wysokie wartości dokładności dla wszystkich modeli, z SVM osiągającym imponujące 98.86%, a za nim XGBoost (97.30%) i

MLP (96.76%). Jest to mocny sygnał, że większość próbek jest poprawnie klasyfikowana.

Tabela 5.22: Tabela z wynikami eksperymentów metody referencyjnej - pytanie Q10

Metryka	**XGBoost**	**SVM**	**DECISION TREE**	**MLP**
Accuracy	0.9730	0.9886	0.9105	0.9676
Balanced Accuracy	0.6142	0.6186	0.6083	0.6003
F_1 (0)	0.1911	0.1186	0.2193	0.1300
F_1 (1)	0.9766	0.9886	0.9686	0.9676
Precision (0)	0.2255	0.1186	0.2063	0.2195
Precision (1)	0.9766	0.9886	0.9686	0.9676
Recall (0)	0.1721	0.1186	0.2360	0.1164
Recall (1)	0.9766	0.9886	0.9686	0.9676

Jednakże analiza zbalansowanej dokładności ujawnia znacznie niższe wartości, wahające się od 60.03% (MLP) do 61.86% (SVM). Taka rozbieżność między dokładnością a zbalansowaną dokładnością sugeruje, że wysoka dokładność wynika głównie z doskonałego przewidywania klasy dominującej, a modele mają trudności z klasą mniejszościową.

Tabela 5.23 prezentuje wyniki dla pytania "Czy autor przyznaje, że przedstawione fakty są zmyślone?". Ponownie widzimy bardzo wysoką dokładność dla XGBoost (98.36%), SVM (95.25%) i MLP (95.43%).

Podobnie jak we wcześniejszym pytaniu, zbalansowana dokładność jest znacznie niższa, oscylując wokół 58%-61%. XGBoost (61.20%) osiąga najwyższą wartość w tej metryce, co sugeruje nieco lepszą równowagę w klasyfikacji obu klas, choć nadal z widocznymi ograniczeniami.

Tabela 5.23: Tabela z wynikami eksperymentów metody referencyjnej - pytanie Q11

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.9836	0.9525	0.9269	0.9543
Balanced Accuracy	0.6120	0.5825	0.5944	0.5846
F_1 (0)	0.1679	0.0825	0.1853	0.0911
F_1 (1)	0.9836	0.9525	0.9621	0.9543
Precision (0)	0.1883	0.0825	0.1750	0.1617
Precision (1)	0.9836	0.9525	0.9621	0.9543
Recall (0)	0.1563	0.0825	0.1987	0.0879
Recall (1)	0.9836	0.9525	0.9621	0.9543

Tabela 5.24 przedstawia wyniki dla pytania "Czy stwierdzenie zawiera obietnice polityczne?". Tutaj SVM (94.89%) i MLP (92.31%) ponownie wykazują najwyższą dokładność, z XGBoost (91.65%) plasującym się niewiele niżej.

W przypadku zbalansowanej dokładności, drzewo decyzyjne (64.62%) i XGBoost (64.31%) osiągają najlepsze wyniki, sugerując lepszą równowagę w klasyfikacji. MLP osiąga 63.81%, podczas gdy SVM 59.13% i jest najsłabszy w tej metryce.

Tabela 5.24: Tabela z wynikami eksperymentów metody referencyjnej - pytanie Q12

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.9165	0.9489	0.8528	0.9231
Balanced Accuracy	0.6431	0.5913	0.6462	0.6381
F_1 (0)	0.3033	0.0913	0.3328	0.2828
F_1 (1)	0.9659	0.9613	0.9495	0.9640
Precision (0)	0.3755	0.0913	0.3125	0.3973
Precision (1)	0.9652	0.9489	0.9603	0.9615
Recall (0)	0.2608	0.0913	0.3573	0.2397
Recall (1)	0.9659	0.9613	0.9351	0.9640

Tabela 5.25 prezentuje wyniki dla pytania "Czy stwierdzenie zawiera treści religijne? ". Modele SVM i MLP osiągają kolejno 98.68% i 98.03% , a XGBoost 97.89%.

Jednakże, zbalansowana dokładność dla wszystkich modeli oscyluje w zakresie 60%-62%. Model XGBoost (62.63%) osiąga najwyższą wartość, ale różnice są minimalne.

Tabela 5.25: Tabela z wynikami eksperymentów metody referencyjnej - pytanie Q13

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.9789	0.9868	0.9413	0.9803
Balanced Accuracy	0.6263	0.6168	0.6029	0.6106
F_1 (0)	0.2013	0.1168	0.1946	0.1269
F_1 (1)	0.9789	0.9868	0.9525	0.9803
Precision (0)	0.2358	0.1168	0.1832	0.1752
Precision (1)	0.9789	0.9868	0.9525	0.9803
Recall (0)	0.1816	0.1168	0.2091	0.1199
Recall (1)	0.9789	0.9868	0.9525	0.9803

5.3.2 Wyniki dla metody LFM

W niniejszym podrozdziale przedstawiono i omówiono wyniki uzyskane dla eksperymentów z zaproponowaną hybrydową metodą ekstrakcji cech LFM.

Wyniki dla pytania Q1 ("Czy istnieje co najmniej jedno wiarygodne źródło, które potwierdza wszystkie informacje zawarte w treści?") są przedstawione w Tabeli 5.26. Modele osiągają dokładność w zakresie od 71.33% (XGBoost) do 72.57% (SVM). Zbalansowana dokładność pozostaje na podobnym poziomie dla wszystkich modeli, oscylując w granicach 71.79–72.62%. Jest to znacząca poprawa w stosunku do metody referencyjnej, co sugeruje, że metoda LFM skutecznie redukuje problem niezbalansowania klas.

Tabela 5.26: Wyniki metody LFM – pytanie Q1

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.7133	0.7257	0.7158	0.7232
Balanced Accuracy	0.7219	0.7262	0.7179	0.7182
F_1 (0)	0.7342	0.7649	0.7517	0.7736
F_1 (1)	0.6903	0.6784	0.6735	0.6580
Precision (0)	0.8125	0.8136	0.8062	0.8043
Precision (1)	0.6297	0.6365	0.6278	0.6307
Recall (0)	0.6734	0.7234	0.7061	0.7466
Recall (1)	0.7704	0.7290	0.7298	0.6898

Tabela 5.27: Wyniki metody LFM – pytanie Q2

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.7141	0.7217	0.7180	0.7206
Balanced Accuracy	0.7216	0.7232	0.7166	0.7151
F_1 (0)	0.7300	0.7539	0.7576	0.7694
F_1 (1)	0.6969	0.6844	0.6705	0.6582
Precision (0)	0.8020	0.8009	0.7935	0.7910
Precision (1)	0.6402	0.6440	0.6385	0.6385
Recall (0)	0.6733	0.7137	0.7257	0.7499
Recall (1)	0.7700	0.7328	0.7076	0.6803

Dla pytania Q2 ("Czy większość podanych informacji jest potwierdzona przez wiarygodne źródła?"), wyniki w Tabeli 5.27 są bardzo podobne. Dokładność mieści się w zakresie 71.41% (XGBoost) do 72.17% (SVM), a zbalansowana dokładność wynosi od 71.51% dla MLP do 72.32% dla SVM. Wartości metryki F_1 dla klasy 0 (od 73.00% dla XGBoost do 76.94% dla MLP) i dla klasy 1 (od 65.82% dla MLP do 69.69% dla XGBoost) ponownie wskazują na zbalansowaną wydajność klasyfikatorów, potwierdzając skuteczność LFM w radzeniu sobie z niezbalansowaniem danych.

Wyniki dla pytania Q3 ("Czy żadna z informacji nie jest potwierdzona przez wiarygodne źródła?") przedstawiono w Tabeli 5.28. W tym przypadku, dokładność jest niższa niż dla wcześniejszych dwóch pytań, w zakresie 63,.10% (MLP) do 65.45% (SVM). Jednakże, zbalansowana dokładność pozostaje wysoka i stabilna dla wszystkich modeli. Jest to istotne, ponieważ odpowiedzi na to pytanie były niezbalansowane.

Tabela 5.28: Wyniki metody LFM – pytanie Q3

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.6324	0.6545	0.6418	0.6310
Balanced Accuracy	0.7040	0.7039	0.7045	0.7025
F_1 (0)	0.6045	0.5948	0.6012	0.6031
F_1 (1)	0.6578	0.7033	0.6771	0.6559
Precision (0)	0.4874	0.4881	0.4881	0.4865
Precision (1)	0.9184	0.9159	0.9178	0.9165
Recall (0)	0.8734	0.8207	0.8530	0.8716
Recall (1)	0.5346	0.5871	0.5560	0.5334

Tabela 5.29: Wyniki metody LFM – pytanie Q4

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.6858	0.7129	0.7045	0.7072
Balanced Accuracy	0.7226	0.7291	0.7232	0.7236
F_1 (0)	0.7098	0.7629	0.7516	0.7565
F_1 (1)	0.6593	0.6494	0.6460	0.6441
Precision (0)	0.8842	0.8851	0.8800	0.8801
Precision (1)	0.5574	0.5659	0.5605	0.5612
Recall (0)	0.6054	0.6773	0.6636	0.6713
Recall (1)	0.8397	0.7809	0.7829	0.7759

Dla pytania Q4 („Czy stwierdzenie odnosi się do aktualnych danych?”) wyniki przedstawione w Tabeli 5.29 wskazują, że dokładność modeli mieści się w przedziale od 68.58% (XGBoost) do 71.29% (SVM). Zbalansowana dokładność pozostaje zbliżona dla wszystkich modeli, wynosząc od 72.26% (XGBoost) do 72.91% (SVM), co świadczy o efektywności metody LFM w tworzeniu zbalansowanego zestawu cech.

