Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH

Instytut Instytut Automatyki i Informatyki Stosowanej

Praca dyplomowa magisterska

na kierunku Informatyka
w specjalności Systemy Informacyjno-Decyzyjne

Wczesna detekcja fake newsów w medium społecznościowym na podstawie cech kaskad komunikacyjnych

Maria Oniszczuk
Numer albumu 271199

promotor
dr hab. inż. Mariusz Kamola

WARSZAWA 2023

Wczesna detekcja fake newsów w medium społecznościowym na podstawie cech kaskad komunikacyjnych

Streszczenie. Szacuje się, że w 2023 roku ponad połowa populacji świata (około 4,9 miliarda osób) ma dostęp oraz aktywnie korzysta z mediów społecznościowych [1]. Na przestrzeni lat główne zastosowanie tych portali ewoluowało: z narzędzia służącego do szybkiej komunikacji z bliskimi do źródła informacji na temat wydarzeń z całego świata. Wraz ze wzrostem popularności mediów społecznościowych rośnie skala problemu braku weryfikacji w czasie rzeczywistym znajdujących się na nich informacji. Każdy użytkownik takich portali ma możliwość tworzenia, komentowania oraz udostępniania dalej wybranych treści. To może prowadzić do nieświadomego bądź intencjonalnego rozpowszechniania fałszywych informacji, tak zwanych fake newsów, które w ten sposób mogą dotrzeć do dużej liczby osób. Wpływ celowej dezinformacji na kształtowanie opinii publicznej został zaobserwowany podczas wyborów prezydenckich w 2016 roku w USA prowadząc do manipulacji osądami odbiorców oraz polaryzacji społeczeństwa [2].

Aby zaadresować ten problem niniejsza praca skupia się na badaniach wczesnej detekcji nieprawdziwych informacji na portalu X (wcześniej znanego pod nazwą Twitter). Badaniom zostały poddane dwa zbiory danych możliwe do pobrania z repozytorium FakeNewsNet [3]. Ze względu na brak jednej, uniwersalnej metody pozwalającej na rekonstrukcję ścieżek, którymi dane treści docierały do konkretnych, udostępniających je dalej użytkowników, obliczenia zostały przeprowadzone dla dwóch różnych metod opisanych w literaturze, a wyniki porównano. Cechami branymi pod uwagę były atrybuty możliwe do wyodrębnienia na podstawie struktury propagacji komunikatów w czasie, profili użytkowników oraz właściwości sieci społecznościowej. W toku pracy zaproponowano nowe atrybuty i wykazano, że mają one pozytywny wpływ na jakość predykcji. Tekst wiadomości nie był brany pod uwagę, ponieważ fałszywe informacje są tworzone w taki sposób aby jak najdokładniej przypominać te prawdziwe, co sprawia, że wykrywanie nieprawdziwych wiadomości bazując na ich treści jest nietrywialnym zadaniem. Do klasyfikacji zostały wykorzystane algorytmy niebazujące na sieciach neuronowych, takie jak na przykład las losowy czy maszyna wektorów nośnych oraz grafowe sieci neuronowe, a dokładniej architektury oparte o warstwy GCNConv oraz GraphSAGE. W celu zbadania jak najwcześniejszej detekcji fałszywych informacji grafy odzwierciedlające propagację komunikatów zostały ograniczone zarówno ilościowo - do pierwszych 5, 10, 20, 30, 40, 50 i 100 postów jak również czasowo - dla pierwszych 5 i 30 minut oraz 1, 2, 4, 8, 12 i 24 godzin propagacji. Wyniki uzyskane dla obu typów modeli oraz różnych ograniczeń wielkości próbek w zbiorze danych zostały ze sobą porównane.

Słowa kluczowe: detekcja fake newsów, grafowe sieci neuronowe, uczenie maszynowe

Early fake news detection in social media based on the characteristics of communication cascades

Abstract. It is estimated as of 2023 that more than half of the world's population (about 4.9 billion people) have access to and actively engage with social media [1]. Throughout the years, it primary usage has evolved: from being a tool for communicating quickly with close ones to a go-to source for information on world events. Along with the increasing popularity of social media, the scale of the lack of real-time verification of the information found there is growing. Every user has the ability to create, comment on and further share selected content. This can lead to the unconscious or intentional dissemination of false information, so-called fake news, which can thus reach a great number of people. The impact of intentional disinformation on shaping public opinion was observed during the 2016 U.S. presidential election leading to manipulation of audience judgments and polarization of the society [2].

To address this problem, this thesis is focused on a study of early fake news detection on X (formerly known as Twitter). Two datasets available for download from the FakeNewsNet repository [3] were studied. Due to the lack of a universal method to reconstruct the paths by which the content in question reached the specific users sharing it further, calculations were performed for two different methods described in the literature, and the results were compared. The employed attributes were extracted from the structure of message propagation over time, user profiles and social network properties. New attributes were proposed over the course of the study and were shown to have a positive impact on prediction quality. The text of the news was not considered since fake informations are made to resemble real news as much as possible, thus making the detection of fraudulent messages based on its content a non-trivial task. Non-neural network-based algorithms, such as, for example, random forest or support vector machine, and graph neural networks, more specifically, architectures based on GCNConv and GraphSAGE layers, were used for classification. In order to analyze the earliest possible detection of false information, the graphs reflecting message propagation were limited both quantitatively - to the first 5, 10, 20, 30, 40, 50 and 100 posts - as well as temporally - for the first 5 and 30 minutes and 1, 2, 4, 8, 12 and 24 hours of propagation. The results obtained for both types of models and for different sample size restrictions in the dataset were compared with each other.

Keywords: fake news detection, graph neural networks, machine learning

załącznik nr 3 do zarządzenia
nr 28 /2016 Rektora PW

Politechnika Warszawska

……..
miejscowość i data

……..
imię i nazwisko studenta

……..
numer albumu

…….…….
kierunek studiów

OŚWIADCZENIE

Świadomy/-a odpowiedzialności karnej za składanie fałszywych zeznań oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie, pod opieką kierującego pracą dyplomową.

Jednocześnie oświadczam, że:

- niniejsza praca dyplomowa nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.) oraz dóbr osobistych chronionych prawem cywilnym,
- niniejsza praca dyplomowa nie zawiera danych i informacji, które uzyskałem/-am w sposób niedozwolony,
- niniejsza praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadawaniem dyplomów lub tytułów zawodowych,
- wszystkie informacje umieszczone w niniejszej pracy, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami,
- znam regulacje prawne Politechniki Warszawskiej w sprawie zarządzania prawami autorskimi i prawami pokrewnymi, prawami własności przemysłowej oraz zasadami komercjalizacji.

Oświadczam, że treść pracy dyplomowej w wersji drukowanej, treść pracy dyplomowej zawartej na nośniku elektronicznym (płycie kompaktowej) oraz treść pracy dyplomowej w module APD systemu USOS są identyczne.

.....
czytelny podpis studenta

Spis treści

1. Wstęp

W ostatniej dekadzie nastąpił gwałtowny rozwój mediów społecznościowych związany z ich stale rosnącą popularnością. Przestają być one tylko narzędziem ułatwiającym komunikację z bliskim gronem znajomych, ale dla wielu stanowią również główne źródło szybkiego pozyskiwania informacji. Większość platform społecznościowych daje możliwość tworzenia, komentowania oraz udostępniania różnego rodzaju treści, co napędza proces rozprzestrzeniania się nowych komunikatów, szczególnie w tematach, które wywołują kontrowersje.

Szacuje się, że w 2023 roku około 4,9 miliarda osób korzysta z mediów społecznościowych, a do roku 2027 ta liczba ma wzrosnąć do około 5,85 miliarda [1]. Według badań instytutu Pew Research Center przeprowadzonych w 2022 roku wśród Amerykanów około połowa dorosłych poniżej 30. roku życia jest skłonna zaufać informacjom pozyskanym z mediów społecznościowych [4]. Z kolei inna analiza, z tego samego okresu, pozwala na wgląd jakie portale społecznościowe są głównymi źródłami informacji dla dorosłych obywateli Stanów Zjednoczonych: 31% badanych deklaruje, że jest to Facebook, z kolei około 14% osób czerpie informacje z Twittera [5].

W sieci społecznościowej każdy może dodawać treści i udostępniać je swoim odbiorcom, którzy mają możliwość propagowania ich dalej. To sprawia, że nowe komunikaty pojawiają się w tempie, w którym coraz trudniej weryfikować ich prawdziwość w czasie rzeczywistym, zanim dotrą do szerokiego grona odbiorców. Fałszywe wiadomości dotyczące przede wszystkim obszarów związanych z polityką mogą prowadzić do błędnych osądów na podstawie niekompletnych zestawów informacji oraz polaryzacji społeczeństwa. Jako przykłady wpływu intencjonalnie rozpowszechnianych nieprawdziwych informacji podaje się wybory prezydenckie w USA w 2016 roku [2], odnośnie których pojawiają się zarzuty manipulacji opinią publiczną, oraz propagację fałszywych wiadomości szczególnie w początkowej fazie pandemii COVID-19 [6]. Kontrowersje związane z wybory prezydenckimi z 2016 roku na tyle spopularyzowały frazę "fake news", że stała się ona słowem roku 2017 w słowniku Collins'a [7].

To jak bardzo szkodliwy wpływ ma rozpowszechnianie nieprawdziwych treści w ujęciu jednostkowym oraz całego społeczeństwa sprawiło, że rosnącym zainteresowaniem wśród naukowców cieszy się zagadnienie automatycznego wykrywania fałszywych wiadomości, szczególnie w początkowej fazie ich propagacji.

1.1. Definicja fake newsów

Najczęściej spotykana definicja *fake newsa* stanowi, że jest informacja, która intencjonalnie zostaje stworzona w taki sposób, aby zawierała nieprawdziwe stwierdzenia.[8] Mogą być to jedynie fragmenty zawierające nieścisłości, półprawdy lub plotki, mające na celu

zmanipulowanie opinii czytelnika w określonych przez autora celach, np. politycznych [9].

Według definicji zaprezentowanej w artykule [10], aby daną wiadomość określić mianem fake newsa musi zawierać jeden z trzech czynników: 1) tekst, który intencjonalnie przeczy faktom, 2) obrazy niezwiązane z tematem lub 3) w celowy sposób zmanipulowane. Jak widać, wystarczy aby dana informacja była tylko fragmentarycznie nieprawdziwa, aby zyskała miano fake newsa.

1.2. Zdefiniowanie problemu

W niniejszej pracy zostanie zbadana możliwość klasyfikacji wiadomości na te prawdziwe oraz fake newsy, w szczególności we wczesnej fazie ich rozprzestrzeniania. Analizowanym medium, w którym udostępniane są informacje jest portal społecznościowy Twitter.

Detekcja fałszywych wiadomości będzie się odbywać na podstawie cech, które można wyodrębnić ze wzorców propagacji komunikatów, informacji zawartych w profilach użytkowników oraz bazując na właściwościach sieci społecznościowej w ujęciu lokalnym - pojedynczych węzłów (użytkowników), oraz globalnym - grafu (połączeń pomiędzy użytkownikami). Przetestowane zostaną dwa podejścia: jedno opierające się na klasycznych algorytmach nadzorowanego uczenia maszynowego typu las losowy, maszyna wektorów nośnych czy k najbliższych sąsiadów oraz drugie wykorzystujące grafowe sieci neuronowe. Rozważanym problemem uczenia maszynowego jest klasyfikacja binarna.

Tekst komunikatów nie jest przedmiotem badań, ponieważ fake newsy są tworzone w taki sposób, aby jak najdokładniej przypominać prawdziwe wiadomości, szczególnie teraz, w dobie ogólnego dostępu do zaawansowanych modeli językowych (ang. large language models) takich jak na przykład ChatGPT. Sprawia to, że wykrywanie fałszywych informacji bazując na tekście udostępnianych wiadomości jest nietrywialnym zadaniem. Wykorzystanie atrybutów pozyskanych z kaskad komunikacyjnych zostało już wcześniej zbadane i udowodniono, że fake newsy propagują inaczej niż prawdziwe wiadomości w mediach społecznościowych [11].

Aby sprawdzić możliwości detekcji fake newsów na jak najwcześniejszym etapie ich rozprzestrzeniania, grafy odzwierciedlające propagację informacji w medium społecznościowym zostały ograniczone na dwa sposoby: ilościowo - do pierwszych 5, 10, 20, 30, 40, 50 i 100 komunikatów oraz czasowo - dla pierwszych 5 i 30 minut oraz 1, 2, 4, 8, 12 i 24 godzin propagacji.

1.3. Analiza literatury

W artykule [9] został przedstawiony podział metod detekcji fałszywych komunikatów w mediach społecznościowych w zależności od analizowanego aspektu: tekstu wiadomości, kontekstu społecznego oraz struktury propagacji.

Wykrywanie fake newsów bazując jedynie na analizie tekstu jest szeroko badanym zagadnieniem, jednak posiada również pewne ograniczenia, takie jak na przykład brak uniwersalności pomiędzy różnymi językami. Przykładowe kategorie atrybutów oraz rodzaje algorytmów, które mogą zostać wykorzystane do klasyfikacji wiadomości na prawdziwe oraz fałszywe zostały zaprezentowane w artykule [12].

Modele zbudowane w oparciu o kontekst społeczny wykorzystują cechy wynikające przykładowo z danych demograficznych użytkowników, ich zaangażowania w korzystanie z mediów społecznościowych przedstawione w postaci propagacji treści w czasie lub łączących ich relacji (struktura sieci społecznościowej) [9] [12].

Atrybuty związane ze strukturą rozprzestrzeniania się komunikatów w sieci społecznościowej są często łączone z tymi wynikającymi z kontekstu społecznego, które zostały przedstawione w poprzednim akapicie. Jest to szczególnie ciekawy podtyp atrybutów biorąc pod uwagę badania [11], w których wykazano, że wzorce propagacji prawdziwych i fałszywych informacji w sieciach społecznościowych różnią się między sobą. Detekcja fałszywych wiadomości w oparciu o te cechy mogłaby być odporna na przykład na manipulacje, ponieważ kontrolowanie propagacji komunikatów jest poza zasięgiem możliwości pojedynczych jednostek w sieci społecznościowej [9]. Dodatkowo, te atrybuty są pozbawione ograniczeń, które występują w przypadku analizy tekstu komunikatów.

Do detekcji fałszywych wiadomości mogą zostać wykorzystane różne architektury, bazujące na sieciach neuronowych lub też nie. Klasyczne algorytmy takie jak drzewo decyzyjne, gaussowski naiwny klasyfikator Bayesa, regresja logistyczna oraz las losowy zostały użyte w artykule [13], w którym do detekcji fałszywych wiadomości były brane pod uwagę wszystkie trzy kategorie cech: tekstowe, strukturalne oraz temporalne (wynikające z kontekstu społecznego). Maszynę wektorów nośnych użyto w artykule [14], w którym detekcja fake newsów opierała się na cechach wynikających z profili użytkowników oraz treści komunikatów. Algorytmy k najbliższych sąsiadów, Adaboost oraz głosowanie większościowe zostały użyte w artykule [15] do klasyfikacji komunikatów na prawdziwe oraz fałszywe na podstawie atrybutów tekstowych. Generalizację stosową wykorzystali z kolei autorzy analizy [16].