Tabela 5.30: Tabela z wynikami eksperymentów metody LFM pytanie Q5

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.6602	0.6749	0.6619	0.6871
Balanced Accuracy	0.6984	0.6981	0.6970	0.6955
F_1 (0)	0.6389	0.6263	0.6353	0.6101
F_1 (1)	0.6799	0.7157	0.6859	0.7456
Precision (0)	0.5387	0.5390	0.5378	0.5373
Precision (1)	0.8573	0.8558	0.8554	0.8526
Recall (0)	0.8198	0.7718	0.8087	0.7221
Recall (1)	0.5770	0.6243	0.5853	0.6689

Wyniki dla pytania „Czy do właściwego zrozumienia treści wymagane są dodatkowe informacje?” przedstawiono w tabeli 5.30. Dokładność modeli mieści się w przedziale od 66.02% (XGBoost) do 68.71% (MLP). Zbalansowana dokładność pozostaje dla nich zbliżona, wynosząc od 69.55% (MLP) do 69.84% (XGBoost), co wskazuje na istotne ograniczenie problemu niezbalansowania danych obserwowanego w metodzie referencyjnej.

Tabela 5.31: Tabela z wynikami eksperymentów metody LFM pytanie Q6

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.6589	0.6696	0.6584	0.6642
Balanced Accuracy	0.7098	0.7062	0.7031	0.7052
F_1 (0)	0.6393	0.6272	0.6299	0.6290
F_1 (1)	0.6771	0.7060	0.6841	0.6943
Precision (0)	0.5291	0.5274	0.5247	0.5264
Precision (1)	0.8886	0.8823	0.8798	0.8820
Recall (0)	0.8555	0.8111	0.8313	0.8227
Recall (1)	0.5640	0.6013	0.5750	0.5877

Dla pytania "Czy treść zawiera nieścisłości?", wyniki w Tabeli 5.31 wykazują Dokładność w zakresie 65.84% do 66.96%. Zbalansowana dokładność wynosi od 70.31% do 70.98%, co ponownie wskazuje na sukces metody LFM w tworzeniu zbalansowanego środowiska klasyfikacyjnego. Wartości wskaźnika F_1 dla klasy 0 (od 62.72% dla SVM do 63.93% dla XGBoost) i dla klasy 1 (od 67.71% dla XGBoost do 70.60% dla SVM) są bardzo zbliżone, co potwierdza skuteczne radzenie sobie z niezbalansowaniem klas.

Tabela 5.32: Tabela z wynikami eksperymentów metody LFM pytanie Q7

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.6331	0.6504	0.6479	0.6457
Balanced Accuracy	0.7070	0.6999	0.7087	0.7034
F_1 (0)	0.6084	0.5921	0.6043	0.5987
F_1 (1)	0.6557	0.6983	0.6854	0.6857
Precision (0)	0.4897	0.4859	0.4913	0.4880
Precision (1)	0.9218	0.9112	0.9218	0.9159
Recall (0)	0.8825	0.8172	0.8529	0.8404
Recall (1)	0.5316	0.5825	0.5644	0.5665

Wyniki dla pytania Q7 („Czy stwierdzenie zawiera fragmenty wyrwane z kontekstu?") przedstawiono w Tabeli 5.32. Modele osiągnęły dokładność od 63.31% (XGBoost) do 65.04% (SVM). Zbalansowana dokładność waha się od 69.99% dla SVM do 70.87% dla drzewa decyzyjnego. Jest to szczególnie warte podkreślenia, ponieważ w metodzie referencyjnej wyniki dla tych pytań wskazywały na silne niezbalansowanie.

Tabela 5.33: Tabela z wynikami eksperymentów metody LFM pytanie Q8

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.6335	0.6523	0.6407	0.6475
Balanced Accuracy	0.6959	0.6951	0.6962	0.6959
F_1 (0)	0.6102	0.5994	0.6070	0.6030
F_1 (1)	0.6549	0.6965	0.6706	0.6856
Precision (0)	0.4973	0.4971	0.4976	0.4975
Precision (1)	0.8937	0.8919	0.8937	0.8931
Recall (0)	0.8546	0.8040	0.8373	0.8190
Recall (1)	0.5372	0.5862	0.5550	0.5728

Dla pytania Q8 ("Czy autor stwierdzenia stosuje wybiórcze przedstawianie faktów?"), wyniki w tabeli 5.33 pokazują dokładność w zakresie 63.35% (XGBoost) do 65.23% (SVM). Zbalansowana dokładność jest spójna dla wszystkich modeli (od 69.51% dla SVM do 69.62% dla drzewa decyzyjnego).

Tabela 5.34: Tabela z wynikami eksperymentów metody LFM pytanie Q9

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.4770	0.4915	0.5462	0.4890
Balanced Accuracy	0.6662	0.6493	0.6865	0.6782
F_1 (0)	0.4413	0.4212	0.4570	0.4539
F_1 (1)	0.5093	0.5492	0.6158	0.5207
Precision (0)	0.3307	0.3137	0.3516	0.3436
Precision (1)	0.9692	0.9518	0.9900	0.9826
Recall (0)	0.9521	0.8878	0.8984	0.9641
Recall (1)	0.3804	0.4109	0.4746	0.3924

Wyniki dla pytania Q9 ("Czy autor stwierdzenia próbuje wprowadzić czytelnika w błąd?") przedstawiono w Tabeli 5.34. W przeciwieństwie do poprzednich pytań, tutaj dokładność jest zauważalnie niższa, oscylując od 47.70% do 54.62%. Pomimo tego, zbalansowana dokładność utrzymuje się w przedziale od 64.93% (SVM) do 68.65% (drzewo decyzyjne). Co ważne uzyskane wyniki w metryce F_1 są znacznie bardziej zbliżone niż w metodzie referencyjnej, co wskazuje na poprawę w radzeniu sobie z klasą mniejszościową, nawet jeśli ogólna dokładność jest niższa.

Tabela 5.35: Tabela z wynikami eksperymentów metody LFM pytanie Q10

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.4136	0.4715	0.4468	0.3989
Balanced Accuracy	0.6677	0.6888	0.6577	0.6574
F_1 (0)	0.3342	0.3492	0.3184	0.3214
F_1 (1)	0.4791	0.5623	0.5406	0.4630
Precision (0)	0.2591	0.2738	0.2434	0.2456
Precision (1)	0.9743	0.9871	0.9569	0.9600
Recall (0)	0.9726	0.9565	0.9174	0.9600
Recall (1)	0.3547	0.4212	0.3979	0.3390

Dla pytania Q10 ("Czy treść ma charakter satyryczny?"), wyniki w Tabeli 5.35 pokazują niższą niż we wcześniejszych pytaniach dokładność (od 39.89% dla MLP do 47.15% dla SVM). Jednakże zbalansowana dokładność wynosi od 65.74% dla MLP do 68.88% dla SVM, co świadczy o lepszej zdolności metody LFM do klasyfikacji obu klas, mimo wyzwań związanych z ogólną dokładnością.

Dla pytania Q11 ("Czy autor przyznaje, że przedstawione fakty są zmyślone?"), wyniki w Tabeli 5.36 są podobne do Q9 i Q10, z niską dokładnością (od 36.18% dla MLP do 42.07% dla drzewa decyzyjnego). Zbalansowana dokładność pozostaje w zakresie (od 64.90% dla MLP do 66.63% dla XGBoost).

Tabela 5.36: Tabela z wynikami eksperymentów metody LFM pytanie Q11

Metryka	**XGBoost**	**SVM**	**DECISION TREE**	**MLP**
Accuracy	0.3839	0.4176	0.4207	0.3618
Balanced Accuracy	0.6663	0.6572	0.6505	0.6490
F_1 (0)	0.3136	0.3022	0.2984	0.2954
F_1 (1)	0.4434	0.5053	0.5121	0.4185
Precision (0)	0.2480	0.2372	0.2342	0.2296
Precision (1)	0.9786	0.9674	0.9653	0.9600
Recall (0)	0.9786	0.9423	0.9239	0.9600
Recall (1)	0.3302	0.3721	0.3770	0.3073

Wyniki dla pytania Q12 ("Czy stwierdzenie zawiera obietnice polityczne?") są przedstawione w Tabeli 5.37. Dokładność jest stosunkowo niska (od 43.94% do 48.01%). Jednakże, zbalansowana dokładność mieści się w zakresie od 66.19% dla XGBoost do 68.23% dla MLP. Wyniki metryki F_1 dla klasy 0 (od 39.48% dla XGBoost do 41.82% dla MLP) i dla klasy 1 (od 47.89% dla XGBoost do 54.25% dla drzewa decyzyjnego) są zrównoważone, co wskazuje na to, że LFM pozwala na bardziej efektywną klasyfikację obu klas, pomimo niższej ogólnej dokładności.

Dla pytania Q13 ("Czy stwierdzenie zawiera treści religijne?"), wyniki w Tabeli 5.38 również wykazują niską dokładność (od 33.08% do 37.61%). Mimo to, zbalansowana dokładność jest akceptowalna (od 65,31% do 66.91%). Wyniki metryki F_1 dla klasy 0 (od 27.81% do 29.30%) i dla klasy 1 (od 37.61% do 45.41%) są znacznie bardziej zbalansowane niż w metodzie referencyjnej, co sugeruje lepszą zdolność do rozróżniania klas.

Tabela 5.37: Tabela z wynikami eksperymentów metody LFM pytanie Q12

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.4394	0.4691	0.4801	0.4542
Balanced Accuracy	0.6619	0.6761	0.6734	0.6823
F_1 (0)	0.3948	0.4093	0.4034	0.4182
F_1 (1)	0.4789	0.5200	0.5425	0.4869
Precision (0)	0.2917	0.3075	0.3016	0.3156
Precision (1)	0.9537	0.9701	0.9623	0.9793
Recall (0)	0.9537	0.9654	0.9438	0.9793
Recall (1)	0.3508	0.3867	0.4031	0.3634

Tabela 5.38: Tabela z wynikami eksperymentów metody LFM pytanie Q13

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.3409	0.3662	0.3761	0.3308
Balanced Accuracy	0.6585	0.6691	0.6531	0.6545
F_1 (0)	0.2830	0.2930	0.2781	0.2796
F_1 (1)	0.3915	0.4279	0.4541	0.3761
Precision (0)	0.2221	0.2322	0.2179	0.2188
Precision (1)	0.9604	0.9704	0.9564	0.9572
Recall (0)	0.9604	0.9704	0.9553	0.9572
Recall (1)	0.2888	0.3165	0.3306	0.2776

5.3.3 Wyniki dla metody GEM

W niniejszym podrozdziale przedstawiono i omówiono wyniki uzyskane dla eksperymentów z zaproponowaną hybrydową metodą ekstrakcji cech GEM.

Dla pytania Q1 ("Czy istnieje co najmniej jedno wiarygodne źródło, które potwierdza wszystkie informacje zawarte w treści?"), wyniki przedstawione w Tabeli 5.39 pokazują, że dokładność modeli waha się od 81,21% (drzewo decyzyjne) do 83.21% (SVM). Natomiast zbalansowana dokładność wynosi od 81.61% (drzewo decyzyjne) do 82.90% (SVM).

Wyniki dla pytania Q2 ("Czy większość podanych informacji jest potwierdzona przez wiarygodne źródła?"), widoczne w Tabeli 5.40, są bardzo podobne. Dokładność utrzymuje się w zakresie od 81.68% (XGBoost) do 82.87% (SVM), a

zbalansowana dokładność jest konsekwentnie wysoka. Analogicznie, metryka F_1 dla klasy 0 (od 83.48% dla XGBoost do 87.18% dla MLP) i dla klasy 1 (od 76.69% dla MLP do 79.34% dla XGBoost) potwierdzają zbalansowaną dokładność klasyfikatorów, świadcząc o zdolności metody GEM do tworzenia zbalansowanego zestawu cech.

Tabela 5.39: Wyniki metody GEM – pytanie Q1

Metryka	**XGBoost**	**SVM**	**DECISION TREE**	**MLP**
Accuracy	0.8153	0.8321	0.8121	0.8164
Balanced Accuracy	0.8253	0.8290	0.8161	0.8221
F_1 (0)	0.8380	0.8751	0.8522	0.8693
F_1 (1)	0.7875	0.7820	0.7719	0.7641
Precision (0)	0.9185	0.9194	0.9048	0.9023
Precision (1)	0.7252	0.7412	0.7236	0.7241
Recall (0)	0.7689	0.8357	0.8045	0.8404
Recall (1)	0.8816	0.8222	0.8298	0.8073

Tabela 5.40: Wyniki metody GEM – pytanie Q2

Metryka	**XGBoost**	**SVM**	**DECISION TREE**	**MLP**
Accuracy	0.8168	0.8287	0.8203	0.8246
Balanced Accuracy	0.8239	0.8266	0.8194	0.8224
F_1 (0)	0.8348	0.8662	0.8579	0.8718
F_1 (1)	0.7934	0.7864	0.7753	0.7669
Precision (0)	0.9098	0.9140	0.9009	0.8995
Precision (1)	0.7300	0.7433	0.7342	0.7335
Recall (0)	0.7739	0.8263	0.8183	0.8471
Recall (1)	0.8740	0.8268	0.8215	0.8044

W przypadku pytania Q3 ("Czy żadna z informacji nie jest potwierdzona przez wiarygodne źródła?"), wyniki w Tabeli 5.41 wskazują na niższą w porównaniu do pytań Q1 i Q2 dokładność (od 73.37% dla MLP do 75.68% dla SVM). Jednakże, zbalansowana dokładność wynosi od 80.09% dla MLP do 80.61% dla SVM. Jest to szczególnie istotne, ponieważ ta metoda dobrze radzi sobie z danymi niezbalansowanymi i nie wykazuje problemów typowych dla innych metod.

Dla pytania Q4 ("Czy stwierdzenie odnosi się do aktualnych danych?"), wyniki w Tabeli 5.42 pokazują dokładność w zakresie od 79.14% do 81.14%. Zba-

lansowana dokładność jest również wysoka i spójna (od 82.54% dla XGBoost i drzewa decyzyjnego do 82.88% dla SVM). Uzyskane wyniki są dobrze zbalansowane, co potwierdza skuteczność GEM w generowaniu cech sprzyjających równomiernej klasyfikacji.

Tabela 5.41: Wyniki metody GEM – pytanie Q3

Metryka	**XGBoost**	**SVM**	**DECISION TREE**	**MLP**
Accuracy	0.8380	0.8568	0.8481	0.8337
Balanced Accuracy	0.8056	0.8061	0.8040	0.8009
F_1 (0)	0.7035	0.6965	0.7062	0.7028
F_1 (1)	0.7642	0.8007	0.8763	0.8592
Precision (0)	0.5910	0.5887	0.5871	0.5831
Precision (1)	0.9151	0.9141	0.9182	0.9176
Recall (0)	0.9626	0.9219	0.9655	0.9695
Recall (1)	0.6487	0.6904	0.6511	0.6377

Tabela 5.42: Wyniki metody GEM – pytanie Q4

Metryka	**XGBoost**	**SVM**	**DECISION TREE**	**MLP**
Accuracy	0.7914	0.8109	0.8082	0.8114
Balanced Accuracy	0.8254	0.8288	0.8254	0.8286
F_1 (0)	0.8143	0.8626	0.8547	0.8588
F_1 (1)	0.7623	0.7540	0.7524	0.7527
Precision (0)	0.9837	0.9816	0.9781	0.9792
Precision (1)	0.6563	0.6622	0.6582	0.6576
Recall (0)	0.7031	0.7789	0.7675	0.7725
Recall (1)	0.9477	0.8787	0.8822	0.8803

Wyniki dla pytania Q5 (“Czy do właściwego zrozumienia treści wymagane są dodatkowe informacje?”), przedstawione w Tabeli 5.43, pokazują dokładność od 81.01% do 86.86%. Zbalansowana dokładność mieści się w zakresie od 71.12% (SVM) do 78.08% (MLP). Warto zwrócić uwagę, że SVM, mimo najwyższej dokładności, ma najniższą zbalansowaną dokładność, co sugeruje faworyzowanie klasy większościowej.

Dla pytania Q6 ("Czy treść zawiera nieścisłości?"), wyniki w Tabeli 5.44 wykazują dokładność w zakresie od 81.78% do 88.83%. Podobnie jak w poprzednim pytaniu, zbalansowana dokładność dla SVM wynosi 71.38% i jest najniższa.

Tabela 5.43: Tabela z wynikami eksperymentów metody GEM na pytanie Q5

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.8601	0.8686	0.8101	0.8572
Balanced Accuracy	0.7769	0.7112	0.7755	0.7808
F_1 (0)	0.5620	0.2113	0.6419	0.5790
F_1 (1)	0.9798	0.9045	0.9063	0.9730
Precision (0)	0.6358	0.2113	0.6207	0.6409
Precision (1)	0.9273	0.8686	0.9287	0.9299
Recall (0)	0.5125	0.2113	0.6657	0.5381
Recall (1)	0.9413	0.9513	0.8853	0.9235

Tabela 5.44: Tabela z wynikami eksperymentów metody GEM na pytanie Q6

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.8740	0.8883	0.8178	0.8717
Balanced Accuracy	0.7740	0.7138	0.7741	0.7776
F_1 (0)	0.5373	0.2138	0.6241	0.5530
F_1 (1)	0.9967	0.9194	0.9211	0.8921
Precision (0)	0.6152	0.2138	0.6022	0.6205
Precision (1)	0.9423	0.8883	0.9444	0.9445
Recall (0)	0.4875	0.2138	0.6489	0.5079
Recall (1)	0.9605	0.9538	0.8993	0.9473

Tabela 5.45: Tabela z wynikami eksperymentów metody GEM na pytanie Q7

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.8771	0.9534	0.8463	0.9094
Balanced Accuracy	0.7441	0.7426	0.7790	0.7803
F_1 (0)	0.4789	0.2426	0.5943	0.5187
F_1 (1)	0.8976	0.9735	0.8598	0.9298
Precision (0)	0.5621	0.2426	0.5726	0.5953
Precision (1)	0.8441	0.8534	0.8825	0.8811
Recall (0)	0.3285	0.1426	0.5193	0.3741
Recall (1)	0.9588	0.9826	0.8386	0.9767
Specificity	0.8857	0.9033	0.9262	0.9168

Wyniki dla pytania Q7 ("Czy stwierdzenie zawiera fragmenty wyrwane z kontekstu?") znajdują się w Tabeli 5.45. Dokładność waha się od 84.63% (drzewo decyzyjne) do 95.34% (SVM). Zbalansowana dokładność jest na poziomie od 74.26% (SVM) do 78.03% (MLP). Tutaj SVM również, mimo najwyższej dokładności, nie ma najlepszej zbalansowanej dokładności.

Dla pytania Q8 ("Czy autor stwierdzenia stosuje wybiórcze przedstawianie faktów?"), wyniki w Tabeli 5.46 pokazują dokładność w zakresie od 72.65% (drzewo decyzyjne) do 83.74% (SVM). Zbalansowana dokładność wynosi od 64.08% (SVM) do 67.08% (XGBoost).