Najnowsze badania skupiają się jednak na wykorzystaniu grafowych sieci neuronowych oraz atrybutów wyodrębnionych na podstawie badań zaangażowania użytkowników w czasie oraz struktury propagacji komunikatów. Cechy wynikających z analizy profili użytkowników posłużyły do zbudowania modelu opartego o grafowe sieci neuronowe i użytego do detekcji fałszywych wiadomości [17]. Architektura została stworzona w oparciu o algorytmy GraphSAGE oraz Diffpool. W artykule [9] został wykorzystany algorytm

grafowych konwolucyjnych sieci neuronowych (GCNConv) trenowany na danych zawierających cechy wynikające z analizy tekstu, propagacji komunikatów oraz kontekstu społecznego. Hybrydowy model uczenia maszynowego został zaproponowany w pracy [18], który składa się z grafowej sieci neuronowej (GCNConv) zbudowanej w oparciu o cechy propagacji wiadomości oraz dwukierunkowego enkodera, którego zadaniem jest analiza tekstu. Grafowe sieci neuronowe (GATConv) działające w oparciu o mechanizm atencji zostały wykorzystane w badaniu [19] dotyczącym multimodalnej detekcji fake newsów.

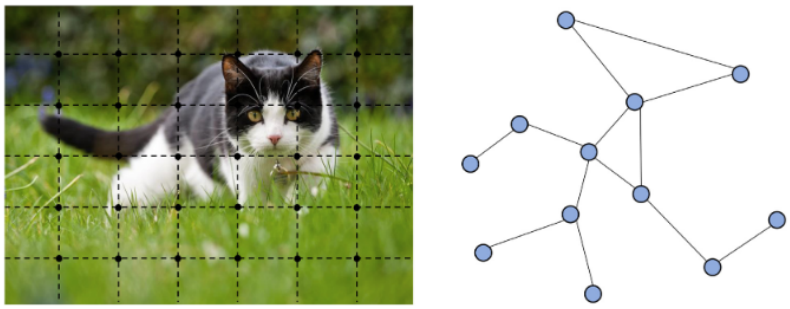
2. Metodologia

W tym rozdziale zostaną opisane algorytmy uczenia maszynowego wprowadzone w analizie literatury, w podziale na grafowe sieci neuronowe oraz algorytmy klasycznie, niebazujące na sieciach neuronowych. Dodatkowo wyjaśnione zostaną wykorzystane w tej pracy miary ewaluacji błędów.

2.1. Grafowe sieci neuronowe

Grafy to struktury danych opisujące zestaw obiektów (wierzchołków) oraz relacji, które je ze sobą łączą (krawędzi) [20]. Ich reprezentacje znajdują się w domenie nieeuklidesowej i mogą zawierać złożone relacje i współzależności pomiędzy obiektami [21].

Grafy są szczególnym typem danych wykorzystywanych w uczeniu maszynowym. Każda próbka może różnić się między sobą ilością wierzchołków oraz rozkładem krawędzi. Węzły w grafie mogą mieć więcej niż jedno krawędź, a co za tym idzie zmienną liczbę połączonych z nimi sąsiadów. Z tego względu jest to jeden z trudniejszych typów danych wykorzystywanych w uczeniu maszynowym. To wszystko sprawia, że operacje, które są proste do policzenia w domenie euklidesowej, są trudne do zastosowania w przestrzeni grafowej. Wiele algorytmów uczenia maszynowego działa dzięki założeniu niezależności pomiędzy poszczególnymi obiektami. Z kolei w grafach każdy wierzchołek może mieć jedną lub więcej krawędzi łączących go z innymi węzłami [21]. Grafowe sieci neuronowe



Rysunek 2.1. Po lewej obraz w przestrzeni euklidesowej, po prawej graf w przestrzeni nieeuklidesowej [21].

zostały po raz pierwszy zaprezentowane w artykule [22]. Autorzy podjęli pracę nad tym zagadnieniem motywowani faktem, że w wielu dziedzinach nauki relacje pomiędzy danymi można jedynie przedstawić za pomocą grafów. Zaproponowany przez nich model bazuje na sieciach neuronowych, ale poszerza ich zakres zastosowań o dane wejściowe w postaci grafowej, które są następnie przetwarzane do reprezentacji w przestrzeni euklidesowej.[21] Aktualnie uznaje się, że zaprezentowane w tych badaniach rozwiązania należą do przestarzałej kategorii rekurencyjnych grafowych sieci neuronowych. Ich podstawowym mechanizmem działania jest dyfuzja informacji. Stany węzłów w zaproponowanym rozwiązaniu aktualizowane są poprzez iteracyjną wymianę informacji o ich sąsiedztwie, aż

do osiągnięcia stanu równowagi. Taki sposób uczenia jest kosztowny obliczeniowo, ale stanowił inspirację do kolejnych badań, w szczególności pomysł przekazywania wiadomości pomiędzy węzłami, który został wykorzystany w późniejszych architekturach [21].

Wraz z rozwojem uczenia maszynowego zwłaszcza głębokiego oraz wprowadzeniem konwolucyjnych sieci neuronowych (ang. convolutional neural networks, w skrócie CNN) nastąpił rozwój w obszarze grafowych sieci neuronowych, pozwalający na przezwyciężenie dotychczasowych ograniczeń. Dzięki rozwiązaniom zaproponowanym w architekturze CNN oraz wektorowym reprezentacjom dystrybucyjnym (ang. embedding) grafów, powstały o wiele bardziej złożone, nowe warianty architektur grafowych [20].

Skrótowo proces uczenia grafowych sieci neuronowych do zadania rozważanego w tej pracy, czyli klasyfikacji można podzielić na następujące etapy: pierwszy krok polega na wygenerowaniu wektorowych reprezentacji węzłów (ang. node embeddings) poprzez iteracyjne przekazywanie wiadomości pomiędzy grupą węzłów w sieci, najczęściej znajdujących się w bezpośrednim sąsiedztwie rozważanego węzła. Następnie uzyskane reprezentacje wierzchołków służą do stworzenia nowej, z reguły mniejszej reprezentacji grafu. W ostatnim kroku nowo uzyskana reprezentacja sieci jest wejściem do zespołu warstw odpowiedzialnych za klasyfikację.[18]

2.1.1. Algorytm Graph Convolutional Operator - GCNConv

GCNConv jest algorytmem, który działa w oparciu o mechanizm splotu w grafowych sieciach neuronowych. Jest to operacja zaczerpnięta z konwolucyjnych sieci neuronowych, ale dostosowana do działania na grafach. Dzięki operacji splotu możliwe jest przygotowanie nowych reprezentacji wierzchołków grafu [23]. W ujęciu pojedynczego węzła następuje to poprzez aktualizowanie wartości jego cech w oparciu o własne atrybuty oraz zagregowane cechy sąsiednich węzłów. Daje to możliwość "nauczenia się"takich cech, które obrazują relacje oraz wzorce znajdujące się w lokalnym sąsiedztwie wierzchołka [18]. Ten proces jest nazywany propagacją komunikatów.

Zaproponowane rozwiązanie pozwala na wydajne przetwarzanie nawet bardzo dużych grafów, ponieważ model skaluje się liniowo wraz z liczbą krawędzi w grafie. Reguła przekazywania komunikatów w ujęciu całej warstwy jest zdefiniowana w następujący sposób:

$$\begin{aligned} H^{(l+1)} &= \left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \\ \tilde{A} &= A + I_N \end{aligned} \tag{1}$$

gdzie:

$H^{(l)}$ - wejście do warstwy $l+1$, czyli wyjście z warstwy l. Wejście do pierwszej warstwy jest zdefiniowane następująco, $H^{(0)} = X$, gdzie X jest to macierz cech węzłów dla oryginalnego grafu,

$\tilde{A}$ - macierz sąsiedztwa A z dodanymi połączeniami własnymi (ang. self-connections),

I_N - macierz jednostkowa,

$W^{(l)}$ - to macierz wag dla warstwy l, której wartości są uczone podczas treningu sieci,

$\tilde{D}$ - diagonalna macierz stopni odpowiadająca za normalizację wierzchołków o dużych stopniach w celu uniknięcia niestabilności numerycznych.

Macierze sąsiedztwa oznaczana symbolem A w większości przypadków składają się z zer oraz jedynek, gdzie element a_{ij} ma wartość 1 jeśli wierzchołki i oraz j są ze sobą połączone. W przeciwnym wypadku, element a_{ij} ma wartość równą 0. W przypadku tego algorytmu, macierz sąsiedztwa może zawierać wartości spoza tego zakresu, a dokładniej reprezentować wagi krawędzi.

2.1.2. Algorytm Graph Attentional Operator - GATConv

Algorytm GATConv został po raz pierwszy opisany w artykule [24] i działa w oparciu o mechanizmy splotu oraz uwagi (ang. attention). Algorytm rozszerza wcześniej zaproponowane architektury oparte o operacje konwolucji dodając mechanizm samo-uwagi (ang. self-attention). Ten mechanizm pozwala na przypisywanie indywidualnych wag (współczynników uwagi) węzłom znajdującym się w bezpośrednim sąsiedztwie rozważanego wierzchołka podczas tworzenia jego nowej reprezentacji. Wagi obliczane są w oparciu o atrybuty tych węzłów. Następnie cechy sąsiadów rozważanego wierzchołka są ze sobą sumowane w sposób ważony.

Daje to możliwość selektywnego agregowania informacji, w przeciwieństwie do algorytmu GCNConv, w którym wszystkie sąsiadujące wierzchołki są traktowane w ten sam sposób.

Znormalizowane współczynniki atencji dla poszczególnych wierzchołków są obliczane w następujący sposób:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{ik})} \tag{2}$$

gdzie:

e_{ij} - współczynnik określający jak ważne są cechy wierzchołka j z perspektywy węzła i,

N - sąsiedztwo wierzchołka i.

Następnie znormalizowane współczynniki uwagi są wykorzystywane do obliczenia wartości cech wyjściowych dla każdego węzła:

$$h'_i = \sigma\left(\sum_{j \in N_i} \alpha_{ij} W h_j\right) \tag{3}$$

gdzie:

h - reprezentacja wierzchołka wygenerowana w poprzedniej iteracji,

h' - nowa reprezentacja wierzchołka,

W - macierz wag, której wartości są uczone podczas treningu sieci,

$\sigma(\cdot)$ - nieliniowa funkcja aktywacji.

2.1.3. Algorytm GraphSAGE Operator

Algorytm GraphSAGE (skrót od: SAmple and aggreGatE) został zaprezentowany w artykule [25]. Opracowano go jako rozszerzenie grafowych konwolucyjnych sieci neuronowych.

W GCNConv nowa reprezentacja każdego węzła jest generowana na podstawie wszystkich jego sąsiadów. W algorytmie GraphSage wykorzystywane jest próbkowanie pozwalające na wyznaczenie mniejszej liczby rozważanych sąsiadów dla każdego wierzchołka. Koncepcja stojąca za tym podejściem to nauczenie się jak skutecznie łączyć informacje o cechach wybranych węzłów z lokalnego sąsiedztwa. Funkcjami, które mogą być użyte do tego zadania są: średnia, łączenie (ang. pooling) czy też LSTM. Zagregowane w ten sposób atrybuty sąsiadów są wykorzystywane do aktualizacji reprezentacji rozważanego wierzchołka.

Algorytm generujący reprezentacje węzłów działa w sposób iteracyjny, początkowo agregowane są informacje o bliskim sąsiedztwie, ale wraz z kolejnymi iteracjami wierzchołki zdobywają wiedzę o coraz bardziej oddalonych częściach grafu.

$$\begin{aligned} h_v^k &= \sigma\left(W \cdot f_k\left(h_v^{(k-1)}, h_{N(v)}^{k-1}\right)\right) \\ h_{N(v)}^{k-1} &= h_u^{k-1}, \forall c \in N(v) \end{aligned} \tag{4}$$

gdzie:

h_v^k - reprezentacja wierzchołka w iteracji k,

$\sigma(\cdot)$ - nieliniowa funkcja aktywacji,

W - macierz wag,

f_k - funkcja agregująca np. średnia, suma,

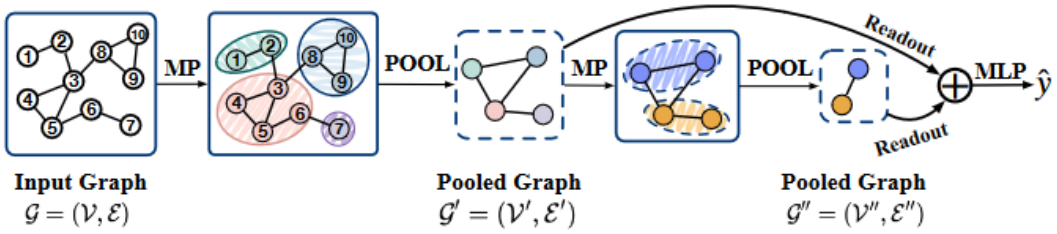
$h_{N(v)}^{k-1}$ - wektor zawierający zagregowane reprezentacje wierzchołków wybranych z sąsiedztwa węzła v,

$h_v^{(k-1)}$ - reprezentacja wierzchołka wygenerowana w poprzedniej iteracji [25].

Rysunek 2.2. Reprezentacja jak działa próbkowanie i agregacja w algorytmie GraphSage [25].

2.1.4. Algorytm łączący (ang. pooling)

W zadaniu klasyfikacji całych grafów należy wygenerować reprezentacje nie pojedynczych wierzchołków, ale całej sieci. W celu zmniejszenia złożoności struktury grafu można wykorzystać algorytm łączący (ang. pooling) oraz algorytm odczytu (ang. readout). Omawiane rozwiązania zostały zaczerpnięte z konwolucyjnych sieci neuronowych (CNN), w których te operacje są stosowane w celu zmniejszenia rozdzielczości przetwarzanych obrazów. W przypadku grafowych sieci neuronowych redukcji ulega liczba wierzchołków [26].

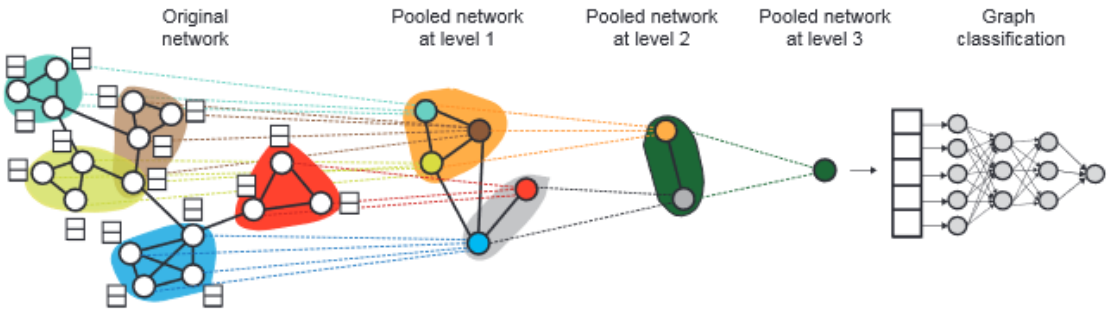


Rysunek 2.3. Rysunek reprezentujący działanie warstw łączących (ang. pooling). MP - algorytm propagacji wiadomości (ang. message passing), POOL - algorytm łączący [27]

Poniżej zostały zaprezentowane algorytmy łączące wykorzystane w pracy:

2.1.4.1. DiffPool Algorytm łączący DiffPool został zaprezentowany w artykule [28]. Pozwala na generowanie hierarchicznych reprezentacji grafów i stanowi swojego rodzaju odpowiednik operacji łączenia przestrzennego w konwolucyjnych sieciach neuronowych. Podstawową koncepcją stojącą za tym algorytmem jest nauczenie się w jaki sposób należy grupować wierzchołki w klastry w oparciu o ich reprezentacje wektorowe, które są następnie wykorzystywane do stworzenia reprezentacji całego grafu, będącego wejściem do kolejnej warstwy sieci neuronowej. Ten proces jest powtarzany aż do momentu, w którym pozostanie tylko jeden klaster [29]. Wyjściem z algorytmu jest hierarchia reprezentacji grafów, gdzie reprezentacje wyższych poziomów pozwalają na uchwycenie globalnych informacji, a każde kolejne uczą się bardziej wyrafinowanych reprezentacji wektorowych.