Tabela 5.46: Tabela z wynikami eksperymentów metody GEM na pytanie Q8

Metryka	**XGBoost**	**SVM**	**DECISION TREE**	**MLP**
Accuracy	0.7920	0.8374	0.7265	0.7652
Balanced Accuracy	0.6708	0.6408	0.6668	0.6653
F_1 (0)	0.4103	0.1408	0.4919	0.4363
F_1 (1)	0.9161	0.9619	0.8391	0.8873
Precision (0)	0.4878	0.1408	0.4718	0.4741
Precision (1)	0.8622	0.8374	0.8609	0.8594
Recall (0)	0.3625	0.1408	0.5149	0.4112
Recall (1)	0.9790	0.9808	0.8186	0.9195

Tabela 5.47: Tabela z wynikami eksperymentów metody GEM na pytanie Q9

Metryka	**XGBoost**	**SVM**	**DECISION TREE**	**MLP**
Accuracy	0.8990	0.9698	0.8197	0.9310
Balanced Accuracy	0.6273	0.6388	0.6248	0.6503
F_1 (0)	0.2469	0.1388	0.3127	0.2601
F_1 (1)	0.9630	0.9788	0.9341	0.9830
Precision (0)	0.3059	0.1388	0.2978	0.3417
Precision (1)	0.9553	0.9698	0.9496	0.9762
Recall (0)	0.2167	0.1388	0.3305	0.2264
Recall (1)	0.9630	0.9788	0.9192	0.9830

Wyniki dla pytania Q9 ("Czy autor stwierdzenia próbuje wprowadzić czytelnika w błąd?"), przedstawione w Tabeli 5.47, pokazują wysoką dokładność dla SVM (96.98%) i MLP (93.10%). Jednak zbalansowana dokładność jest znacznie niższa, w zakresie od 62.48% do 65.03%.

Tabela 5.48: Tabela z wynikami eksperymentów metody GEM na pytanie Q10

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.9835	0.9663	0.9222	0.9590
Balanced Accuracy	0.6514	0.6263	0.6212	0.6282
F_1 (0)	0.2282	0.1263	0.2327	0.1872
F_1 (1)	0.9835	0.9663	0.9520	0.9590
Precision (0)	0.2641	0.1263	0.2189	0.2768
Precision (1)	0.9835	0.9663	0.9520	0.9590
Recall (0)	0.2087	0.1263	0.2505	0.1630
Recall (1)	0.9835	0.9663	0.9520	0.9590

Tabela 5.49: Tabela z wynikami eksperymentów metody GEM na pytanie Q11

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.9766	0.9803	0.9496	0.9660
Balanced Accuracy	0.6357	0.6403	0.6151	0.6276
F_1 (0)	0.1919	0.1403	0.2044	0.1579
F_1 (1)	0.9766	0.9803	0.9561	0.9660
Precision (0)	0.2132	0.1403	0.1946	0.2218
Precision (1)	0.9766	0.9803	0.9561	0.9660
Recall (0)	0.1800	0.1403	0.2171	0.1452
Recall (1)	0.9766	0.9803	0.9561	0.9660

Dla pytania "Czy treść ma charakter satyryczny?", wyniki w Tabeli 5.48 również prezentują wysoką dokładność (od 92.22% dla drzewa decyzyjnego do 98.35% dla XGBoost). Zbalansowana dokładność jest znacznie niższa (od 62.12% do 65.14%). Analizując uzyskane wyniki widoczny jest problem niezbalansowania, gdzie modele doskonale klasyfikują klasę dominującą, ale z trudnością rozpoznają mniejszościową.

Dla pytania Q11 ("Czy autor przyznaje, że przedstawione fakty są zmyślone?"), wyniki w Tabeli 5.49 pokazują wysoką dokładność (od 94.96% do 98.03%). Zbalansowana dokładność waha się od 61.51% (drzewo decyzyjne) do 64.03% (SVM). Podobnie jak w poprzednich pytaniach, rezultaty metryki F_1 dla klasy 0 są bardzo niskie (od 14.03% dla SVM do 20.44% dla drzewa decyzyjnego), a dla klasy 1 bardzo wysokie (od 95.61% dla drzewa decyzyjnego do 98.03% dla SVM).

Tabela 5.50: Tabela z wynikami eksperymentów metody GEM na pytanie Q12

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.9707	0.9765	0.9078	0.9496
Balanced Accuracy	0.7042	0.6365	0.7035	0.6726
F_1 (0)	0.3753	0.1365	0.3904	0.3455
F_1 (1)	0.9882	0.9765	0.9876	0.9506
Precision (0)	0.4468	0.1365	0.3682	0.4810
Precision (1)	0.9882	0.9765	0.9876	0.9506
Recall (0)	0.3315	0.1365	0.4177	0.2832
Recall (1)	0.9882	0.9765	0.9873	0.9506

Wyniki dla pytania "Czy stwierdzenie zawiera obietnice polityczne?"są przedstawione w Tabeli 5.50. Dokładność wynosi od 90.78% dla drzewa decyzyjnego do 97.65% dla SVM. Zbalansowana dokładność przyjmuje wartości od 63.65% (SVM) do 70.42% (XGBoost).

Dla pytania Q13 ("Czy stwierdzenie zawiera treści religijne?"), wyniki w Tabeli 5.51 również wykazują wysoką dokładność (od 96.11% dla XGBoost do 98.07% dla MLP). Zbalansowana dokładność wynosi od 63.82% (XGBoost) do 65.13% (drzewo decyzyjne).

Tabela 5.51: Tabela z wynikami eksperymentów metody GEM na pytanie Q13

Metryka	XGBoost	SVM	DECISION TREE	MLP
Accuracy	0.9611	0.9780	0.9725	0.9807
Balanced Accuracy	0.6382	0.6200	0.6513	0.6460
F_1 (0)	0.2124	0.1200	0.2427	0.1812
F_1 (1)	0.9611	0.9780	0.9725	0.9807
Precision (0)	0.2475	0.1200	0.2299	0.2570
Precision (1)	0.9611	0.9780	0.9725	0.9807
Recall (0)	0.1927	0.1200	0.2594	0.1654
Recall (1)	0.9611	0.9780	0.9725	0.9807

Analiza statystyczna przeprowadzona dla wszystkich trzech grup pytań potwierdziła spójność uzyskanych rezultatów. Test Friedmana wykazał istnienie statystycznie istotnych różnic między badanymi rozwiązaniami w każdym z rozważanych przypadków:

- grupa czynników weryfikacyjnych $\chi^2 = 22.00$, $p = 0.00002$,
- grupa czynników manipulacyjnych $\chi^2 = 22.00$, $p = 0.00002$,
- grupa czynników metafizycznych $\chi^2 = 20.18$, $p = 0.00004$.

Szczegółowe testy post-hoc Wilcoxona z zastosowaniem korekcji Holma-Bonferroniego wykazały, że we wszystkich grupach różnice między każdą parą metod są statystycznie istotne.

We wszystkich analizowanych scenariuszach hierarchia rozwiązań pozostała niezmienna, co obrazuje poniższe zestawienie średnich rang Friedmana (tab.5.52).

Tabela 5.52: Zestawienie średnich rang Friedmana

Grupa czynników	GEM	LFM	Referencyjna
weryfikacyjnych	1.00	2.00	3.00
manipulacyjnych	1.00	2.00	3.00
metafizycznych	1.00	2.09	2.91

Najlepsze wyniki konsekwentnie osiągała metoda GEM, która w każdej grupie uzyskała najniższą możliwą średnią rangę. Na drugim miejscu uplasowała się metoda LFM, natomiast najniższą efektywność odnotowano dla metody referencyjnej. Uzyskane wyniki pokazują, że zastosowanie hybrydowych metod ekstrakcji cech w problemie detekcji dezinformacji statystycznie istotnie poprawia skuteczność detekcji.

5.4 Analiza porównawcza z wynikami przedstawionymi w innych pracach naukowych

W niniejszym podrozdziale dokonano analizy porównawczej wyników uzyskanych w ramach przeprowadzonych badań z rezultatami dostępnymi w innych badaniach naukowych. W analizie tej skupiono się na porównaniu wyników z polskim zbiorem Openfact.

W tabeli 5.53 przedstawione zostały wyniki zaprezentowane w pracy [98]. Wyniki te zawierają porównanie modeli GPT oraz BERT.

Tabela 5.53: Porównanie modeli na zbiorze OpenFact na podstawie [98]

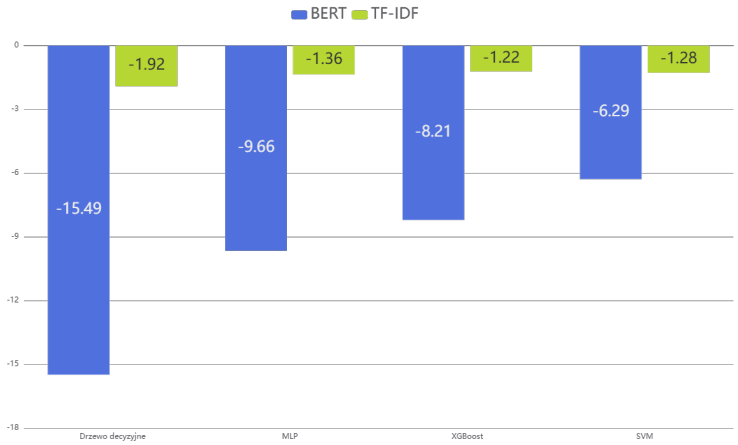
Model	F_1	**Precision**	**Recall**	**Accuracy**
GPT-3 curie fine-tuned curated	0.898	0.948	0.852	0.934
DeBERTa v3 base fine-tuned	0.894	0.978	0.824	0.934
GPT-3 davinci fine-tuned curated	0.876	0.946	0.815	0.921
RoBERTa base fine-tuned	0.862	0.966	0.778	0.915
RoBERTa base fine-tuned with custom optimizer layer-wise learning rate decay	0.860	0.976	0.769	0.915
LightGBM ensemble of all BERT-based models and additional embeddings	0.854	0.976	0.759	0.912
ELECTRA fine-tuned	0.851	0.954	0.769	0.909
AlBERT large v2 fine-tuned	0.848	0.976	0.750	0.909
DistilBERT base uncased fine-tuned	0.827	0.952	0.731	0.896
GPT-3 curie fine-tuned random	0.826	1.000	0.704	0.899
GPT neo 125M fine-tuned	0.800	0.961	0.685	0.884
GPT-4 few-shot learning	0.788	0.867	0.722	0.868
GPT-4 zero-shot learning	0.778	0.710	0.861	0.833
GPT-4 Chain-of-Thought	0.722	0.574	0.972	0.745
LFM (metoda autorska)	**0.8347**	**0.8554**	**0.8165**	**0.8744**
GEM (metoda autorska)	**0.9047**	**0.9254**	**0.8865**	**0.9444**

Analiza uzyskanych wyników dla zaproponowanych metod (Tabele 5.11 oraz 5.12) pozwala na interesujące porównanie z wynikami modeli opisanych w literaturze.

Mimo, że część modeli przedstawionych w Tabeli 5.53 to duże modele językowe (LLM), wyniki uzyskane w niniejszej pracy pokazują, że klasyczne algorytmy uczenia maszynowego w połączeniu z zaproponowanymi hybrydowymi metodami ekstrakcji cech (LFM i GEM), okazują się być konkurencyjne, a nawet osiągają lepsze wyniki niż te przedstawione we wspomnianej publikacji. W szczególności analizując wyniki uzyskane w ramach metody GEM widać wyraźną poprawę wyników.

5.5 Analiza wpływu komponentów modelu na otrzymane wyniki

Przeprowadzone badanie wpływu komponentów modelu miało na celu szczegółową analizę wpływu poszczególnych modułów zaproponowanych metod. Celem badania była ocena synergicznego działania wszystkich elementów zaproponowanych metod oraz ocena wpływu braku któregokolwiek z nich na działanie metody. Badanie to zostało przeprowadzone na wszystkich zbiorach danych wykorzystanych w niniejszej pracy, a uzyskane wyniki uśrednione i przedstawione poniżej. Jako miarę wykorzystano metrykę dokładności.



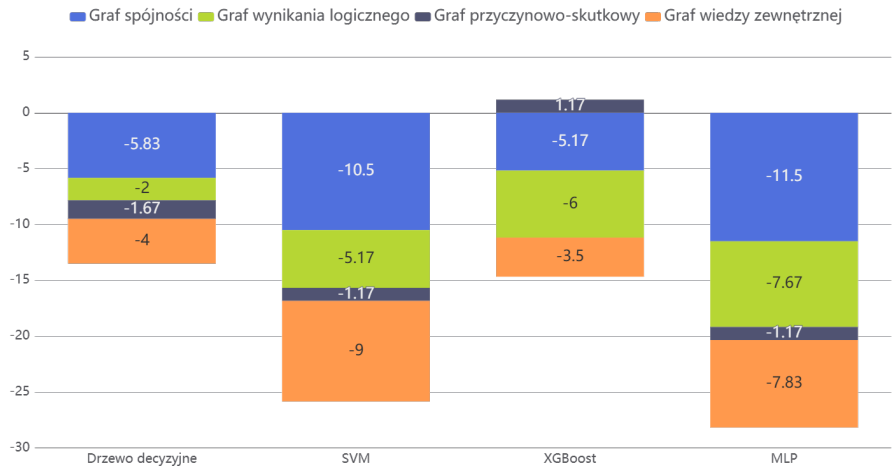
Rysunek 5.1: Uśrednione wartości wpływu komponentów w metodzie LFM

Na wykresie 5.1 zostały przedstawione wyniki metody LFM. Uzyskane wyniki jednoznacznie wskazują, że najlepsze rezultaty w tej metodzie osiągane są w sytuacji, gdy reprezentacje tekstu pochodzą jednocześnie z modelu BERT (DistilBERT) i TF-IDF. Jak widać na przedstawionym wykresie zaproponowana metoda z synergicznym działaniem obu komponentów, osiąga najwyższe wyniki. Natomiast w przypadku scenariusza, w którym zabrakło danych z modelu DistilBERT uzyskane wyniki są gorsze od 15.49% dla modelu drzew decyzyjnych do 6.29% dla modelu SVM. Natomiast sama reprezentacja BERT, bez TF-IDF dała rezultaty niższe o -1.92% dla drzew decyzyjnych do -1.22% dla modelu XGBoost.

Na wykresie 5.2 zostały przedstawione wyniki metody GEM. Z analizy otrzymanych rezultatów wynika, że największy negatywny wpływ na otrzymane wyniki ma usunięcie grafu spójności (kolor niebieski). Jego usunięcie powoduje spadek skuteczności od -5.83% dla drzew decyzyjnych do aż -11.5% dla MLP. Istotne znaczenie mają również graf wiedzy zewnętrznej (kolor pomarańczowy) i graf wynikania logicznego (kolor zielony), które najmocniej osłabiają modele SVM i

MLP.

Najmniejszy wpływ na wyniki ma usunięcie grafu przyczynowo-skutkowego (kolor szary). Co interesujące, w przypadku modelu XGBoost jego brak powoduje nawet nieznaczną poprawę wyniku o 1.17%. Najbardziej wrażliwymi na brak analizowanych komponentów grafowych okazały się modele MLP i SVM, gdzie zanotowano największe łączne spadki skuteczności.



Rysunek 5.2: Uśrednione wartości wpływu komponentów w metodzie GEM

Podsumowując, analiza wyników obu metod uwydatnia znaczenie odpowiedniego doboru i synergicznego łączenia różnych komponentów reprezentacji tekstu (LFM) oraz strategii przetwarzania tekstu (GEM). Takie podejście jest niezbędne do osiągnięcia dużej efektywności w zadaniach związanych z detekcją dezinformacji, co bezpośrednio przekłada się na skuteczność jej zwalczania. Uzyskane wyniki jednoznacznie pokazują, że zastosowanie modeli hybrydowych znacząco poprawia skuteczność systemów, w pełni potwierdzając postawioną w pracy tezę.

6. Podsumowanie

W niniejszej rozprawie zaproponowano nowe autorskie hybrydowe metody ekstrakcji cech tekstu przeznaczone do detekcji dezinformacji.

W przedłożonej pracy udowodniono słuszność podejścia polegającego na wykorzystaniu metod hybrydowych oraz ich przewagę w efektywności detekcji dezinformacji nad metodami bazującymi na modelach transformerowych. Ponadto w pracy udowodniono, iż proponowane metody lepiej radzą sobie także w przypadku danych niezbalansowanych. W rozprawie zademonstrowano także dobrą skuteczność hybrydowych metod wykrywania dezinformacji na zbiorach danych w językach mniej zasobnych, np. w języku polskim.

W pracy potwierdzono hipotezę, że zastosowanie hybrydowych metod ekstrakcji cech, łączących reprezentacje semantyczne, logiczne, kontekstowe oraz przyczynowo-skutkowe z mechanizmem wag adaptacyjnych, pozwala na istotne zwiększenie skuteczności detekcji dezinformacji w porównaniu z podejściami opartymi wyłącznie na modelach transformerowych. Niniejszą hipotezę potwierdzono poprzez wykonanie następujących zadań badawczych:

1. **Opracowanie hybrydowej metody ekstrakcji cech uwzględniającej strukturę leksykalną oraz kontekst semantyczny.**
   To zadanie zostało zrealizowane poprzez opracowanie metody LFM. Metoda ta opiera się na zastosowaniu modelu DistilBERT, który umożliwia efektywne wychwycenie kontekstu semantycznego w tekście oraz statystycznej metody TF-IDF, wykorzystywanej do uzyskania reprezentacji tekstu z uwzględnieniem jego struktury leksykalnej. Dzięki połączeniu tych dwóch podejść możliwe jest stworzenie reprezentacji tekstu, która bierze pod uwagę zarówno znaczenie słów w kontekście, jak i ich względną ważność w całym zbiorze danych. Takie podejście pozwala na bardziej precyzyjne modelowanie informacji zawartych w tekście, uwzględniając jednocześnie aspekty syntaktyczne i semantyczne.

2. **Opracowanie metody łączącej informacje semantyczne, logiczne, kontekstowe oraz przyczynowo-skutkowe w celu poprawy jakości ekstrakcji cech.**
   To zadanie zostało zrealizowane poprzez opracowanie metody GEM, wyko-

rzystującej zintegrowane podejście oparte na czterech typach grafów reprezentujących różne aspekty struktury tekstu. Wyniki uzyskane z poszczególnych komponentów zostały następnie zintegrowane w jeden wektor cech, który odzwierciedla zarówno znaczenie semantyczne, jak i relacje logiczne oraz kontekstowe w analizowanym tekście. Takie połączenie umożliwiło uzyskanie bogatszej i bardziej informatywnej reprezentacji tekstu, zwiększającej jakość ekstrakcji informacji w porównaniu z podejściami bazującymi wyłącznie na pojedynczych modelach semantycznych.

3. **Zaprojektowanie mechanizmu wag adaptacyjnych, którego celem jest dostosowanie znaczenia poszczególnych typów reprezentacji.**
   To zadanie zostało zrealizowane poprzez zaprojektowanie mechanizmu wag adaptacyjnych w ramach enkodera (w metodzie LFM) opartego na dwóch warstwach liniowych z funkcją aktywacji ReLU oraz mechanizmem Dropout. Mechanizm ten umożliwia automatyczne dostosowanie znaczenia poszczególnych typów reprezentacji wejściowych zarówno pochodzących z modelu BERT, jak i z wektoryzacji TF-IDF. Dzięki zastosowaniu wag adaptacyjnych sieć może dynamicznie przyznawać większą lub mniejszą istotność każdej reprezentacji, co pozwala na uzyskanie bardziej precyzyjnej i informatywnej reprezentacji tekstu. Takie podejście integruje zalety obu typów reprezentacji, łącząc semantyczną głębię modelu BERT z bardziej statystycznym podejściem TF-IDF.