2.1.4.2. TopKPooling TopKPooling został opisany w artykule [30] pod nazwą gPool i jest stosowany w grafowych sieciach neuronowych. W wyniku tej operacji wybierane jest k najlepszych węzłów. Wybór tych wierzchołków jest dokonywany na podstawie uczącego się wektora projekcji p, który jest używany do obliczania punktów (ang. score) dla każdego węzła w grafie. Punkty przyznawane wierzchołkom można interpretować jako ilość informacji o węzłach, które mogą zostać zachowane. W kolejnym kroku przypisane wierzchołkom punkty są wykorzystywane do wyboru k najlepszych z nich, wraz z ich cechami i połączeniami. Wybrane węzły są zachowywane w grafie wyjściowym, a pozostałe są odrzucane [31].



Rysunek 2.4. Wysokopoziomowa reprezentacja działania algorytmu DiffPool [28]

2.2. Modele uczenia maszynowego niebazujące na sieciach neuronowych

Wszystkie przedstawione w tym podrozdziale algorytmy są metodami nadzorowanego uczenia maszynowego.

2.2.1. Drzewo decyzyjne

Drzewo decyzyjne jest metodą używaną zarówno w zadaniach regresyjnych jak i klasyfikacyjnych. Najczęściej są to drzewa binarne, gdzie z jednego węzła wychodzą dokładnie dwie gałęzie prowadzące do węzłów potomnych. W algorytmie drzew decyzyjnych można wyróżnić trzy podstawowe koncepty: węzły, gałęzie oraz liście. Próbki wejściowe propagowane są od korzeni do liści, w dół. Węzłami są wewnętrzne wierzchołki znajdujące się pomiędzy korzeniem a liśćmi drzewa, które stanowią reprezentację wyników kolejnych podziałów, dokonywanych na podstawie wartości atrybutów (cech) znajdujących się w zbiorze danych. Gałęzie stanowią pewnego rodzaju ścieżkę od węzła, w którym dokonywany jest podział do węzłów potomnych. W liściach, czyli węzłach ostatniego, najniższego poziomu, w zadaniu klasyfikacji, znajdują się etykiety klas lub rozkłady prawdopodobieństwa klas, które są nadawane próbkom podczas predykcji [32].

2.2.2. Gaussowski naiwny klasyfikator Bayesa

Gaussian naive Bayes classifier, czyli w wolnym tłumaczeniu naiwny klasyfikator bayesowski dla danych o rozkładzie normalnym lub gaussowski naiwny klasyfikator Bayesa, jest rozszerzeniem klasycznego algorytmu naiwnego klasyfikatora Bayesa o dodatkowe założenie dotyczące rozkładu wartości ciągłych cech, a dokładniej przyjmuje hipotezę, że mają one rozkład normalny. Ten algorytm jest wykorzystywany w zadaniu klasyfikacji (binarnej lub wieloklasowej).

Założeniem w oparciu o które działa ten algorytm jest niezależność cech w obrębie jednej klasy, a dokładniej dla konkretnej próbki wartość jednego atrybutu jest niezależna od pozostałych. Taka hipoteza jest jednak rzadko prawdziwa w realnych zastosowaniach [33].

Metoda działania algorytmu jest następująca: Obliczane są prawdopodobieństwa a priori dla wszystkich etykiet poprzez podzielenie częstości występowań danej klasy przez

liczbę wszystkich próbek w zbiorze treningowym. Przy użyciu funkcji gęstości prawdopodobieństwa (ang. probability density function) obliczane jest prawdopodobieństwo z jakim dana wartość cechy wystąpi w danej klasie. Do tego celu wykorzystywane jest odchylenie standardowe i średnia wartość atrybutu dla rozważanej etykiety [34]. Następnie obliczane jest prawdopodobieństwo a posteriori na podstawie wzoru z rachunku prawdopodobieństwa przedstawionego przez Thomasa Bayesa [32]. Wynik określa z jakim prawdopodobieństwem dana próbka charakteryzująca się konkretnymi wartościami atrybutów należy do pewnej klasy. Na końcu wybierana jest klasa, dla której prawdopodobieństwo a posteriori ma najwyższą wartość.

2.2.3. Regresja logistyczna

Regresja logistyczna jest algorytmem używanym w problemie klasyfikacji. Dla zadania binarnego wyjściem z funkcji jest prawdopodobieństwo, z jakim rozważana próbka należy do klasy pozytywnej. Reprezentacją wewnętrzną jest funkcja liniowa, będąca kombinacją cech wejściowych, zdefiniowana w następujący sposób:

$$f = w^T \cdot x + b \tag{5}$$

gdzie:

w - wektor wag,

x - wektor cech wejściowych danej próbki,

b - wyraz wolny (ang. bias).

W trakcie treningu model uczy się optymalnych wartości wag oraz wyrazu wolnego starając się zminimalizować funkcję kosztu.

Funkcja logistyczna, pozwalająca na przekształcenie wyjścia z funkcji liniowej do prawdopodobieństwa, jest zdefiniowana w następujący sposób:

$$\Pi(x) = P(1|x) = \frac{1}{1 + \exp\left(-f\right)} \tag{6}$$

Jeżeli prawdopodobieństwo jest większe lub równe 0.5, to model zwraca klasę pozytywną, w przeciwnym wypadku zwracana jest klasa negatywna. [33] [32]

2.2.4. Maszyna wektorów nośnych

Algorytm SVM (ang. support vector machine) może być wykorzystany w problemach klasyfikacji binarnej oraz wieloklasowej, jak również regresji. Jego celem jest określenie płaszczyzny, która pozwala na rozdzielenie próbek należących do różnych klas z jak największym marginesem, czyli położonej jak najdalej od analizowanych przykładów. Margines jest wyznaczany jako minimalna odległość od próbek danej klasy do płaszczyzny. Ten sposób pozwala na zmniejszenie szansy na nadmierne dopasowanie modelu do danych. Algorytm dopuszcza jednak tzw. "miękki margines", czyli możliwość występowania

próbek, których nie da się poprawnie sklasyfikować. Przykłady ze zbioru treningowego, które położone są na marginesie, czyli najbliżej płaszczyzny nazywane są wektorami nośnymi. W przypadkach, w których nie jest możliwe liniowe odseparowanie klas używana jest tzw. sztuczka jądrowa (ang. kernel trick). Sztuczka jądrowa pozwala na odnalezienie liniowej płaszczyzny separującej, ale w wyższym wymiarze przestrzeni cech, do której mapowany jest zbiór treningowy [32] [33].

2.2.5. k Najbliższych sąsiadów

Algorytm k najbliższych sąsiadów, w skrócie k-NN, może być stosowany zarówno w problemach regresji jak i klasyfikacji. W zagadnieniu klasyfikacji algorytm dla nowej, wcześniej niewidzianej próbki przygotowuje jej reprezentację w przestrzeni cech, a następnie wyszukuje jej k najbliższych sąsiadów ze zbioru treningowego w przestrzeni cech. Liczba k jest ustalana przez użytkownika, i dla klasyfikacji binarnej najczęściej ma wartość nieparzystą, ponieważ przypisanie etykiety dla nowej próbki odbywa się w najpowszechniejszym przypadku przez głosowanie większościowe. Odległość między próbkami może być mierzona na różne sposoby, używając na przykład odległości euklidesowej lub metryki miejskiej (ang. Manhattan distance) [33] [35].

2.2.6. Las losowy

Las losowy to algorytm zespołowy, który również jest wykorzystywany w dwóch różnych zadaniach uczenia maszynowego: regresyjnych oraz klasyfikacyjnych. Jego podstawą są modele bazowe, czyli proste drzewa decyzyjne oraz technika zwana baggingiem (skrót od bootstrap aggregation). Każdy kolejny model bazowy jest tworzony na podstawie zbioru danych wylosowanego w próbie bootstrapowej, w której losowanych jest n elementów ze zbioru treningowego o rozmiarze n. Jest to losowanie ze zwracaniem, czyli w zbiorze dla konkretnego drzewa decyzyjnego mogą znajdować się powtarzające się przykłady. To pozwala na wprowadzenie zróżnicowania pomiędzy kolejnymi modelami bazowymi, zmniejszając ryzyko zbyt dużego dopasowania modelu do danych treningowych. Wszystkie drzewa tworzone są w tym samym czasie. W algorytmie lasu losowego każde drzewo decyzyjne głosuje na klasę, która ma zostać zwrócona przez model. Wykorzystywane jest głosowanie większościowe [32].

2.2.7. AdaBoost

AdaBoost (ang. Adaptive Boosting) jest algorytmem zespołowym, opartym na technice boostingu, której założeniem jest iteracyjna budowa wielu prostych modeli bazowych [36]. Proces tworzenia modelu wyjściowego jest sekwencyjny, poprzez dodawanie i trenowanie kolejnych, nowych modeli bazowych. Najczęściej rolę modeli bazowych pełnią bardzo małe drzewa decyzyjne, składające się z dwóch poziomów - korzenia, zawierającego cechę na podstawie której nastąpi podział oraz liści. Każde kolejne drzewo decyzyjne jest tworzone w taki sposób, aby korygować błędy poprzednika, w tym celu próbkom w

zbiorze danych nadawane są wagi. Im wyższa waga, tym większy nacisk na próbkę, dla której wcześniejsze modele bazowe popełniły błędy.

2.3. Zespołowe metody uczenia maszynowego

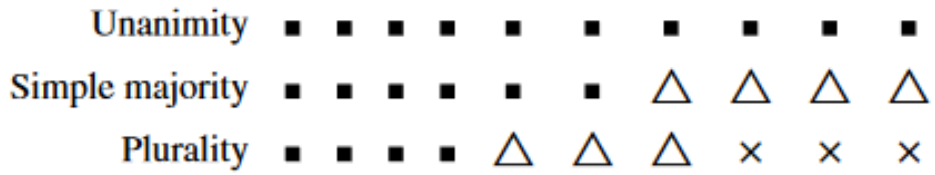
Metody uczenia zespołowego często pozwalają na osiągnięcie lepszych wyników czy to w przypadku klasyfikacji czy zadania regresyjnego, ponieważ łącząc wiele modeli, pozwalają na przezwyciężanie ograniczeń wynikających z użycia pojedynczego modelu.

2.3.1. Głosowanie większościowe

Głosowanie większościowe (ang. majority voting) to algorytm zespołowego uczenia maszynowego, łączący wyjścia z wielu modeli. Może być użyty zarówno w zagadnieniach regresyjnych jak i klasyfikacyjnych. Taki zespół może zostać stworzony z dowolnych dwóch lub więcej istniejących, wytrenowanych modeli uczenia maszynowego. Pozwala na osiągnięcie potencjalnie lepszych wyników niż przy użyciu pojedynczego modelu wchodzącego w skład zespołu.

W zastosowaniu do klasyfikacji wyróżnia się dwa podejścia, głosowanie twarde (ang. hard voting) oraz miękkie (ang. soft voting). Pierwsze z nich polega na zwracaniu klasy, na którą zostało oddane najwięcej głosów przez modele wchodzące w skład zespołu. W głosowaniu miękkim pojedyncze modele zwracają rozkład prawdopodobieństwa dla możliwych klas. Następnie prawdopodobieństwa są sumowane dla każdej etykiety i zwracana jest klasa z największym zagregowanym prawdopodobieństwem.

Wyróżniane są trzy wzorce konsensusu: jednomyślność (ang. unanimity), zwykła większość głosów (ang. simple majority) czyli przypadek, w którym przewaga danej klasy następuje jednym głosem (50% + 1), oraz względna większość głosów (ang. plurarity) [37].



Rysunek 2.5. Trzy wzorce consensusu [37]. Kształty odpowiadają klasom, pojedyncza figura to indywidualny klasyfikator w zespole.

Dla nowej, nieznanej próbki każdy klasyfikator z zespołu o rozmiarze L przypisuje klasę s. Następnie liczona jest liczba głosów oddanych na każdą klasę ω_k, gdzie $k = 1, ..., c$.

$$P(k) = \sum_{i=1}^{L} I(s_i, \omega_k) \tag{7}$$

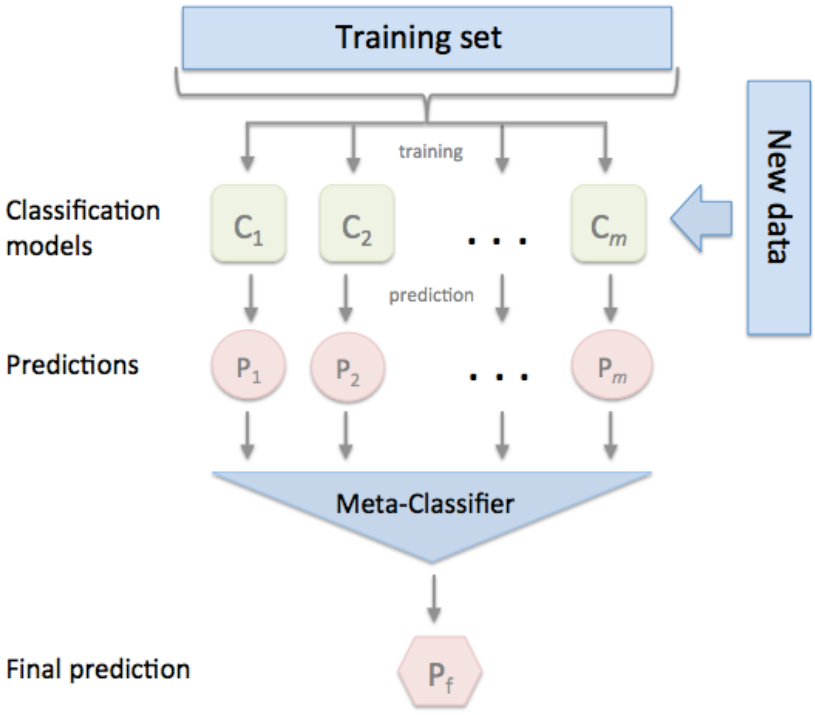
gdzie:
$I(a,b) = 1$ jeśli $a = b$, i 0 w przeciwnym wypadu.

Etykieta k^* jest przypisywana dla obiektu, na podstawie równania:

$$k^* = arg \max_{k=1}^{c} P(k) \tag{8}$$

2.3.2. Generalizacja stosowa

Generalizacja stosowa (ang. stacked generalization) jest konceptem wprowadzonym w artykule [38]. Jest to wariant zespołowego uczenia maszynowego, w którym można wyróżnić dwa hierarchicznie połączone ze sobą poziomy. Pierwszy składa się z m modeli bazowych, gdzie $m > 1$. W przeciwieństwie do technik takich jak boosting lub bagging, gdzie wszystkie modele bazowe są najczęściej tym samym typem architektury (np. drzewem decyzyjnym), w generalizacji stosowej są to zwykle różne, niepowtarzające się typy modeli klasyfikacyjnych, takie jak np. drzewo decyzyjne, regresja logistyczna itd. Nie ma ograniczeń nałożonych na to, jakie modele mogą zostać wykorzystane w tym zagadnieniu. Wszystkie modele bazowe uczone są na dokładnie tym samym zbiorze danych. W drugim poziomie znajduje się finalny model, który dokonuje ostatecznej predykcji. Wejściem do tego modelu są połączone predykcje dokonane przez modele bazowe. Taki sposób łączenia modeli pozwala na lepsze radzenie sobie z ograniczeniami wynikającymi ze stosowania pojedynczych architektur wykorzystanych w modelach bazowych [38].



Rysunek 2.6. Blokowa reprezentacja klasyfikacji stosowej [39].