4. **Badanie skuteczności metod hybrydowych w detekcji dezinformacji w porównaniu z referencyjnym podejściem opartym wyłącznie na modelach transformerowych.**
   To zadanie zostało zrealizowane poprzez zastosowanie metod hybrydowych ekstrakcji cech, które wskazują wyraźną przewagę w porównaniu z referencyjnym podejściem opartym wyłącznie na modelach transformerowych. Analiza eksperymentalna wykazała, że zastosowanie autorskiej metody GEM pozwala na uzyskanie ogólnej średnie skuteczności wyższej o około 10 punktów procentowych - 16 p.p. w odniesieniu do referencyjnej metody. Metoda LFM również zwiększa ogólną skuteczność w stosunku do metody referencyjnej, choć w mniejszym stopniu niż ma to miejsce w przypadku metody GEM. Wyniki uzyskane przez tą metodę mieszczą się w przedziale od 0.77 p.p. (XGBoost) do 11.44 p.p. (SVM). Warto zwrócić jednak uwagę na szczególnie istotne zmiany w metrykach klasy mniejszościowej, gdzie metoda LFM zwiększa F_1, czułość oraz zbalansowaną dokładność nawet o kilkadziesiąt punktów procentowych, co wskazuje na skuteczne radzenie sobie z problemem niezbalansowanych danych. Metoda GEM nie tylko poprawia F_1 i czułość dla obu klas, ale również zwiększa ogólną precyzję, co przekłada się na bardziej zbalansowaną i stabilną detekcję dezinformacji.

W przypadku wszystkich klasyfikatorów GEM osiąga najlepsze wyniki w większości metryk, zapewniając średnią poprawę w skuteczności o ponad 10 p.p. Wyniki te jednoznacznie potwierdzają, że integracja różnych typów reprezentacji cech w metodach hybrydowych znacząco zwiększa skuteczność detekcji dezinformacji, zarówno w kontekście jakości klasyfikacji, jak i odporności na nierównomierne rozłożenie klas.

5. **Przeprowadzenie eksperymentalnej ewaluacji proponowanych metod hybrydowych w zestawieniu z innymi podejściami opisanymi w literaturze w kontekście klasyfikacji dezinformacji.**
   Przeprowadzona eksperymentalna ewaluacja autorskich metod hybrydowych LFM oraz GEM, w kontekście klasyfikacji dezinformacji wykazała wyraźną przewagę nad metodami opartymi wyłącznie ma modelach transformerowych, w tym GPT-3, GPT-4 czy Electra. Wyniki przedstawione w literaturze pokazują, że metoda GEM osiąga najwyższą skuteczność spośród testowanych podejść. Natomiast metoda LFM mimo osiągnięcia niższych wyników, charakteryzuje się wyższą równowagą między precyzją a czułością w porównaniu do metod dostępnych w literaturze. Ponadto podejście to charakteryzuje się odpornością na problem niezbalansowania klas. Uzyskane wyniki i porównane z dostępnymi źródłami wskazują, że integracja różnych typów reprezentacji cech w metodach hybrydowych znacząco zwiększa skuteczność detekcji dezinformacji, jak i odporność na nierównomierne rozłożenie klas.

Bibliografia

[1] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *SIGKDD Explor.*, vol. 19, no. 1, pp. 22–36, 2017.

[2] T. Quandt, L. Frischlich, S. Boberg, and T. Schatto-Eckrodt, "Fake news," pp. 1–6, Apr. 2019.

[3] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018.

[4] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *J. Econ. Perspect.*, vol. 31, no. 2, pp. 211–236, 2017.

[5] R. Gill and R. Goolsby, *COVID-19 disinformation: A multi-national, whole of society perspective*. Cham, Switzerland: Springer Nature, 2022.

[6] P. Akhtar, A. M. Ghouri, H. U. R. Khan, M. Amin ul Haq, U. Awan, N. Zahoor, Z. Khan, and A. Ashraf, "Detecting fake news and disinformation using artificial intelligence and machine learning to avoid supply chain disruptions," *Ann. Oper. Res.*, vol. 327, no. 2, pp. 633–657, 2023.

[7] K. Shu, S. Wang, D. Lee, and H. Liu, *Disinformation, misinformation, and fake news in social media: Emerging research challenges and opportunities*. Berlin, Germany: Springer, 2020.

[8] E. Aïmeur, S. Amri, and G. Brassard, "Fake news, disinformation and misinformation in social media: a review," *Soc. Netw. Anal. Min.*, vol. 13, no. 1, 2023.

[9] E. Aïmeur, S. Amri, and G. Brassard, "Fake news, disinformation and misinformation in social media: a review," *Social network analysis and mining*, vol. 13, no. 1, p. 30, 2023. [Online]. Available: http://dx.doi.org/10.1007/s13278-023-01028-5

[10] "Komunikat komisji do parlamentu europejskiego, rady, europejskiego komitetu ekonomiczno-społecznego i komitetu regionów zwalczanie dezinformacji w internecie: podejście europejskie," dostęp: 2025-05-06. [Online]. Available: https://eur-lex.europa.eu/legal-content/PL/TXT/PDF/?uri=CELEX:52018DC0236

[11] A. Gelfert, "Fake news: A definition," *Informal logic*, vol. 38, no. 1, p. 84–117, 2018. [Online]. Available: http://dx.doi.org/10.22329/il.v38i1.5068

[12] J. Baptista and A. Gradim, "A working definition of fake news," *Encyclopedia*, vol. 2, no. 1, p. 632–645, 2022. [Online]. Available: http://dx.doi.org/10.3390/encyclopedia2010043

[13] S. R. Khater, O. H. Al-sahlee, D. M. Daoud, and M. S. A. El-Seoud, "Clickbait detection," in *Proceedings of the 7th International Conference on Software and Information Engineering*. New York, NY, USA: ACM, 2018.

[14] S. T. Smith, E. K. Kao, E. D. Mackin, D. C. Shah, O. Simek, and D. B. Rubin, "Automatic detection of influential actors in disinformation networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 118, no. 4, p. e2011216118, 2021. [Online]. Available: http://dx.doi.org/10.1073/pnas.2011216118

[15] H. T. Phan, N. T. Nguyen, and D. Hwang, "Fake news detection: A survey of graph neural network methods," *Applied soft computing*, vol. 139, no. 110235, p. 110235, 2023. [Online]. Available: http://dx.doi.org/10.1016/j.asoc.2023.110235

[16] E. Broda and J. Strömbäck, "Misinformation, disinformation, and fake news: lessons from an interdisciplinary, systematic literature review," *Annals of the International Communication Association*, vol. 48, no. 2, p. 139–166, 2024. [Online]. Available: http://dx.doi.org/10.1080/23808985.2024.2323736

[17] S. Koka, A. Vuong, and A. Kataria, "Evaluating the efficacy of large language models in detecting fake news: A comparative analysis," 2024.

[18] B. Hu, Q. Sheng, J. Cao, Y. Shi, Y. Li, D. Wang, and P. Qi, "Bad actor, good advisor: Exploring the role of large language models in fake news detection," *Proc. Conf. AAAI Artif. Intell.*, vol. 38, no. 20, pp. 22 105–22 113, 2024.

[19] J. Zhang, Y. Huang, S. Liu, Y. Gao, and X. Hu, "Do BERT-like bidirectional models still perform better on text classification in the era of LLMs?" 2025.

[20] Kowsari, J. Meimandi, Heidarysafa, Mendu, Barnes, and Brown, "Text classification algorithms: A survey," *Information (Basel)*, vol. 10, no. 4, p. 150, 2019.

[21] S. K. Hamed, M. J. Ab Aziz, and M. R. Yaakub, "A review of fake news detection approaches: A critical analysis of relevant studies and highlighting key challenges associated with the dataset, feature representation, and data fusion," *Heliyon*, vol. 9, no. 10, p. e20382, 2023.

[22] D. de Beer and M. Matthee, "Approaches to identify fake news: A systematic literature review," in *Integrated Science in Digital Age 2020*. Cham: Springer International Publishing, 2021, pp. 13–22.

[23] C. Zhang, A. Gupta, C. Kauten, A. V. Deokar, and X. Qin, "Detecting fake news for reducing misinformation risks using analytics approaches," *Eur. J. Oper. Res.*, vol. 279, no. 3, pp. 1036–1052, 2019.

[24] S. Garg and D. Kumar Sharma, "Linguistic features based framework for automatic fake news detection," *Comput. Ind. Eng.*, vol. 172, no. 108432, p. 108432, 2022.

[25] B. Horne and S. Adali, "This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 11, no. 1, pp. 759–766, 2017.

[26] E. Saquete, D. Tomás, P. Moreda, P. Martínez-Barco, and M. Palomar, "Fighting post-truth using natural language processing: A review and open challenges," *Expert Syst. Appl.*, vol. 141, no. 112943, p. 112943, 2020.

[27] K. Taha, P. D. Yoo, C. Yeun, D. Homouz, and A. Taha, "A comprehensive survey of text classification techniques and their research applications: Observational and experimental insights," *Computer Science Review*, vol. 54, no. 100664, p. 100664, 2024. [Online]. Available: http://dx.doi.org/10.1016/j.cosrev.2024.100664

[28] J. Camacho-Collados and M. T. Pilehvar, "On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis," 2017. [Online]. Available: http://arxiv.org/abs/1707.01780

[29] V. K. Pant, R. Sharma, and S. Kundu, "An overview of stemming and lemmatization techniques," *Advances in Networks, Intelligence and Computing*, pp. 308–321, 2024.

[30] C. W. Schmidt, V. Reddy, H. Zhang, A. Alameddine, O. Uzan, Y. Pinter, and C. Tanner, "Tokenization is more than compression," 2024. [Online]. Available: http://arxiv.org/abs/2402.18376

[31] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, K. Erk and N. A. Smith, Eds. Berlin, Germany: Association for Computational Linguistics, 2016, p. 1715–1725.

[32] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," 2016. [Online]. Available: http://arxiv.org/abs/1609.08144

[33] E. Gow-Smith, H. Tayyar Madabushi, C. Scarton, and A. Villavicencio, "Improving tokenisation by alternative treatment of spaces," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Stroudsburg, PA, USA: Association for Computational Linguistics, 2022, p. 11430–11443.

[34] S. Yehezkel and Y. Pinter, "Incorporating context into subword vocabularies," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, A. Vlachos and I. Augenstein, Eds. Stroudsburg, PA, USA: Association for Computational Linguistics, 2023, p. 623–635.

[35] M. Choraś, K. Demestichas, A. Giełczyk, Á. Herrero, P. Ksieniewicz, K. Remoundou, D. Urda, and M. Woźniak, "Advanced machine learning techniques for fake news (online disinformation) detection: A systematic mapping study," *Appl. Soft Comput.*, vol. 101, no. 107050, p. 107050, 2021.

[36] W. B. Cavnar and J. M. Trenkle, "N-gram-based text categorization," https://dsacl3-2019.github.io/materials/CavnarTrenkle.pdf, accessed: 2023-12-11.

[37] H. Jiang, Y. Xiao, and W. Wang, "Explaining a bag of words with hierarchical conceptual labels," *World Wide Web*, vol. 23, no. 3, pp. 1693–1713, 2020.

[38] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval (1972)," in *Ideas That Created the Future*. The MIT Press, 2021, pp. 339–348.

[39] L. Shi and R.-L. Xu, "Research on deep learning model based on word2vec and improved TF-IDF algorithm," *Computer and Digital Engineering*, vol. 49, no. 05, pp. 966–970, 2021.

[40] S. Raza and C. Ding, "Fake news detection based on news content and social contexts: a transformer-based approach," *Int. J. Data Sci. Anal.*, vol. 13, no. 4, pp. 335–362, 2022.

[41] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018.

[42] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019.

[43] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," 2019.

[44] L. Gomes, R. da Silva Torres, and M. L. Côrtes, "BERT- and TF-IDF-based feature extraction for long-lived bug prediction in FLOSS: A comparative study," *Inf. Softw. Technol.*, vol. 160, no. 107217, p. 107217, 2023.

[45] K. Jia, F. Meng, J. Liang, and P. Gong, "Text sentiment analysis based on BERT-CBLBGA," *Comput. Electr. Eng.*, vol. 112, no. 109019, p. 109019, 2023.

[46] Z. Zhu and K. Mao, "Knowledge-based BERT word embedding fine-tuning for emotion recognition," *Neurocomputing*, vol. 552, no. 126488, p. 126488, 2023.

[47] N. T. M. Trang and M. Shcherbakov, "Vietnamese question answering system f rom multilingual BERT models to monolingual BERT model," in *2020 9th International Conference System Modeling and Advancement in Research Trends (SMART)*. IEEE, 2020.

[48] G. Kątek, M. Gackowska, J. Komorniczak, P. Ksieniewicz, R. Kozik, M. Pawlicki, and M. Choraś, *Involving society to protect society from fake news and disinformation: Crowdsourced datasets and text reliability assessment*. Singapore: Springer Nature Singapore, 2024, p. 384–395.

[49] F. Farhangian, R. M. O. Cruz, and G. D. C. Cavalcanti, "Fake news detection: Taxonomy and comparative study," *Inf. Fusion*, vol. 103, no. 102140, p. 102140, 2024.

[50] M. K. Elhadad, K. F. Li, and F. Gebali, "A novel approach for selecting hybrid features from online news textual metadata for fake news detection," in *Advances on P2P, Parallel, Grid, Cloud and Internet Computing*. Cham: Springer International Publishing, 2020, pp. 914–925.

[51] M. Liang and T. Niu, "Research on text classification techniques based on improved TF-IDF algorithm and LSTM inputs," *Procedia Comput. Sci.*, vol. 208, pp. 460–470, 2022.

[52] A. Onan, "Hierarchical graph-based text classification framework with contextual node embedding and BERT-based dynamic fusion," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 7, p. 101610, 2023.

[53] H. T. Phan, N. T. Nguyen, and D. Hwang, "Fake news detection: A survey of graph neural network methods," *Appl. Soft Comput.*, vol. 139, no. 110235, p. 110235, 2023.

[54] K. Martínez-Gallego, A. M. Álvarez-Ortiz, and J. D. Arias-Londoño, "Fake news detection in spanish using deep learning techniques," 2021.

[55] Y. Blanco-Fernández, J. Otero-Vizoso, A. Gil-Solla, and J. García-Duque, "Enhancing misinformation detection in spanish language with deep learning: BERT and RoBERTa transformer models," *Appl. Sci. (Basel)*, vol. 14, no. 21, p. 9729, 2024.

[56] L. Ibañez-Lissen, L. González-Manzano, J. M. de Fuentes, and M. Goyanes, "On the feasibility of predicting volumes of fake news—the spanish case," *IEEE Trans. Comput. Soc. Syst.*, vol. 11, no. 4, pp. 5230–5240, 2024.

[57] P. X. Moreno-Vallejo, G. K. Bastidas-Guacho, P. R. Moreno-Costales, and J. J. Chariguaman-Cuji, "Fake news classification web service for spanish news by using artificial neural networks," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 3, 2023.

[58] R. Catelli, L. Bevilacqua, N. Mariniello, V. S. Di Carlo, M. Magaldi, H. Fujita, G. De Pietro, and M. Esposito, "A new italian cultural heritage data set: detecting fake reviews with BERT and ELECTRA leveraging the sentiment," *IEEE Access*, pp. 1–1, 2023.

[59] M. C. Buzea, S. Trausan-Matu, and T. Rebedea, "Automatic fake news detection for romanian online news," *Information (Basel)*, vol. 13, no. 3, p. 151, 2022.

[60] M. Bucos and G. Țucudean, "Text data augmentation techniques for fake news detection in the romanian language," *Appl. Sci. (Basel)*, vol. 13, no. 13, p. 7389, 2023.

[61] L. Dinu, E. C. Fusu, and D. Gifu, "Veracity analysis of romanian fake news," *Procedia Comput. Sci.*, vol. 225, pp. 3303–3312, 2023.

[62] A. Valeanu, D. P. Mihai, C. Andrei, C. Puscasu, A. M. Ionica, M. I. Hinoveanu, V. P. Predoi, E. Bulancea, C. Chirita, S. Negres, and C. D. Marineci, "Identification, analysis and prediction of valid and false information related to vaccines from romanian tweets," *Front. Public Health*, vol. 12, p. 1330801, 2024.

[63] E. V. Moisi, B. C. Mihalca, S. M. Coman, A. M. Pater, and D. E. Popescu, "Romanian fake news detection using machine learning and transformer-based approaches," *Appl. Sci. (Basel)*, vol. 14, no. 24, p. 11825, 2024.

[64] M. S. Farooq, A. Naseem, F. Rustam, and I. Ashraf, "Fake news detection in urdu language using machine learning," *PeerJ Comput. Sci.*, vol. 9, p. e1353, 2023.

[65] Z. Iqbal, F. M. Khan, I. U. Khan, and I. U. Khan, "Fake news identification in urdu tweets using machine learning models," *The Asian Bulletin of Big Data Management*, vol. 4, no. 1, 2024.

[66] S. Munir and M. Asif Naeem, "BiL-FaND: leveraging ensemble technique for efficient bilingual fake news detection," *Int. J. Mach. Learn. Cybern.*, vol. 15, no. 9, pp. 3927–3949, 2024.

[67] M. A. Al Ghamdi, M. S. Bhatti, A. Saeed, Z. Gillani, and S. H. Almotiri, "A fusion of BERT, machine learning and manual approach for fake news detection," *Multimed. Tools Appl.*, vol. 83, no. 10, pp. 30 095–30 112, 2023.

[68] S. Harris, H. J. Hadi, N. Ahmad, and M. A. Alshara, "Multi-domain urdu fake news detection using pre-trained ensemble model," *Sci. Rep.*, vol. 15, no. 1, p. 8705, 2025.

[69] S. E. Sorour and H. E. Abdelkader, "Afnd: Arabic fake news detection with an ensemble deep CNN-lstm model," *Journal of Theoretical and Applied Information Technology*, vol. 100, 2022.

[70] M. Azzeh, A. Qusef, and O. Alabboushi, "Arabic fake news detection in social media context using word embeddings and pre-trained transformers," *Arab. J. Sci. Eng.*, vol. 50, no. 2, pp. 923–936, 2025.

[71] M. E Almandouh, M. F. Alrahmawy, M. Eisa, M. Elhoseny, and A. S. Tolba, "Ensemble based high performance deep learning models for fake news detection," *Sci. Rep.*, vol. 14, no. 1, p. 26591, 2024.

[72] R. Mohawesh, S. Maqsood, and Q. Althebyan, "Multilingual deep learning framework for fake news detection using capsule neural network," *J. Intell. Inf. Syst.*, pp. 1–17, 2023.

[73] A. J. Keya, H. H. Shajeeb, M. S. Rahman, and M. F. Mridha, "FakeStack: Hierarchical Tri-BERT-CNN-LSTM stacked model for effective fake news detection," *PLoS One*, vol. 18, no. 12, p. e0294701, 2023.

[74] Y. Han, S. Karunasekera, and C. Leckie, "Graph neural networks with continual learning for fake news detection from social media," 2020.

[75] Y.-J. Lu and C.-T. Li, "GCAN: Graph-aware Co-Attention networks for explainable fake news detection on social media," 2020.

[76] U. Roy, M. S. Tahosin, M. M. Hasan, T. Islam, F. Imtiaz, M. R. Sadik, Y. Maleh, R. B. Sulaiman, and M. S. Hassan Talukder, "Enhancing bangla fake news detection using bidirectional gated recurrent units and deep learning techniques," in *Proceedings of the 7th International Conference on Networking, Intelligent Systems and Security*, vol. 2020. New York, NY, USA: ACM, 2024, pp. 1–10.