Dane do treningu są dzielone na partie przy użyciu skrośnej k-krotnej walidacji (ang. k cross-validation), która dzieli zbiór na k partii, gdzie $k-1$ partii jest użytych do uczenia, a jedna do walidacji. Finalny model otrzymuje wyniki predykcji dokonane na zestawie

danych walidacyjnych. Pary predykcja i poprawna klasa stanowią wejście do tego modelu, które może być dodatkowo wzbogacone o wybrane cechy znajdujące się w zbiorze danych, w celu dostarczenia kontekstu. [40]

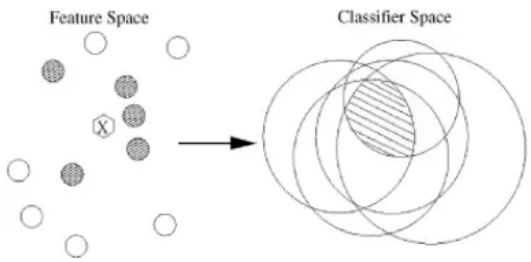
2.3.3. Dynamiczne dobieranie zespołu

Algorytm dynamicznego doboru zespołu (ang. dynamic ensemble selection), a dokładniej KNORA (k-Nearest Neighbor Oracle) został opisany w artykule [41]. Jest to metoda, w której automatycznie wybierany jest podzbiór elementów zespołu w momencie, w którym należy przeprowadzić predykcję.

Na początku wszystkie modele wchodzące w skład zespołu są uczone na zbiorze treningowym. Następnie jest wybierany ich podzbiór, który powinien zapewniać najlepszą wydajność predykcji dla konkretnej, nowej próbki, w oparciu o zestaw wartości jej cech. Dla dowolnego elementu zbioru testowego określa się jego K najbliższych sąsiadów w przestrzeni cech zbioru walidacyjnego. W kolejnym kroku wybierane są klasyfikatory, które dokonały poprawnej predykcji na tych sąsiadujących próbkach i tworzy się z nich zespół do klasyfikacji analizowanego przykładu.

Metoda jest dynamiczna ponieważ próbki mogą mieć różną charakterystykę i wymagać różnych kombinacji klasyfikatorów bazowych. Autorzy opisują 4 rodzaje algorytmu KNORA: KNORA-Eliminate, KNORA-Union, KNORA-Eliminate-W, KNORA-Union-W, przy czym dwa ostatnie różnią się od dwóch pierwszych tylko dodatkowymi wagami, które obliczane są na podstawie odległości euklidesowej między sąsiadem próbki, a przykładem testowym.

KNORA-Eliminate jest algorytmem, w którym wybierane są wszystkie klasyfikatory, będące w stanie poprawnie przewidzieć klasę wszystkich k najbliższych przykładów w sąsiedztwie próbki testowej X. Jeśli ani jeden klasyfikator nie jest w stanie osiągnąć 100% poprawności wyników, rozmiar branego pod uwagę sąsiedztwa jest zmniejszany o 1.

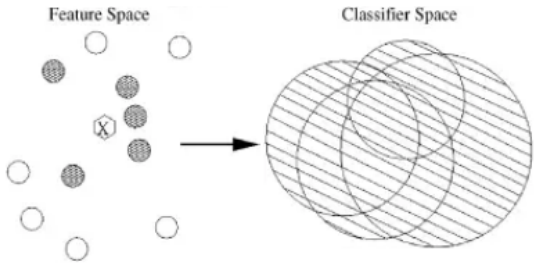


Rysunek 2.7. Schematyczna reprezentacja algorytmu KNORA-E. Próbka testowa jest oznaczona sześciokątem, zaciemnione próbki to najbliższe sąsiedztwo. Po prawej stronie jest zobrazowane poprzez zaciemnienie przecięcie w przestrzeni klasyfikatorów, które mogą zostać wykorzystane w zespole dla tej konkretnej próbki.[41]

Proces ten powtarza się aż do momentu gdy jeden lub więcej modeli osiągają wymaganą dokładność, a następnie tworzony jest z nich zespół wykorzystywany do predykcji

dla nowego przykładu. Ostateczny wynik jest zwracany przy użyciu głosowania większościowego [41].

KNORA-Union, w przeciwieństwie do *KNORA-Eliminate*, jest algorytmem w którym zespół tworzony jest ze wszystkich klasyfikatorów, które dokonują choć jednej poprawnej predykcji dla sąsiadów próbki testowej X. Dla każdego klasyfikatora w zespole jest przypisana waga proporcjonalna do liczby prawidłowych klasyfikacji sąsiadów. Ostateczny wynik jest obliczany na podstawie głosowania klasyfikatorów [41].



Rysunek 2.8. Schematyczna reprezentacja algorytmu KNORA-U. Próbka testowa jest oznaczona sześciokątem, zaciemnione próbki to najbliższe sąsiedztwo. Po prawej stronie zobrazowano poprzez zaciemnienie przecięcie w przestrzeni klasyfikatorów, które mogą zostać wykorzystane w zespole dla tej konkretnej próbki [41].

2.4. Miary ewaluacji błędów

Macierz pomyłek (ang. confusion matrix) jest tabelą, która reprezentuje rozkład przewidzianych klas w analizowanym zbiorze danych. Jest często używana do oceny jak wytrenowany model radzi sobie z predykcją w podziale na konkretne etykiety.

W klasyfikacji binarnej stosowane najczęściej miary ewaluacji błędów opierają się na macierzy pomyłek, w której znajdują się dwie kolumny i dwa wiersze, gdzie wiersze odnoszą się do dokonanych predykcji, a kolumny do klas rzeczywistych. Na rysunku 2.9 została przedstawiona przykładowa tablica pomyłek w której każdy możliwy element jest zdefiniowany opisowo [32].

		Actual	
		Positive	Negative
Predicted	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

Rysunek 2.9. Przykładowa macierz pomyłek [42].

Nazewnictwo opiera się na dwóch podstawowych zasadach - pierwszy człon (true/false) określa, czy dokonana przez model predykcja pokrywa się z klasą rzeczywistą

znajdującą się w zbiorze danych. Drugi człon nazwy (positive/negative) odnosi się odpowiednio do klasy pozytywnej lub negatywnej zwracanej przez model.[32]
Wyróżnia się następujące miary:

- *Prawdziwe pozytywna*, w skrócie TP (ang. True Positive) - jest to suma przypadków, w których model dokonuje poprawnej predykcji na klasie pozytywnej, czyli przewidywana jest klasa pozytywna i rozważany przykład faktycznie jest elementem tej klasy,
- *Prawdziwie negatywna*, w skrócie TN (ang. True Negative) - tak jak w przypadku miary True Positive, jest to suma przypadków, w których model dokonuje poprawnej predykcji, ale tym razem klasy negatywnej. Model zwraca klasę negatywną i badana próbka należy do tej klasy,
- *Fałszywie pozytywna*, w skrócie FP (ang. False Positive) - suma przypadków, w których dokonana predykcja jest niepoprawna. Model zwraca klasę pozytywną dla próbki, która w rzeczywistości jest elementem klasy negatywnej,
- *Fałszywie negatywna*, w skrócie FN (ang. False Negative) - tak jak w przypadku miary False Positive, to liczba przypadków, w których model dokonuje niepoprawnej klasyfikacji, a dokładniej przewiduje klasę negatywną, kiedy rzeczywistą etykietą danej próbki jest klasa pozytywna [43].

Na podstawie powyższych metryk można wyliczyć miary ewaluacji błędów takie jak: recall, precyzja [44], accuracy, f1-score [45]:

- **recall**, nazywany po polsku czułością, jest współczynnikiem obliczanym jako stosunek przykładów, które zostały poprawnie zaklasyfikowane jako klasa pozytywna do wszystkich próbek należących do klasy pozytywnej ze zbioru danych. Można tę miarę również opisać jako liczba próbek z klasy pozytywnej, dla których algorytm zwrócił poprawną etykietę [44].

$$\text{Recall} = \frac{TP}{TP + FN} \tag{9}$$

- **precyzja** (ang. precision) jest definiowana jako stosunek liczby przykładów dla których model dokonał poprawnej predykcji klasy pozytywnej, do wszystkich próbek którym model przypisał etykietę pozytywną. Innymi słowy jest to współczynnik mówiący o tym ile obserwacji zaklasyfikowanych jako pozytywne, faktycznie należy do tej klasy [44].

$$\text{Precision} = \frac{TP}{TP + FP} \tag{10}$$

- **Miara F1** (ang. F1-score) - jest to współczynnik agregujący dwie poprzednio opisane miary, czyli czułość oraz precyzję. Możliwe do uzyskania wartości znajdują się w zakresie od 0,0 do 1,0. Matematycznie miara F1 jest zdefiniowana jako średnia

harmoniczna:[44]

$$\text{F1-score} = \frac{1}{\frac{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}}{2}} = 2\frac{\text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}} \tag{11}$$

- **dokładność** (ang. accuracy) - współczynnik zdefiniowany jako stosunek liczby próbek, dla których model dokonał poprawnej predykcji, niezależnie od tego do której klasy należały do sumarycznej liczby wszystkich obserwacji dostępnych w zbiorze danych. Chcąc użyć tej miary należy wziąć pod uwagę fakt, że najlepiej działa na zbalansowanych danych [46].

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

3. Zbiór danych

3.1. Wymagania względem zbioru danych

Biorąc pod uwagę specyfikę rozważanego zadania badawczego, nie każdy zbiór danych stworzony z myślą o detekcji fake newsów może zostać wykorzystany.

Aby zestaw danych mógł potencjalnie zostać użyty w tym konkretnym zagadnieniu musi spełniać dokładnie określone wymogi. W takim zbiorze powinna znajdować się lub być możliwa do zrekonstruowania sieć użytkowników wymieniających pomiędzy sobą komunikaty. Kluczowe jest aby posiadał informacji o powiązaniach pomiędzy tymi osobami, tak aby móc wnioskować o strukturze sieci, czyli przykładowo informacje o tym, kto jest z kim połączony relacją obserwowania. Dodatkowym warunkiem jest aby zbiór danych zawierał cechy związane z tym jak propagowały wiadomości w sieci, a konkretnie kto dany komunikat stworzył, jaki jest znacznik czasowy tej wiadomości, podstawowe informacje o autorze oraz reakcje użytkowników na aktywność autora (kto dany post podał dalej, kiedy to nastąpiło oraz podstawowe informacje o użytkowniku udostępniającym oryginalny komunikat).

3.2. Pierwotne zagadnienie badawcze

Zbiorem danych, który został początkowo przeanalizowany jest struktura sieci społecznościowej polskiego portalu wykop.pl, spełniającego większość z przedstawionych wcześniej wymagań względem zestawu danych. Medium pozwala na tworzenie kont, obserwowanie użytkowników, dodawanie postów oraz komunikację poprzez komentowanie. Funkcjonalność jest bardzo podoba do tej, którą udostępnia Twitter.

Zbiór danych składał się z dwóch rodzajów informacji. Pierwsza dotyczyła użytkowników oraz połączeń pomiędzy nimi, co pozwalało na zbudowanie skierowanej sieci społecznościowej powiązań pomiędzy użytkownikami. Drugim dostępnym typem danych były informacje o udostępnianych postach oraz aktywności użytkowników w postaci komentarzy.

Pierwotnym celem badawczym dla zbioru danych z portalu Wykop było zbudowanie w jego oparciu modelu, który wiązałby dynamikę rozprzestrzeniania się wpisu ze strukturą sieci kontaktów użytkownika publikującego dany post.

Sieć społecznościowa łącząca konta relacją obserwowania została zrekonstruowana, a w oparciu o dane dotyczące aktywności użytkowników w postaci postów/komentarzy dokonano wstępnej analizy zagadnienia. Przeanalizowane zostały cechy typowo sieciowe, takie jak:

1. uśredniony współczynnik gronowania,
2. rozmiary klik występujących w grafie,
3. współczynnik asortatywności sieci,
4. rozkład stopni wejściowych i wyjściowych wierzchołków.

oraz atrybuty wynikające z łącznej analizy sieci społecznościowej oraz aktywności użytkowników w postaci:

1. zależności pomiędzy liczbą obserwujących, a liczbą uzyskiwanych komentarzy pod dodawanymi postami,
2. aktywności w komentarzach osób obserwujących konto udostępniające nowy wpis,
3. zależności między odwzajemnioną obserwacją a częstotliwością wzajemnego komentowania postów.

Niestety, w przeprowadzonych badaniach nie zauważono znamion związku popularności postów ze strukturą społeczności wokół udostępniających ich użytkowników. Takie wnioski doprowadziły do zmodyfikowania zagadnienia badawczego oraz analizowanego medium społecznościowego.

3.3. Medium społecznościowe

Rozważanym medium społecznościowym jest Twitter, obecnie znany pod nazwą X. Na potrzeby tej pracy, używana będzie nazwa przed zmianą. Portal daje możliwość publikowania treści, wchodzenia w interakcje z innymi użytkownikami oraz obserwowanie ulubionych twórców. Według danych z grudnia 2022 roku, Twitter ma miesięcznie 450 milionów aktywnych użytkowników [47].

Do podstawowych funkcjonalności portalu należą: możliwość tworzenia kont, budowanie sieci znajomych poprzez śledzenie wybranych użytkowników, dodawanie postów na swojej tablicy czyli tworzenie tzw. *tweetów*, udostępnianie postów innych twórców na swojej tablicy, które są nazywane *retweetami* oraz reagowanie na tweety lub retweety poprzez komentowanie i zostawianie polubień tzw. *like'ów*.

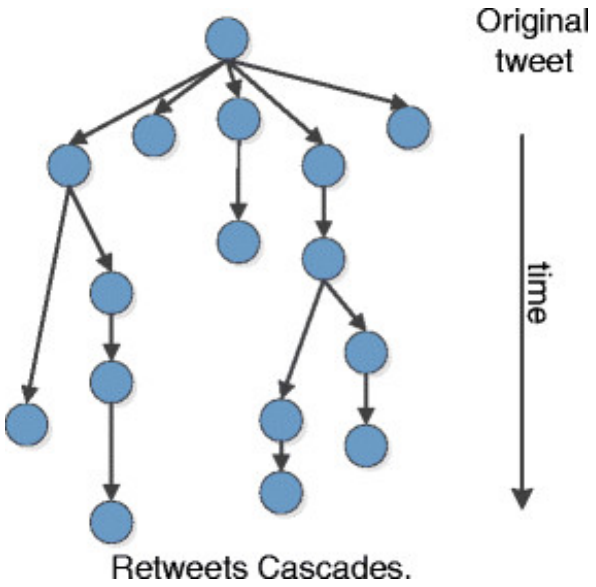
3.4. Struktura propagacji komunikatów

Kaskada komunikacyjna dla pojedynczego, rozpowszechnionego wśród użytkowników postu na Twitterze ma strukturę hierarchiczną, pozwalającą na częściową rekonstrukcję ścieżek, którymi dana informacja docierała do konkretnych odbiorców.

Główny węzeł na rysunku 3.1, przedstawionym poniżej, reprezentuje tweet, a pozostałe wierzchołki, które dołączane są do kaskady wraz z upływem czasu obrazują retweety oryginalnego posta. Strzałki obrazują przepływ informacji a dokładniej czyj post został podany dalej.

3.5. Pobieranie zbioru danych

W pracy zostało wykorzystane publicznie dostępne repozytorium *FakeNewsNet*, które opublikowano po raz pierwszy w artykule [3]. Zostało ono specjalnie przygotowane pod zagadnienie wykrywania fałszywych wiadomości na Twitterze. Zbiór stworzono w odpowiedzi na brak jednego zestawu danych zawierającego zarówno treść propagującego



Rysunek 3.1. Przykładowa kaskada komunikacyjna [48]. Węzeł znajdujący się na samej górze przedstawia oryginalnego tweeta. Wraz z upływem czasu (przedstawionym na osi po prawej) pojawiają się retweety oryginalnego posta oraz innych retweetów.

komunikatu, kontekst społeczny oraz informacje związane z tym jak wiadomości rozprzestrzeniały się w czasie. Możliwy do pobrania zestaw danych zawiera między innymi tweety oraz retweety dotyczące artykułów udostępnianych w następujących witrynach weryfikujących fakty: *GossipCop* (gossipcop.com) i *PolitiFact* (politifact.com).