[77] A. M. Malla and A. A. Banka, "Sustainable signals: a heterogeneous graph neural framework for fake news detection," *Int. J. Syst. Assur. Eng. Manag.*, 2024.

[78] S. Frisli, "Semi-supervised self-training for COVID-19 misinformation detection: analyzing twitter data and alternative news media on norwegian twitter," *J. Comput. Soc. Sci.*, vol. 8, no. 2, 2025.

[79] P. Wanda and M. Diqi, "DeepNews: enhancing fake news detection using generative round network (GRN)," *Int. J. Inf. Technol.*, 2024.

[80] E. Canhasi, R. Shijaku, and E. Berisha, "Albanian fake news detection," *ACM Trans. Asian Low-resour. Lang. Inf. Process.*, vol. 21, no. 5, pp. 1–24, 2022.

[81] S. M. Isa, G. Nico, and M. Permana, "Indobert for indonesian fake news detection," *ICIC Express Letters*, vol. 16, no. 03, 2022.

[82] B. Xie, X. Ma, X. Shan, A. Beheshti, J. Yang, H. Fan, and J. Wu, "Multiknowledge and llm-inspired heterogeneous graph neural network for fake news detection," *IEEE Transactions on Computational Social Systems*, pp. 1–13, 2024.

[83] Z. Kang, Y. Cao, Y. Shang, T. Liang, H. Tang, and L. Tong, *Fake news detection with heterogenous deep graph convolutional network*. Cham: Springer International Publishing, 2021, p. 408–420.

[84] L. Sun and H. Wang, "Topic-aware fake news detection based on heterogeneous graph," *IEEE Access*, vol. 11, pp. 103 743–103 752, 2023.

[85] L. Xu, J. Peng, X. Jiang, E. Chen, and B. Luo, "Graph neural network based on graph kernel: A survey," *Pattern recognition*, vol. 161, no. 111307, p. 111307, 2025. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2024.111307

[86] A. S. Karnyoto, C. Sun, B. Liu, and X. Wang, "Augmentation and heterogeneous graph neural network for aaai2021-covid-19 fake news detection," *International journal of machine learning and cybernetics*, vol. 13, no. 7, p. 2033–2043, 2022. [Online]. Available: http://dx.doi.org/10.1007/s13042-021-01503-5

[87] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter,” 2020.

[88] Hugging Face, “Distilbert base uncased,” https://huggingface.co/distilbert/distilbert-base-uncased, 2019, dostęp: 11.09.2024.

[89] S. Dadas, “Polish distilroberta,” https://huggingface.co/sdadas/polish-distilroberta, 2021, dostęp: 11.09.2024.

[90] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” 2017.

[91] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using siamese BERT-networks,” 2019.

[92] M. Grootendorst, “BERTopic: Neural topic modeling with a class-based TF-IDF procedure,” 2022.

[93] Pawan Kumar Verma, Prateek Agrawal, Radu Prodan, “WELFake dataset for fake news detection in text data,” 2021. [Online]. Available: https://zenodo.org/records/4561253

[94] C. Bisaillon, “Isot fake and real news dataset,” 2020. [Online]. Available: https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset

[95] Uniwersytet Ekonomiczny w Poznaniu, “Openfact,” 2024. [Online]. Available: https://openfact.org/wynikifazyi

[96] Science4People spółka z ograniczoną odpowiedzialnością, Uniwersytet Medyczny w Białymstoku, Polsko-Japońska Akademia Technik Komputerowych, “Infotester,” 2024. [Online]. Available: https://infotester.pl/projekt-br/

[97] R. Kozik, J. Komorniczak, P. Ksieniewicz, A. Pawlicka, M. Pawlicki, and M. Choraś, “Swarog project approach to fake news detection problem,” in *Computational Intelligence in Security for Information Systems Conference*. Springer, 2023, pp. 79–88.

[98] M. Sawiński, K. Węcel, E. Księżniak, M. Stróżyna, W. Lewoniewski, P. Stolarski, and W. Abramowicz, “Notebook for the CheckThat! lab at CLEF 2023,” accessed: 2025-7-17.

Spis tabel

2.1 Podsumowanie zbiorów danych i technologii do wykrywania fałszywych wiadomości w różnych językach 19

4.1 Struktura i format zbioru danych WELFake 41

4.2 Struktura i format zbioru danych ISOT 42

5.1 Tabela z wynikami eksperymentów metody referencyjnej na zbiorze WELFake 50

5.2 Tabela z wynikami eksperymentów metody LFM na zbiorze WELFake . . 51

5.3 Tabela z wynikami eksperymentów dla metody GEM na zbiorze WELFake 52

5.4 Tabela z wynikami eksperymentów dla metody referencyjnej na zbiorze ISOT 53

5.5 Tabela z wynikami eksperymentów dla metody LFM na zbiorze ISOT . 54

5.6 Tabela z wynikami eksperymentów metody GEM na zbiorze ISOT 54

5.7 Tabela z wynikami eksperymentów dla metody referencyjnej na zbiorze Infotester 56

5.8 Tabela z wynikami eksperymentów dla metody LFM na zbiorze Infotester 57

5.9 Tabela z wynikami eksperymentów dla metody GEM na zbiorze Infotester 57

5.10 Tabela z wynikami eksperymentów dla metody referencyjnej na zbiorze OpenFact 58

5.11 Tabela z wynikami eksperymentów dla metody LFM na zbiorze OpenFact 59

5.12 Tabela z wynikami eksperymentów dla metody GEM na zbiorze OpenFact 59

5.13 Wyniki metody referencyjnej – pytanie Q1 61

5.14 Wyniki metody referencyjnej – pytanie Q2 61

5.15 Wyniki metody referencyjnej – pytanie Q3 62

5.16 Wyniki metody referencyjnej – pytanie Q4 62

5.17 Tabela z wynikami eksperymentów metody referencyjnej - pytanie Q5 63

5.18 Tabela z wynikami eksperymentów metody referencyjnej - pytanie Q6 63

5.19 Tabela z wynikami eksperymentów metody referencyjnej - pytanie Q7 64

5.20 Tabela z wynikami eksperymentów metody referencyjnej - pytanie Q8 65

5.21 Tabela z wynikami eksperymentów metody referencyjnej - pytanie Q9 65

5.22 Tabela z wynikami eksperymentów metody referencyjnej - pytanie Q10 66

5.23 Tabela z wynikami eksperymentów metody referencyjnej - pytanie Q11 67

5.24 Tabela z wynikami eksperymentów metody referencyjnej - pytanie Q12 67

5.25 Tabela z wynikami eksperymentów metody referencyjnej - pytanie Q13 68

5.26 Wyniki metody LFM – pytanie Q1 69

5.27 Wyniki metody LFM – pytanie Q2 69

5.28 Wyniki metody LFM – pytanie Q3 70
5.29 Wyniki metody LFM – pytanie Q4 70
5.30 Tabela z wynikami eksperymentów metody LFM pytanie Q5 71
5.31 Tabela z wynikami eksperymentów metody LFM pytanie Q6 71
5.32 Tabela z wynikami eksperymentów metody LFM pytanie Q7 72
5.33 Tabela z wynikami eksperymentów metody LFM pytanie Q8 72
5.34 Tabela z wynikami eksperymentów metody LFM pytanie Q9 73
5.35 Tabela z wynikami eksperymentów metody LFM pytanie Q10 73
5.36 Tabela z wynikami eksperymentów metody LFM pytanie Q11 74
5.37 Tabela z wynikami eksperymentów metody LFM pytanie Q12 75
5.38 Tabela z wynikami eksperymentów metody LFM pytanie Q13 75
5.39 Wyniki metody GEM – pytanie Q1 76
5.40 Wyniki metody GEM – pytanie Q2 76
5.41 Wyniki metody GEM – pytanie Q3 77
5.42 Wyniki metody GEM – pytanie Q4 77
5.43 Tabela z wynikami eksperymentów metody GEM na pytanie Q5 78
5.44 Tabela z wynikami eksperymentów metody GEM na pytanie Q6 78
5.45 Tabela z wynikami eksperymentów metody GEM na pytanie Q7 78
5.46 Tabela z wynikami eksperymentów metody GEM na pytanie Q8 79
5.47 Tabela z wynikami eksperymentów metody GEM na pytanie Q9 79

5.48 Tabela z wynikami eksperymentów metody GEM na pytanie Q10 80

5.49 Tabela z wynikami eksperymentów metody GEM na pytanie Q11 80

5.50 Tabela z wynikami eksperymentów metody GEM na pytanie Q12 81

5.51 Tabela z wynikami eksperymentów metody GEM na pytanie Q13 81

5.52 Zestawienie średnich rang Friedmana 82

5.53 Porównanie modeli na zbiorze OpenFact na podstawie [98] 83

Spis rysunków

2.1 Etapy procesu tworzenia modelu służącego do klasyfikacji tekstu. [Opracowanie własne] 16

2.2 Przegląd technik wykrywania dezinformacji (na podstawie [53]) 19

3.1 Architektura metody LFM 26

3.2 Schemat działania metody GEM 31

3.3 Schemat działania grafu spójności tekstu 32

3.4 Schemat działania modułu wynikania logicznego 33

3.5 Schemat działania modułu przyczynowo-skutkowego 34

3.6 Schemat działania modułu wiedzy zewnętrznej 35

4.1 Schemat procedury przeprowadzonych eksperymentów 38

4.2 Rozkład liczności etykiet w zbiorze danych WELFake 40

4.3 Rozkład liczności etykiet w zbiorze danych ISOT 41

4.4 Rozkład liczności etykiet w zbiorze danych Infotester 43

4.5 Rozkład etykiet w zbiorze danych SWAROG 45

5.1 Uśrednione wartości wpływu komponentów w metodzie LFM 84

5.2 Uśrednione wartości wpływu komponentów w metodzie GEM 85