W serwisie Politifact dziennikarze oraz eksperci powiązani z tą domeną oceniają artykuły dotyczące polityki a następnie stwierdzają, czy dany tekst zawiera prawdziwe czy fałszywe informacje [3].

Witryna internetowa Gossipcop zajmuje się innym obszarem, a dokładniej jest to strona na której można sprawdzić prawdziwość artykułów ze świata show-biznesu, zebranych z różnych źródeł w Internecie. System oceniania nie jest binarny tylko w postaci skali od 0 do 10, gdzie zero indykuje fałszywość danego artykułu a 10 jego prawdziwość. Artykuły z oceną mniejszą niż 5 uznawane są za nieprawdziwe, a z oceną wyższą niż 5 za zawierające fakty. Tak więc następuje przejście ze skali 0-10 do problemu binarnego, co zostało wykorzystane przez autorów repozytorium FakeNewsNet. Ze względu na duży procent wiadomości ocenionych jako nieprawdziwe, autorzy artykuły zebrali również teksty z portalu E! Online, który jest również witryną publikującą treści o tematyce rozrywkowej. Uznano, że wszystkie artykuły pochodzące z tego źródła można zaklasyfikować jako prawdziwe [3].

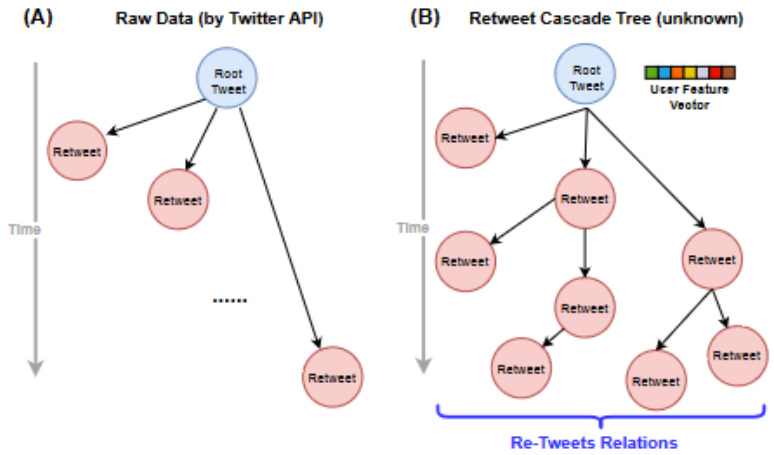
Zdecydowaną zaletą tego zbioru danych są wiarygodne etykiety z adnotacjami dziennikarzy i ekspertów zajmujących się tą tematyką na co dzień. Dodatkowo, ten zestaw danych spełnia wymagania przedstawione w podrozdziale 3.1 a dokładniej zawiera cechy związane z propagacją wiadomości, powiązaniami pomiędzy użytkownikami oraz informacje obrazujące interakcje użytkowników z tweetami i retweetami w czasie.

Zestaw danych FakeNewsNet nie jest publicznie dostępny w Internecie ze względu na politykę prywatności Twittera oraz prawa autorskie wydawców newsów. Repozytorium stworzone przez autorów artykułu [3] pozwala na samodzielne pobranie zbioru danych przy wykorzystaniu API Twittera [49]. Aby się z nim połączyć należy użyć własnego klucza dostępowego, który może zostać wygenerowany na stronie *developer.twitter.com*. W ten sposób uzyskane dane mogą służyć jedynie do własnego użytku i nie mogą być udostępniane innym osobom. Wykorzystując udostępniony kod można pobrać 7 typów cech: artykuły, tweety, retweety, profile użytkowników, 200 najnowszych tweetów danego użytkownika, osoby śledzące dane konto, oraz osoby, które ten użytkownik obserwuje.

Należy zwrócić uwagę na fakt, że każde uruchomienie kodu będzie skutkowało pobraniem innego zestawu danych, jako że Twitter jest "żywym"medium, gdzie cały czas publikowane są nowe wpisy, które zostają następnie podawane dalej, tworzone są nowe konta oraz usuwane stare. Zmiany mogą dotyczyć na przykład liczby retweetów poszczególnych tweetów, jak również profili użytkowników udostępniających treści w postaci dodania nowych kont do obserwowania lub zaprzestania relacji obserwowania.

3.6. Struktura pobranych danych

Bardzo istotnym rodzajem informacji z perspektywy badanego tematu jest struktura poszczególnych kaskad komunikacyjnych, czyli to w jakiej kolejności następują po sobie kolejne retweety tej samej wiadomości. Przykładową kaskadę, tak jak na rysunku 3.1, można przedstawić w sposób, w którym tweety i retweety są węzłami w sieci, a krawędzie reprezentują relację pomiędzy nimi. Można to uogólnić do następującej zasady: nowy węzeł m jest tworzony kiedy post n zostaje podany dalej (zretweetowany). Dodawana jest wtedy również krawędź pomiędzy węzłami n oraz m, skierowana w stronę nowego wierzchołka.

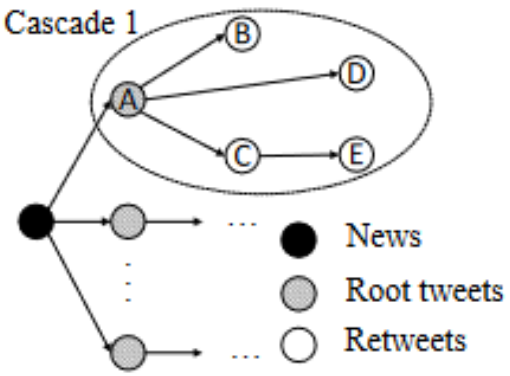


Rysunek 3.2. Po lewej stronie przedstawiona jest struktura danych po pobraniu ich z API Twittera, po prawej kaskada po rekonstrukcji [50].

Cechą charakterystyczną Twittera jest fakt, że zarówno tweet jak i retweet może być udostępniany dalej. Niestety, podczas pobierania danych z API rozważanego serwisu społecznościowego ten kontekst jest zatracany, ponieważ nie jest zwracana informacja o tym, co jest bezpośrednim źródłem danego retweeta. W uzyskanych danych nie znajdują się informacje pozwalające na zidentyfikowanie kto po kim udostępnił danego retweeta, ponieważ wszystkie retweety są przypisane do źródła tzn. oryginalnego tweeta, tak jak jest to pokazane na rysunku 3.2. Prowadzi to do tego, że na podstawie danych możliwych do pobrania z oficjalnego API Twittera nie można jednoznacznie zrekonstruować kaskad komunikacyjnych do ich oryginalnej formy.

3.7. Metody rekonstrukcji kaskad komunikacyjnych

Tak jak wspomniano, nie ma możliwości jednoznacznego odtworzenia ścieżek propagacji wiadomości, można jedynie podjąć próby jak najbardziej zbliżonego do oryginału odbudowania kaskad. Należy jednak pamiętać i brać pod uwagę fakt, że użytkownicy mogą dowiadywać się o konkretnych postach niebezpośrednio z Twittera, a na przykład widząc je udostępnione w innych mediach społecznościowych, przytaczane w internetowych serwisach informacyjnych lub otrzymując linki do nich od znajomych w aplikacjach do komunikacji typu Messenger. Te dodatkowe, zewnętrzne bodźce prowadzące do udostępnienia dalej danego tweeta komplikują zadanie rekonstrukcji kaskad, a dokładniej proces wyznaczania źródła retweeta. W wielu metodach odbudowy ścieżek propagacji komunikatów zakłada się, że jeżeli spośród kont, które obserwuje dany użytkownik któreś udostępni wcześniej rozważanego tweeta, to jest ono źródłem retweeta. Jednak w przypadku, w którym osoba dowiaduje się o danym poście niebezpośrednio z Twittera, to założenie przestaje być prawdziwe. To sprawia, że tym trudniejszym zadaniem jest jak najdokładniejsze odtworzenie kaskad komunikacyjnych.



Rysunek 3.3. Kaskady komunikacyjne połączone za pomocą głównego węzła [17]

Rekonstrukcja ścieżek propagacji jest jednak niezwykle ważnym etapem przygotowania danych do dalszych obliczeń. Jak najdokładniejsze odwzorowanie wyjściowego

grafu pozwoli na trenowanie modeli na danych jak najbardziej odpowiadających tym rzeczywistym. W literaturze występują różne algorytmy rekonstrukcyjne, w niniejszej pracy zbiory zostały przygotowane na dwa różne sposoby. W każdej z tych metod pojedyncze kaskady dotyczące tego samego artykułu są łączone przy pomocy dodatkowego, głównego węzła, tak jak zostało to przedstawione na rysunku 3.3. Wszystkie wartości atrybutów dla tego wierzchołka są ustawione na zero, ponieważ jest jedynie sposobem na zagregowanie kaskad dotyczących tej samej udostępnianej dalej informacji [17].

3.7.1. Rekonstrukcja na podstawie artykułów [13] [51]

Pierwsza z metod rekonstrukcji oryginalnych kaskad opiera się na sposobie zdefiniowanym w artykułach [13] oraz [51]. Aby wywnioskować co jest potencjalnym źródłem retweeta *i* identyfikowane są konta, które dany użytkownik obserwuje i które również udostępniły dany artykuł. Jeżeli znacznik czasowy badanego retweeta *i* wskazuje na to, że został on utworzony później niż retweet *j* jednej z osób, które analizowany użytkownik obserwuje, to zakłada się, że *j* jest źródłem dla *i*. Jeśli więcej niż jeden ze znajomych może być potencjalnym źródłem, typowany jest ten retweet, który nastąpił najpóźniej, czyli różnica czasu pomiędzy *i* oraz *j* jest najmniejsza. W przypadku, gdy nie znaleziono żadnego retweeta udostępnionego przez konta, które obserwuje rozważany użytkownik, zakłada się, że został podany dalej oryginalny tweet. Ta metoda w dalszych częściach pracy będzie nazywana *metodą dołączania do oryginalnego tweeta*.

3.7.2. Rekonstrukcja na podstawie artykułu [52]

Drugi algorytm został przedstawiony w artykule [52]. Podobnie jak w pierwszym podejściu, jeżeli przed dodaniem retweeta *i*, osoby, które śledzi dany użytkownik udostępniły ten sam artykuł, to zakłada się, że źródłem jest retweet dodany najpóźniej, ponieważ najnowsze posty są prezentowane w pierwszej kolejności na osi czasu Twittera, a zatem mają większe prawdopodobieństwo zretweetowania. Jeżeli użytkownik publikuje retweeta *i* i nie obserwuje żadnego konta występującego przed nim w sekwencji retweetów, w tym konta udostępniającego oryginalnego tweeta, szacuje się, że wiadomość rozprzestrzenia się od konta z największą liczbą obserwujących, ponieważ takie posty, zgodnie z zasadami dystrybucji treści na Twitterze, mają większą szansę na wyświetlenie na stronie głównej danego użytkownika. Ta metoda w dalszych częściach pracy będzie nazywana *metodą dołączania do posta najpopularniejszego użytkownika*.

3.8. Charakterystyka ilościowa zbioru danych

W tabeli 3.1 zostały zebrane informacje ilościowe dotyczące pobranych danych. Jak można zauważyć, nie dla wszystkich użytkowników dostępne są informacje związane z ich profilami lub z tym kogo obserwują na badanym portalu społecznościowym. Wynika to z ograniczeń pamięciowych podczas pobierania odpowiedzi z API Twittera. Dodatkowo można zauważyć, że mimo iż w zbiorze Gossipcop znajduje się o wiele więcej artykułów,

liczba aktywnych użytkowników uczestniczących w propagacji jest mniejsza niż w zbiorze Politifact.

Tabela 3.1. Charakterystyka zbiorów danych.

Lp	Parametr	Politifact	Gossipcop
1	Liczba aktywnych użytkowników biorących udział w kaskadach komunikacyjnych	497 240	489 942
2	Liczba prawdziwych newsów	403	15 652
3	Liczba nieprawdziwych newsów	385	5 056
4	Liczba tweetów	431 889	1 228 761
5	Liczba tweetów dotyczących prawdziwych informacji	315 602	766 981
6	Liczba tweetów dotyczących nieprawdziwych informacji	116 287	461 780
7	Liczba retweetów	526 951	469 498
8	Liczba retweetów dotyczących prawdziwych informacji	443 896	176 071
9	Liczba retweetów dotyczących nieprawdziwych informacji	83 055	293 427
10	Liczba pobranych profilów użytkowników uczestniczących w propagacji	256 959	257 594
11	Liczba pobranych informacji o followowaniu użytkowników uczestniczących w propagacji	299 498	263 391

Z tabeli 3.1 wynika, że rozkład próbek w klasach dla zbioru Politifact jest dosyć podobny. Z kolei tylko ok. 25% próbek w zbiorze Gossipcop ma klasę negatywną. Tak duży rozstrzał między licznościami klas może prowadzić do niskiej jakości predykcji, szczególnie dla klasy, która jest w mniejszości. W związku z tym zestaw danych został zbalansowany i aktualnie obie klasy zawierają taką samą liczbę próbek.

3.9. Przygotowanie danych

W ramach wstępnego przygotowania danych przed ich dalszym procesowaniem do formy stanowiącej wejście do modelu wykonano dwie operacje. Pierwsza z nich polega na odrzuceniu kaskad, które zawierają tylko jeden tweet i zero retweetów, ze względu na brak propagacji komunikatów. Druga polega na usuwaniu retweetów dla których czas pomiędzy udostępnieniem oryginalnego tweeta a stworzeniem retweeta przyjmuje wartości ujemne, co może sugerować błędy w API Twittera.

4. Analiza grafu użytkowników

Sieć społecznościowa Twittera powinna zostać odtworzona poprzez połączenie wszystkich użytkowników, którzy są ze sobą powiązani relacją obserwowania w tym serwisie. Do zbudowania tego grafu należałoby wykorzystać próbki zarówno ze zbioru Politifact jak i Gossipcop, ponieważ posiadają aż 63 454 wspólnych użytkowników. Taka rekonstrukcja zawierałaby informacje dotyczące kto kogo obserwuje tylko dla danego punku w czasie, w którym niezbędne informacje były pobierane z API. Rozważane medium to "żywa"społeczność, w której dynamicznie zmieniają się relacje obserwowania, użytkownicy zaczynają śledzić nowe konta, ale również mogą przestać obserwować wybrane. Ze względu na fakt, iż topologia grafu zmienia się wraz z upływem czasu, jest to graf dynamiczny [53]. Sieć, która powstałaby w wyniku tych operacji byłaby grafem skierowanym, w którym krawędź miałaby zwrot od użytkownika i do użytkownika j, jeżeli konto i obserwuje rozważane konto j. Krawędzie nie miałyby przypisanych wag, więc byłyby równoważne.

Sumaryczna liczba użytkowników, która posłużyłaby do stworzenia takiego grafu to 150 763 332. Wyjściowa sieć wciąż byłaby w pewien sposób niepełna, ze względu na tylko częściowe pobranie informacji o tym kto jak jest z kim połączony relacją obserwowania, co jest następstwem czasochłonności tego procesu oraz ograniczeniami pamięciowymi na urządzeniu. Niestety, mimo zasobów zawierających 50 GB pamięci RAM oraz wykorzystania biblioteki graph-tool, opisanej w podrozdziale 4.1, zbudowanie tak dużej sieci społecznościowej nie było możliwe, ponieważ ta operacja wymagała o wiele większych zasobów pamięciowych.

Graf należało drastycznie uprościć i okroić, tak, aby móc poddać go analizie oraz uzyskać wartości cechy obliczane zarówno w ujęciu całej sieci jak również pojedynczych wierzchołków. Wielkość grafu oraz w szczególności występujące w nim połączenia mogą w znacznym stopniu rzutować na uzyskane wyniki, jednak takie badania mogą pomóc wykazać, czy wzbogacanie cech wejściowych do modelu o takie atrybuty jest obiecującym kierunkiem dalszej analizy, lecz przy wykorzystaniu większych zasobów.

Sieć, która powstała i posłużyła do obliczenia wartości cech opisanych w podrozdziale 5.2.6 składa się z 519 213 wierzchołków (rząd grafu) oraz liczby krawędzi równej 34 166 738 (rozmiar sieci). Graf został zbudowany tylko i wyłącznie przy użyciu dostępnych informacji o tym kto kogo obserwuje pośród aktywnych użytkowników. Porównując rząd sieci z sumaryczną liczbą aktywnych użytkowników w obu grafach, zawartą w tabeli 3.1, widać, że liczba wierzchołków w grafie jest mniejsza. Jest to wynikiem niepełnego pobrania informacji o relacjach pomiędzy użytkownikami z API Twittera, ze względu na ograniczenia pamięciowe na urządzeniu.

4.1. Wykorzystane narzędzia

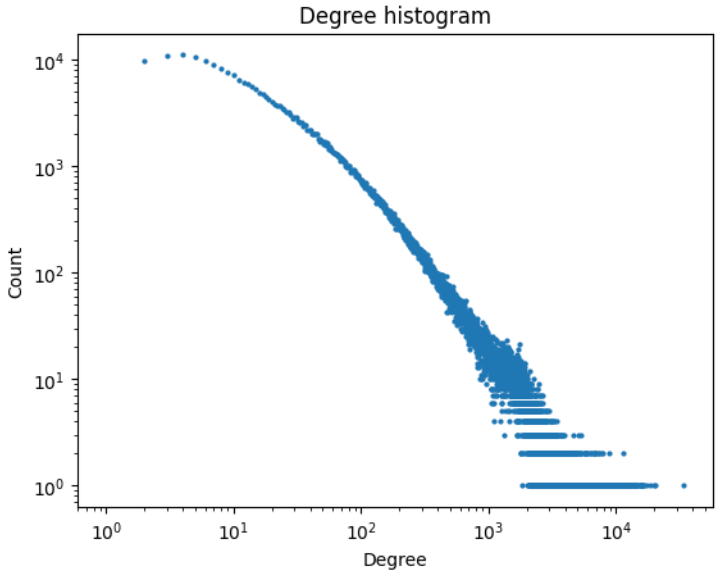
Do próby zbudowania grafu reprezentującego całą sieć społecznościową Twittera została wykorzystana biblioteka graph-tool , ponieważ większość jej algorytmów i struktur danych jest zaimplementowana w C++, co daje jej znaczną przewagę pod względem wydajności nad innymi grafowymi bibliotekami, takimi jak na przykład NetworkX. Niestety, nawet przy użyciu tak zaimplementowanej biblioteki nie udało się wczytać całego grafu do pamięci. Do przygotowania sieci, która zawiera tylko aktywnych użytkowników użyto biblioteki NetworkX, która została również wykorzystana do analizy grafu. Jest to biblioteka w całości zaimplementowana w Pythonie.

4.2. Największa składowa spójna grafu

Największy w pełni połączony podgraf zawiera 378 852 węzłów oraz 33 067 094 krawędzi. Analiza opisana w kolejnych podrozdziałach została przeprowadzona w oparciu o ten wycinek sieci. Ze względu na rozmiar grafu, niektórych cech takich jak stopień pośrednictwa, bliskości czy liczba oraz wielkości klik nie udało się policzyć niezależnie od użytej biblioteki. Obliczenia, które zakończyły się sukcesem zostały przedstawione poniżej.

4.3. Rozkład stopni wierzchołków

Na rysunku 4.1 został zaprezentowany rozkład stopni wierzchołków sieci społecznościowej w skali logarytmicznej na obu osiach.



Rysunek 4.1. Rozkład stopni wierzchołków

Wyraźnie widać, że większość wierzchołków w sieci jest połączona z niewielką ilością innych kont. Wierzchołki o stopniu powyżej 2000 występują wyraźnie rzadziej, w pojedyn-

czych przypadkach. Średni stopień wierzchołka dla analizowanej sieci wynosi ok. 45, a mediana 15.

4.4. Współczynnik mieszania asortatywnego

Współczynnik mieszania asortatywnego (ang. assortative mixing) definiuje skłonność do istnienia powiązań pomiędzy podobnymi wierzchołkami w sieci na podstawie ich atrybutów lub właściwości. Może być mierzony przy użyciu współczynnika asortatywności, który przyjmuje wartości z zakresu od -1 do 1. Dodatnie wartości współczynnika asortatywności pozwalają na sformułowanie wniosku, że wierzchołki z podobnymi atrybutami chętnie łączą się ze sobą, ujemne wartości sugerują łączenie węzłów o różnych atrybutach. Omówiony współczynnik pozwala na zauważenie w jaki sposób atrybuty i cechy wpływają na tworzenie połączeń pomiędzy wierzchołkami [54].

Współczynnik mieszania asortatywnego dla największego spójnego komponentu wynosi 0.079. Jest to wartość bliska zeru, co wskazuje na brak możliwości skorelowania połączonych węzłów bazując na ich cechach. Może to wynikać z faktu, że Twitter jest specyficznym medium społecznościowym, w którym niektórzy użytkownicy cieszą się bardzo dużą popularnością, nie odwzajemniając relacji obserwowania. Często też takie osoby są obserwowane przez użytkowników z dużo mniejszą liczbą obserwujących.

5. Przeanalizowane podejścia

W tej pracy zostały przeanalizowane dwa podejścia do zagadnienia wczesnego wykrywania fake newsów, jedno bazujące na klasycznych modelach uczenia maszynowego, czyli takich, których algorytmy nie opierają się na sieciach neuronowych oraz drugie wykorzystujące grafowe sieci neuronowe.

5.1. Założenia

Wczesne wykrywanie nieprawdziwych informacji daje możliwość jak najszybszego zaalarmowania o fałszywych wiadomościach w istniejącym procesie ich rozpowszechniania, a co za tym idzie powstrzymania go, zanim fake newsy dotrą do szerszego grona odbiorców [55]. Wraz ze wzrastającą ilością udostępnień fałszywych wiadomości, rośnie prawdopodobieństwo tego, że ludzie im zaufają [56], a trudno jest skorygować postrzeganie danego tematu, nawet jeśli poprzednie wrażenie jest mylne [57]. Dlatego też metody jak najwcześniejszego wykrywania fałszywych wiadomości są szeroko badane przez naukowców i korzystne w ujęciu społecznym.

Wczesna detekcja może zostać zdefiniowana w zależności od ilości czasu, który upłynął od pojawienia się pierwszego tweeta na dany temat, lub w zależności od liczby opublikowanych postów dotyczących konkretnej informacji. W literaturze stosowane są oba podejścia [58] [9] [17], dlatego w tej pracy badane są również dwa scenariusze: jeden z ograniczeniem czasowym, a drugi z liczbowym.

Wybrane zakresy do badań to odpowiednio:

- 1, 2, 4, 8, 12 i 24 godzin od pojawienia się pierwszego tweeta zgodnie z podejściem zaprezentowanym w artykule [59], oraz dodatkowo pierwszych 5 i 30 minut,
- pierwszych 10, 20, 30, 40, 50, 100 postów dotyczących danej wiadomości. Liczby zostały wybrane na podstawie artykułów [59] oraz [60]. W pierwszym z cytowanych badań brane pod uwagę są tylko retweety, w drugim zaś tweety. W niniejszej pracy zostały wzięte pod uwagę oba typy postów, ponieważ im częściej dany tweet jest retweetowany, tym większe prawdopodobieństwo, że pojawi się na tablicy większej liczby użytkowników, zgodnie z tym co zostało opisane w podrozdziale 3.7. Aby zbadać możliwości jak najwcześniejszej propagacji, do podanego zakresu dołączany jest jeszcze 5 pierwszych postów.

W celach referencyjnych modele zostały również wytrenowane na całych dostępnych kaskadach, tak aby móc przeprowadzić rzetelne porównanie wydajności przy ograniczonych kaskadach.

Do ewaluacji modeli w trakcie trenowania został wykorzystany zbiór walidacyjny, który został wyodrębniony ze zbioru treningowego i składa się z 20% próbek. Podział nastąpił w taki sposób, aby rozkład klas w obu zbiorach był zbliżony.

Zgodnie z tym, jak były prowadzone badania w przypadku artykułu dotyczącego kla-

sycznych modeli uczenia maszynowego [13] oraz analizującym grafowe sieci neuronowe [17] zestawy próbek ze strony Politifact i portalu Gossipcop nie były ze sobą łączone, modele dla każdego z tych serwisów były trenowane oddzielnie, co oznacza, że zbiory danych również są rozłączne.

5.2. Klasyczne modele uczenia maszynowego

Autorzy artykułu [13] przeprowadzają analizę porównawczą wpływu wyodrębnionych z kaskad cech strukturalnych, temporalnych oraz tekstowych na podstawie hierarchicznych sieci propagacji komunikatów na Twitterze. W badaniu zostały wykorzystane te same zbiory danych co w niniejszej pracy, czyli Politifact i Gossipcop, przy czym należy podkreślić, że nie są one tożsame pod względem zawartości, jako że publicznie dostępny jest jedynie kod służący do wykonywania zapytań do API Twittera, a nie konkretnie te same próbki.

Analizie poddane są kaskady z poziomu makro (struktura tweetów-retweetów) oraz mikro (struktura odpowiedzi na konkretne tweety). Badane jest to w jaki sposób fałszywe oraz prawdziwe wiadomości rozprzestrzeniają się na tych dwóch poziomach, wpływ na jakość predykcji wyodrębnionych atrybutów oraz jak dyskryminatywne są te cechy.

Artykuł jest próbą odpowiedzi na pytanie dotyczące korelacji między strukturą oraz zawartością hierarchicznej sieci propagacji a prawdziwością rozprzestrzenianych wiadomości.

Zaproponowano zestaw cech wyliczanych dla wszystkich kaskad dotyczących tego samego artykułu, połączonych głównym wierzchołkiem, tak jak zaprezentowano to na rysunku 3.3. Tak więc jedną próbką w zbiorze danych są wszystkie kaskady, w których udostępniane dalej są tweety związane z tym samym newsem, połączone dodatkowym węzłem.

Cechy podzielono na następujące kategorie:

- strukturalne,
- temporalne (ang. temporal),
- tekstowe.

Ze względu na to, że w niniejszej pracy wczesna detekcja nieprawdziwych informacji odbywa się bez udziału cech wyodrębnionych z tekstów tweetów oraz retweetów, tylko dwie pierwsze kategorie cech są brane pod uwagę. Kaskady komunikacyjne analizowane są przez pryzmat cech temporalnych dających możliwość ocenienia zaangażowania użytkowników w propagowanie informacji w sieci, pozwalając na wgląd w częstotliwość i intensywność tego procesu oraz poddawane analizie są atrybuty strukturalne.

Dodatkowo odwołując się do zdefiniowanego w podrozdziale 1.2 zagadnienia badawczego, w tej pracy analizę przeprowadzono jedynie dla sieci na poziomie makro, czyli takiej, która pozwala na poznanie wzorców publikowania oraz udostępniania dalej komunikatów,

tak aby zrozumieć globalny wzorzec rozprzestrzeniania się prawdziwych oraz fałszywych wiadomości.

5.2.1. Cechy strukturalne

Cechy strukturalne przedstawione w analizowanym artykule oraz użyte w testowanym podejściu:

1. *Głębokość drzewa (kaskady komunikacyjnej)* - maksymalna głębokość pojedynczej sieci makro. Pozwala na zobrazowanie jak szeroko informacje są rozpowszechniane przez użytkowników,
2. *Liczba węzłów* - liczba węzłów w sieci jest tożsama z liczbą użytkowników, którzy udostępniają dalej daną informację,
3. *Maksymalny stopień wychodzący (ang. outdegree)* - maksymalny stopień wychodzący w sieci makro. Wskazuje węzeł (tweet/retweet) o największym znaczeniu na proces propagacji,
4. *Liczba kaskad* - maksymalna liczba kaskad jest tożsama z liczbą unikalnych tweetów odnoszących się do tego samego artykułu,
5. *Głębokość węzła z maksymalnym stopniem wychodzącym* - głębokość w kaskadzie na której znajduje się węzeł z największym stopniem wychodzącym. Ta cecha obrazuje ilość kroków potrzebnych do dotarcia do najbardziej wpływowego węzła,
6. *liczba kaskad z retweetami* - liczba kaskad, w których znajduje się co najmniej jeden retweet,
7. *Odsetek kaskad z retweetami* - ułamek kaskad z retweetami w stosunku do wszystkich kaskad dotyczących tego samego artykułu.

5.2.2. Cechy temporalne (ang. temporal)

Cechy czasowe zaproponowane w przytaczanym badaniu, które zostały użyte w testowanym podejściu:

1. *Średnia czas jaki upływa pomiędzy powstaniem kolejnych retweetów* - cecha pozwalająca na ocenę jak szybko dana informacja rozprzestrzenia się poprzez następujące po sobie retweety,
2. *Różnica czasu pomiędzy pierwszym tweetem a ostatnim retweetem* - wskazuje na długość trwania procesu propagacji wiadomości w danej kaskadzie,
3. *Różnica czasu pomiędzy pierwszym tweetem a tweetem z maksymalnym stopniem wychodzącym* - pozwala na ustalenie ile czasu upływa zanim dany artykuł zostanie udostępniona przez najbardziej wpływowy węzeł,
4. *Różnica czasu pomiędzy pierwszym i ostatnim tweetem* - wskazuje jak długo posty związane z danym artykułem są publikowane na Twitterze,
5. *Różnica czasu w obrębie najgłębszej kaskady informacyjnej pomiędzy tweetem publikującym dany artykuł a ostatnim retweetem* - ta różnica czasu wskazuje maksymalny czas życia danej informacji biorąc pod uwagę wszystkie kaskady odnoszące się do

tego samego artykułu. Pozwala na określenie czy retweety następują po sobie w sposób gwałtowny, czy powolny,

6. *Średnia czas jaki upływa pomiędzy powstaniem kolejnych retweetów w najgłębszej kaskadzie* - cecha wskazująca z jaką częstotliwością dana wiadomość jest retweetowana w najgłębszej kaskadzie komunikacyjnej.
7. *Średni czas pomiędzy tweetami publikującymi daną informację* - określa w jakich odstępach czasu powstają tweety dotyczące tego samego artykułu,
8. *Średni czas jaki upływa pomiędzy tweetem a pierwszym retweetem* - pozwala na ustalenie, jak szybko po pojawieniu się tweeta jest on retweetowany.

5.2.3. Modele wykorzystane w bazowym artykule

Do przeprowadzenia predykcji autorzy wybrali następujące algorytmy uczenia maszynowego:

- gaussowski naiwny klasyfikator Bayesa (ang. Gaussian Naive Bayes),
- drzewo decyzyjne (ang. decision tree),
- regresja logistyczna (ang. logistic regression),
- las losowy (ang. random forest).

W algorytmach nie zastosowano optymalizacji hiperparametrów, były więc uczone na domyślnych wartościach tych parametrów.

Dodatkowo, została przeprowadzona analiza wpływu konkretnych cech na wyniki osiągane przez model. Wykorzystano do tego algorytm ExtraTreeClassifier, który jest modelem zespołowym, podobnym do lasu losowego. Ważności atrybutów w kontekście dokonywanych predykcji była obliczana na podstawie współczynnika Giniego (ang. gini impurity).

5.2.4. Nowe cechy zaproponowane na podstawie przeglądu literatury

W ramach przeglądu literatury do istniejących cech zostały dodane te przedstawione poniżej:

1. *współczynnik śledzenia* - cecha pozwalająca na ocenę reputacji użytkownika, która jest obliczana jako logarytm ze stosunek liczby osób, które dany użytkownik obserwuje do liczby osób, które są przez tego użytkownika obserwowane [61]:

$$\text{Follow ratio(u)} = \log_{10}\left(\frac{|N^t(u)|}{|N^f(u)|}\right) \tag{13}$$

   gdzie:
   $|\cdot|$ oznacza moc zbioru,
   $N^t(u)$ - zbiór użytkowników obserwujących konto u,
   $N^f(u)$ - zbiór użytkowników których obserwuje użytkownik u
2. *Reputacja konta* - ta cecha pozwala na zbadanie różnicy w reputacji kont botów oraz prawdziwych użytkowników, poprzez kombinację liczby obserwujących i znajomych

użytkownika, w sposób zdefiniowany równaniem [62]:

$$\text{Reputation(u)} = \frac{|N^t(u)|}{|N^t(u)| + |N^f(u)|} \tag{14}$$

gdzie oznaczenie jest takie samo jak w równaniu 13.

3. *Popularność tweeta* - w artykule [63] popularność tweeta została zdefiniowana jako całkowita suma retweetów i odpowiedzi na oryginalny tweet/retweety podzielona przez liczbę osób obserwujących użytkownika. Wynikiem jest procent zainteresowania osób obserwujących danego użytkownika udostępnionym przez niego tweetem. W tej pracy nie uwzględniane są odpowiedzi na tweety, więc równanie zostało zmodyfikowane do postaci:

$$\text{Tweet popularity(t)} = \frac{|R(t)|}{|N^t(u)|} * 100 \tag{15}$$

gdzie:
$R(t)$ - zbiór wszystkich retweetów tweeta t dodanego przez konto u

4. *Stosunek wielkości warstw* - numerem warstwy określa się ilość przeskoków dzielących tweeta od danego retweeta. Stosunek rozmiarów warstaw jest zdefiniowany równaniem: [64]

$$\text{Layer ratio} = \frac{|N_2|}{|N_1|} \tag{16}$$

gdzie
N_i jest zbiorem węzłów występujących w danej warstwie i kaskady komunikacjnej

5. *Procent retweetów* - to cecha wyliczana dla całej sieci makro, zdefiniowana jako stosunek wszystkich retwetów w sieci do sumy wszystkich retweetów i tweetów. Może zostać zapisana następującym równaniem [65]:

$$\text{Retweet percentage} = \frac{\sum_1^T |R(t)|}{\sum_1^T |R(t)| + T} \tag{17}$$

gdzie:
T - liczba tweetów w danej kaskadzie,
$R(t)$ - zbiór wszystkich retweetów tweeta t

6. *współczynnik strukturalnego rozprzestrzenianie się wiadomości* - cecha zaproponowana w artykule [66] pod angielską nazwą structural virality, która może być stosowana do analizy sieci społecznościowych. Badane są strukturalne cechy wpływające na to jak udana jest propagacja wiadomości. Współczynnik bierze pod uwagę dwa skrajne style rozprzestrzeniania się wiadomości: transmisję od pojedynczego wierzchołka do dużej liczby węzłów, oraz transmisję, która rozwija się od użytkownika do użytkownika, z wieloma wierzchołkami pośrednimi. Współczynnik jest zdefiniowany

jako średnia odległość pomiędzy parami węzłów w kaskadzie.

$$v(T) = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} \tag{18}$$

gdzie:
$v(T)$ - współczynnik strukturalnego rozprzestrzeniania się wiadomości dla kaskady komunikacyjnej T,
n - liczba wierzchołków w kaskadzie, przy czym $n > 1$,
d_{ij} - długość najkrótszej ścieżki pomiędzy wierzchołkami i oraz j.
Wyższe wartości tej miary sugerują występowanie transmisji od użytkownika do użytkownika.

5.2.5. Nowe cechy wyodrębnione z profili użytkowników

Są to cechy wyodrębnione na podstawie artykułu [17]:

1. informacja o tym, czy konto danego użytkownika jest zweryfikowane czy nie,
2. data utworzenia konta,
3. liczba obserwujących,
4. liczba "przyjaciół",
5. liczba list,
6. liczba ulubionych,
7. liczba statusów,
8. znacznik czasowt tweeta/retweeta.

Dokładne opisy powyższych atrybutów znajdują się w podrozdziale 5.3.1.

5.2.6. Nowe cechy zaproponowane na podstawie analizy sieci społecznościowej

Oprócz atrybutów wyodrębnionych z przeglądu literatury oraz profili kont, dodano również 5 cech bazujących na strukturze sieci społecznościowej łączącej wszystkich użytkowników, opisanej w rozdziale 4:

1. *Page rank* - algorytm pozwalający na wyznaczenie rankingu węzłów w grafie na podstawie krawędzi przychodzących [67]. Stworzony oryginalnie do wyznaczania pozycji stron internetowych i opisany w artykule [68],
2. *Lokalny stopień gronowania węzłów* (ang. clustering coefficient) - miara określająca tendencję węzłów w grafie do łączenia się w klastry. W przypadku grafów skierowanych ten współczynnik jest definiowany jako ułamek wszystkich możliwych skierowanych trójkątów przechodzących przez rozważany węzeł, które istnieją w grafie [69] [70]. Pozwala na wyznaczenie gęstości sieci wokół wierzchołka, mówiącej o tym w stopniu poszczególni obserwatorzy danego użytkownika też się wzajemnie śledzą.

$$c_u = \frac{T(u)}{2(deg^{tot}(u)(deg^{tot}(u)-1) - 2deg^{\leftrightarrow}(u))} \tag{19}$$

gdzie:
$T(u)$ - liczba skierowanych trójkątów przechodzących przez węzeł u
$deg^{tot}(u)$ - suma stopnia wyjściowego oraz wejściowego węzła u
$deg^{\leftrightarrow}(u)$ - jest odwrotnością stopnia wierzchołka (ang. reciprocal degree) u

3. *Współczynnik wzajemności* (ang. total reciprocity) - cecha opisana w artykule [71]. Jest to stosunek liczby węzłów, z którymi rozważany wierzchołek ma odwzajemnione obserwowanie (czyli ma krawędź skierowaną w obu kierunkach) do całkowitej liczby węzłów, które obserwuje (krawędzi wychodzące):

$$TR_p = \frac{|I_p \cap O_p|}{|O_p|} \tag{20}$$

gdzie:
O_p - liczba wierzchołków do których węzeł p ma połączenie wychodzące, czyli liczba kont, która jest obserwowana przez użytkownika p,
I_p - liczba kont obserwujących użytkownika p, czyli liczba wierzchołków które mają połączenie wychodzące do węzła p.

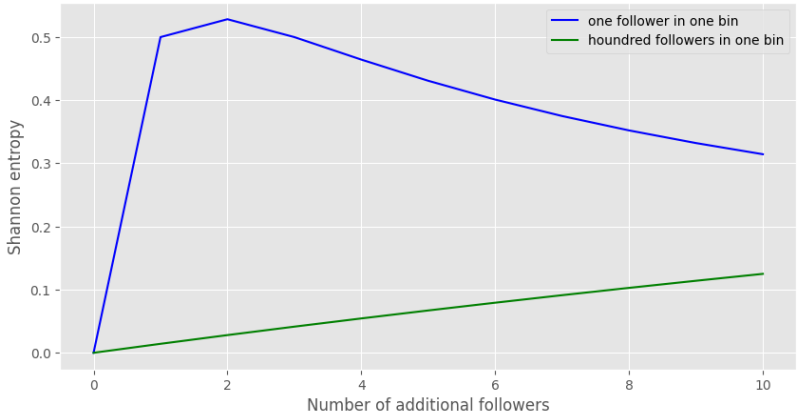
4. *Współczynnik agregujący aktywność użytkownika w okresie poprzedzającym dodanie przez niego konkretnego posta (tweeta lub retweeta)* - jest to nowa cecha zaproponowana w tej pracy. Ten indeks bada niedawną historię aktywności danego konta na Twitterze, określającą w ilu kaskadach uczestniczył przed dodaniem rozważanego posta. Jako okres badawczy zostały przyjęte dwa tygodnie przed znacznikiem czasowym nowego tweeta/retweeta. Taki indeks pozwala ocenić zaangażowanie użytkownika w udostępnianie informacji na portalu. Założeniem dla takiego podejścia jest przypuszczenie, że nieprawdziwi użytkownicy/boty mogą być szczególnie aktywne w krótkim przedziale czasu, rozprzestrzeniając nieprawdziwe wiadomości,
5. *Entropia informacyjna węzła* - jest również nową cechą zaproponowaną w tej pracy, która bazuje na entropii informacyjnej Shannona. Przyjęto założenie, że często fałszywe informacje rozprzestrzeniane są przez nieprawdziwych użytkowników, tzw. boty, które są tworzone w dużych ilościach do wykonania konkretnego zadania, na przykład na Twitterze do propagacji zadanego fake newsa.
   Badana hipoteza jest sformułowana następująco: konta botów są tworzone "hurtowo", kolejni użytkownicy są dodawani w krótkich odstępach czasu i łączą się ze sobą relacją obserwowania. Relacja śledzenia dla prawdziwych użytkowników ma o wiele bardziej chaotyczny charakter i nie da się stworzyć prostego wzorca opisującego ten proces.
   W celu nadania pewnej struktury punktom na osi czasu, stworzono przedziały odpowiadające różnicy czasów powstania dwóch kont - rozważanego użytkownika i jego znajomego. Wybrano następujące przedziału: od 0 sekundy (moment powstania analizowanego użytkownika) do 1 godziny, od 1 godziny do 24 godzin, od 24 godzin do

jednego tygodnia, od tygodnia do miesiąca, od miesiąca do nieskończoności. Entropia jest miarą losowości w zbiorze danych, gdzie im mniej punktowa jest dystrybucja próbek, tym wartość entropii maleje. W tym konkretnym zagadnieniu spodziewamy się małych wartości entropii dla botów oraz wysokich dla prawdziwych użytkowników. Entropia informacyjna Shannona jest liczona na podstawie prawdopodobieństwa [72].

W tym podejściu problematycznym przypadkiem jest ten, w którym prawdziwy użytkownik ma tylko jednego znajomego. W takim wypadku entropia informacyjna węzła może być równa entropii bota, którego wszyscy obserwatorzy powstali w dokładnie jednym przedziale czasowym. Aby poradzić sobie z takimi przypadkami, przyjęto, że dla każdego badanego użytkownika całkowita liczba znajomych zostanie zwiększona o n dodatkowych użytkowników. Jako n wybrano 2. Prawdopodobieństwo wystąpienia zdarzenia w danym przedziale jest liczone następująco:

$$p(k) = x/(m+n) \tag{21}$$

gdzie:
$p(k)$ - prawdopodobieństwo tego, że nowa próbka znajdzie się w przedziale k,
x - liczba próbek znajdujących się w przedziale k,
m - liczba wszystkich próbek,
n - liczba dodatkowych sąsiadów branych pod uwagę.

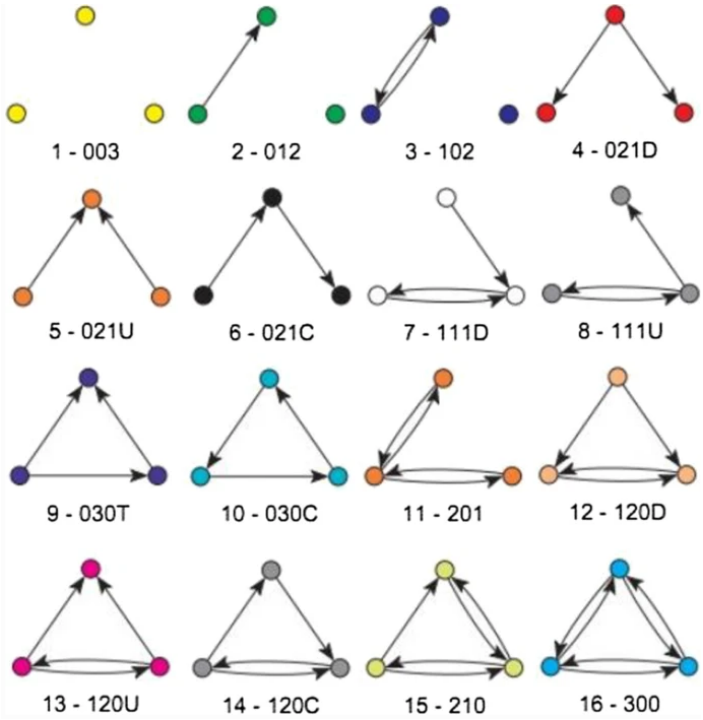


Rysunek 5.1. Zmiana wartości entropii informacyjnej Shannona w uzależnieniu od ilości dodatkowych osób followujących.

Można zauważyć, że przy dużej liczbie śledzących danego użytkownika kont, wpadających w jeden przedział czasowy, dodanie dwóch nowych sąsiadów powoduje minimalną zmianę, gdzie przy jednym użytkowniku w przedziale wartość entropii znacząco wzrasta.

6. *indeks triadowy przechodniości* - jest to nowy indeks zaproponowany w tej pracy, definiowany dla każdego węzła grafu skierowanego, a nie dla całej sieci jak jest to

spotykane w literaturze. Dla podgrafu ego konkretnego wierzchołka wyznaczane są triady przechodnie i nieprzechodnie bazując na bezpośrednim sąsiedztwie węzła (kontach, które danego użytkownika śledzą, oraz tych, które są przez niego obserwowane). Następnie liczony jest stosunek liczby triad przechodnich do sumy triad przechodnich i nieprzechodnich. Na rysunku 5.2 zostały zaprezentowane możliwe triady w sieci:



Rysunek 5.2. Triady, które mogą wystąpić w sieci społecznościowej [73]

Jest to podejście, które kładzie nacisk na pojedyncze węzły, oraz to jak istotne są w sieci.

5.2.7. Zbadane podejścia do uczenia modeli

Oprócz wykorzystania do trenowania takich samych klasyfikatorów, jakie zostały użyte w oryginalnym artykule, sprawdzono również czy można uzyskać wyższą jakość klasyfikacji następującymi sposobami:

- Trenując nowe klasyfikatory:
  - — Adaboost opisany w podrozdziale 2.2.7,
  - — maszynę wektorów nośnych (SVM) przedstawiony w podrozdziale 2.2.4,
  - — k najbliższych sąsiadów (kNN) opisany w podrozdziale 2.2.5,
- Trenując zespoły klasyfikatorów:
  - — głosowania większościowego opisanego w podrozdziale 2.3.1. Do zbudowania zespołu zostały wykorzystane wszystkie dostępne klasyfikatory, czyli: drzewo de-

cyzyjne, las losowy, gaussowski naiwny klasyfikator Bayesa, regresja logistyczna, maszyna wektorów nośnych, Adaboost oraz k najbliższych sąsiadów,

— generalizacji stosowej przedstawiony w podrozdziale 2.3.2. Tutaj również do zbudowania zespołu klasyfikatorów na pierwszym poziomie zostały użyte wszystkie dostępne modele. Klasyfikatorem, który dokonuje ostatecznej predykcji jest algorytm regresji logistycznej,

— dynamicznego dobierania zespołu opisanego w podrozdziale 2.3.3. Do stworzenia zespołu zostały użyte wszystkie dostępne algorytmy, takie same jak dla głosowania większościowego. Wybrany algorytm to KNORA-Eliminate, przedstawiony w podrozdziale 2.3.3.

Dodatkowo, dane zostały znormalizowane poprzez odjęcie wartości średniej oraz podzielenie przez odchylenie standardowe. Wyniki uzyskane wymienionymi wyżej sposobami oraz opisanymi w artykule klasyfikatorami zostały szczegółowo omówione w podrozdziale 6.1 zawierającym ocenę wyników.

Aby zbadać udział danych kategorii cech w jakości predykcji, klasyfikatory były trenowane na różnych zestawach cech zawierających odpowiednio: cechy strukturalne, nowe cechy (zaproponowane na podstawie przeglądu literatury opisane w podrozdziale 5.2.4, profili użytkowników 5.2.5 oraz analizy sieci społecznościowej 5.2.6) oraz temporalne.

5.3. Grafowe modele uczenia maszynowego

W badaniu [17] zostało przedstawione odmienne podejście do klasyfikacji wiadomości na prawdziwe i fałszywe. Zrezygnowano z ręcznego tworzenia cech, których wartości stanowiłyby wejścia do modeli, wykorzystano natomiast grafy i zbudowano architekturę opartą na grafowych sieciach neuronowych. Analizowany artykuł skupia się na próbie stworzenia modelu, który nie bazowałby na takich informacjach jak treść tweetów i odpowiedzi. W tych badaniach położono nacisk na detekcję nieprawdziwych informacji biorąc pod uwagę jedynie to jak dane artykuły rozprzestrzeniają się na Twitterze, ponieważ już wcześniej wykazano, że prawdziwe i fałszywe informacje propagują w mediach społecznościowych w odmienny sposób [11].

W omawianym badaniu zostały również wykorzystane zbiory danych możliwe do pobrania przy pomocy FakeNewsNet: Politifact oraz Gossipcop. Próba rekonstrukcji kaskad komunikacyjnych obywała się w częściowo odmienny sposób niż w metodach, które zostały opisane w podrozdziale 3.7. Tak jak poprzednio retweety znajdujące się w danej kaskadzie są sortowane po znacznikach czasowych ich powstania od najnowszych do najstarszych. W kolejnym kroku następuje proces wyszukiwania potencjalnego źródła danego retweeta spośród tweetów i retweetów opublikowanych przed nim. Krawędź jest dodawana pomiędzy dwoma użytkownikami jeżeli użytkownik wspominał innego użytkownika w swoim poście lub jeśli dany post powstał w określonym czasie po innym tweecie/retweecie. To podejście nie zostało przetestowane w niniejszej pracy z dwóch

powodów. Po pierwsze, pod uwagę brana jest pośrednio treść tweeta, czyli następuje wyciąganie informacji z treści postu dotyczących tego czy rozważany użytkownik dodał oznaczenie innego konta. Po drugie, nie został sprecyzowany przedział czasu dla drugiego warunku, czyli dodawania krawędzi jeśli dany retweet powstał w pewnym odstępie czasu po innym retweecie/tweecie.

5.3.1. Cechy wyodrębnione z profili użytkowników

Jako atrybuty dla węzłów zostały wybrane następujące cechy bazujące na informacjach, które można znaleźć w pobranych danych o profilach użytkowników z API Twittera:

1. *informacja o tym, czy konto danego użytkownika jest zweryfikowane czy nie*,
2. *data utworzenia konta* zakodowana jako czas, który upłynął od momentu powstania Twittera do momentu utworzenia konta, wyrażony w miesiącach,
3. *liczba obserwujących*, których posiada obecnie dany użytkownik,
4. *liczba "przyjaciół"*, czyli użytkowników, którzy śledzą dane konto,
5. *liczba list*, czyli liczba publicznie dostępnych list, do których należy ten użytkownik,
6. *liczba ulubionych*, czyli liczba tweetów, które zostały polubione przez danego użytkownika licząc od momentu powstania konta,
7. *liczba statusów*, jest to liczba tweetów i retweetów utworzonych przez danego użytkownika licząc od momentu powstania konta,
8. *znacznik czasowt tweeta/retweeta* obliczany jako czas, który upłynął od momentu powstania pierwszego tweeta odnoszącego się do danego "newsa", wyrażony w sekundach.

Dokładne opisy cech powstały na podstawie dokumentacji API Twittera [74].

5.3.2. Cechy wyodrębnione na podstawie aktywności użytkowników

Oprócz zaprezentowanych powyżej cech, został stworzony kolejny graf użytkowników, bazujący na "osi czasu"tweetów danego konta. API Twittera udostępnia możliwość pobrania ostatnich 200 tweetów/retweetów danego użytkownika, czyli jego niedawną aktywność. Cechy wyodrębnione na podstawie tego grafu według autorów artykułu mogą pozwolić zauważyć fenomen, w którym mniej wiarygodni użytkownicy mają większą tendencję do współpracy między sobą. W tym grafie węzłami są użytkownicy, a krawędź jest tworzona między kontem i oraz j jeśli dany użytkownik j został wspominany przez innego użytkownika i, przy czym są to krawędzi ważone, gdzie wagą jest liczba razy, kiedy dany użytkownik i oznaczył użytkownika j. Na podstawie tego grafu wyodrębniane są następujące cechy:

1. *stopień wejściowy wierzchołka*, czyli sumaryczna liczba kont, które w swoich tweetach/retweetach oznaczyły użytkownika i,
2. *stopień wyjściowy wierzchołka*, czyli sumaryczna liczba kont, którzy zostali oznaczeni przez użytkownika i w jego postach,

3. *ważony stopień wejściowy wierzchołka*, czyli całkowita liczba sytuacji, w których nastąpiło oznaczenie użytkownika *i* w tweetach/retweetach,
4. *ważony stopień wyjściowy wierzchołka*, czyli całkowita liczba sytuacji, w których użytkownik *i* oznaczył w swoich postach innych użytkowników,
5. *liczba sąsiadów oddalonych o dwie krawędzie skierowane do rozważanego użytkownika i* (ang. hop-2-in-neighbours),
6. *liczba sąsiadów oddalonych o dwie krawędzie skierowane zewnętrzne od rozważanego użytkownika i* (ang. hop-2-out-neighbours)
7. *całkowita liczba tweetów znajdujących się w pobranej osi czasu użytkownika.*

Niestety, pobieranie danych dotyczących ostatniej aktywności danego użytkownika wymagało zbyt dużych zasobów pamięciowych oraz analizy tekstowej tweetów w celu wyszukania oznaczeń, więc opisane wyżej grafy nie zostały włączone w zakres niniejszej pracy.

5.3.3. Architektura modelu

Zaprezentowany przez autorów model opiera się na grafowych sieciach neuronowych w szczególności na warstwach GraphSage przeplecionych warstwami łączącymi (ang. pooling layers), konkretnie DiffPool. Modele były ewaluowane przy użyciu powszechnie stosowanych metryk w klasyfikacji, takich jak dokładność (ang. accuracy), precyzja (ang. precision), czułość (ang. recall) oraz miara F1 (ang. F1 score), które zostały opisane w podrozdziale 2.4. W wyniku testów na różnych wartościach hiperparametrach, autorzy określili, że dla zbioru danych Politifact najlepsze wyniki były osiągane przy czterowarstwowej sieci, z 64 neuronami w warstwie ukrytej. Dla zbioru Gossipcop liczba warstw wynosiła 2, przy takich samych pozostałych parametrach.

Następnie modele zostały porównane z istniejącymi podejściami w literaturze pokazując, że bazując jedynie na informacjach dotyczących propagacji komunikatów można osiągnąć porównywalnie dobre lub lepsze wyniki jak w przypadku cech bazujących na m.in. treści wiadomości.[17]

W artykule została również przeprowadzona analiza pod kątem wczesnej detekcji nieprawdziwych informacji. Przetestowano dwa podejścia. W pierwszym grafy zostały ograniczone przez warunek maksymalnej ilości tweetów dla każdego newsa: brane było pod uwagę pierwszych 100, 200, 500, 1000 tweetów. W drugim było to ograniczenie mieszane, grafy były przycinane do pierwszych 100 tweetów, lub brane były tweety z pierwszej, trzeciej, piątej lub siódmej godziny - w zależności który warunek (ilościowy lub czasowy) był bardziej restrykcyjny. Wyniki ponownie pokazały, że zaproponowany przez autorów model radzi sobie lepiej niż dotychczasowe modele, uważane za najlepsze w tej dziedzinie.[17]

5.3.4. Przygotowanie danych

Twórcy artykułu [17] nie udostępnili publicznie kodu, przez co nie ma możliwości zweryfikowania uzyskanych przez nich wyników. Poniżej zostały opisane kroki, które zostały wykonane w ramach niniejszej pracy.

Grafowe sieci neuronowe są szczególnym typem sieci neuronowych, w których wejściem mogą być grafy, również ze zmienną liczbą wierzchołków oraz krawędzi. Wymagane jest jednak, aby elementem o niezmiennym rozmiarze był wektor atrybutów wierzchołków. Taki zbiór danych został przygotowany przy pomocy biblioteki torch-geometric. Do cech bazujących na profilach użytkowników, opisanych w podrozdziale 5.3.1, zaproponowanych przez autorów artykułu [17] zostały dodane cechy, które można było wyznaczyć dla każdego użytkownika znajdującego się w danej kaskadzie, opisane w podrozdziale 5.2.4 oraz 5.2.6:

- *współczynnik śledzenia*,
- *reputacja konta*,
- *page rank*,
- *lokalny stopień gronowania węzła*,
- *entropia informacyjna węzła*,
- *współczynnik wzajemności dla węzła*,
- *współczynnik agregujący aktywność użytkownika w okresie poprzedzającym dodanie przez niego konkretnego posta (tweeta lub retweeta)*,

Ze względu na to, że nie dla wszystkich użytkowników były dostępne informacje o ich profilach oraz relacji obserwowania, zostało przeprowadzone wzbogacanie cech w macierzy sąsiedztwa opisane w artykule [75]. Zaprezentowany w nim algorytm opiera się na minimalizacji energii Dirichleta.

5.3.5. Ustawienia treningu

W celu znalezienia konfiguracji hiperparametrów zapewniającej najlepszą jakość predykcji została przeprowadzona ich optymalizacja przy użyciu biblioteki *optuna* [76]. Miarą, której wartość była minimalizowana to wartość funkcji straty na zbiorze walidacyjnym. Hiperpatametrami, które były optymalizowane są:

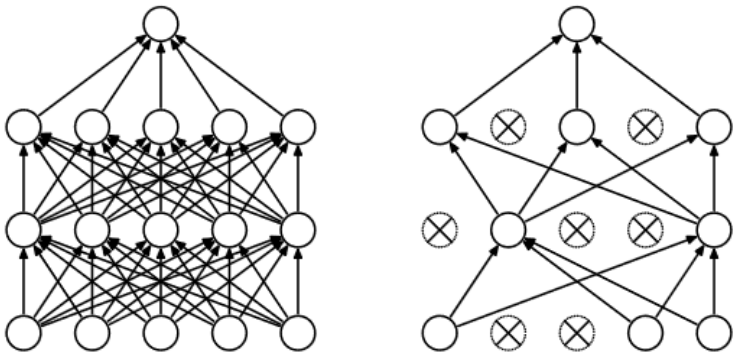
- *rozmiar wsadu (ang. batch size)* - trenowanie wsadowe polega na tym, że zbiór treningowy jest dzielony na mniejsze, równe części (ostatnia część może zawierać mniej elementów, jeśli liczba próbek w zbiorze treningowym nie jest podzielna bez reszty przez rozmiar wsadu) na których iteracyjnie trenowany jest model. Po każdym takim wsadzie wewnętrzne parametry modelu są aktualizowane. Do zalet tej metody należy na pewno mniejsze użycie pamięci oraz szybszy trening, ponieważ parametry sieci są częściej uaktualniane.
- *współczynnik uczenia (ang. learning rate)* - jest to współczynnik, który określa jak w oparciu o szacowany błąd należy zmienić model gdy jego wagi są aktualizowane. Zbyt

mały współczynnik uczenia może skutkować powolnym zbieganiem do minimum, a co za tym idzie długim czasem treningu, a zbyt duża wartość może powodować fluktuacje wartości funkcji straty prowadząc do niestabilnego procesu uczenia, oraz utknięcia w minimum lokalnym, skutkujące niską jakością predykcji [77]. Wartości współczynniku uczenia znajdują się najczęściej w zakresie od 0,0 do 1,0:

$$\theta_t = \theta_{t-1} - \eta_t \nabla L_\theta \tag{22}$$

gdzie:
L_θ - funkcja straty,
∇L_θ - gradient funkcji straty w bieżącej iteracji t,
θ - wagi ,
η_t - współczynnik uczenia, który kontroluje w jak dużym stopniu zostaną uaktualnione wagi θ w bieżącej iteracji t.

- *dropout* - to metoda regularyzacji, która pozwala na radzenie sobie ze zbyt szybkim przeuczeniem dużej sieci neuronowej, w szczególności przy niewielkim zbiorze treningowym. W trakcie procesu uczenia, losowo wybrane neurony są tymczasowo "dezaktywowane"razem ze swoimi połączeniami wejściowymi i wyjściowymi. Losowe odrzucanie neuronów aproksymuje równoległe trenowanie wielu sieci neuronowych o różnych architekturach na tym samym zbiorze danych [78]. Dropout jest używany tylko podczas trenowania sieci. Wartości tego współczynnika wahają się w zakresie od 0,0 do 1,0 i określają jaki procent neuronów z danej warstwy będzie odrzucany podczas danej epoki.



Rysunek 5.3. Reprezentacja metody działania dropoutu [78]

- *liczba warstw ukrytych/głębokich* - liczba określająca ile warstw ukrytych należy dodać do modelu. Po każdej warstwie ukrytej dodawana jest warstwa z funkcją aktywacji, oraz kolejna normalizująca próbki. Normalizacja wsadu (ang. batch normalization) pomaga zminimalizować wewnętrzne przesunięcie kowariancji (ang.

internal covariate shift), które występuje kiedy rozkład danych wejściowych w danym wsadzie do warstw głębokich jest zmienny, szczególnie w głębokich sieciach neuronowych, gdzie parametry wszystkich poprzednich warstw wpływają na wejścia do nowej warstwy. Takie fluktuacje są problematyczne, gdyż warstwy muszą się stale dopasowywać do nowych rozkładów występujących w danych. Normalizacja wsadu pomaga ustandaryzować wartości przykładów wejściowych do warstw głębokich dla każdej partii próbek [79].

- *liczba neuronów w każdej warstwie ukrytej* - liczba określająca jeden z wymiarów reprezentacji cech, wejściowy do warstwy lub wyjściowy. W pierwszej warstwie grafowej sieci neuronowej liczba neuronów ukrytych jest równa liczbie cech w danych wejściowych. Większa liczba cech pozwala na wychwycenie bardziej złożonych zależności występujących w danych, pozwalając modelowi na naukę bardziej skomplikowanych relacji w grafie [80].

Aby uniknąć zjawiska przeuczenia sieci neuronowej, wykorzystane zostało wczesne zatrzymywanie uczenia (ang. early stopping). Jego działanie można opisać w następujący sposób: jeśli w ciągu n następujących po sobie epok, funkcja straty nie osiągnie niższej wartości, to uczenie jest przerywane. W tym przypadku liczba epok jest równa $n = 10$. Funkcja straty/celu określa w jakim stopniu wyniki predykcji zwrócone przez model odbiegają od rzeczywistych wartości znajdujących się w danych.

W niniejszej pracy rozważanym problemem jest klasyfikacja binarna, więc na funkcję straty została wybrana binarna entropia skrośna (ang. binary cross-entropy) zaimplementowana w bibliotece Pytorch *BCEwithLogitsLoss* [81]. Ostatnią warstwą modelu jest warstwa liniowa z jednym neuronem wyjściowym. To, co jest wyjściem z klasyfikatora jest następnie podawane do funkcji straty. Wzór na funkcję straty:

$$
\begin{gathered}
l_n = -w_n[y_n \cdot log\sigma(x_n) + (1-y_n) \cdot log(1-\sigma(x_n))] \\
L = l_1, ..., l_N{}^T \\
l(x,y) = \begin{cases} \text{mean}(L) & \text{if reduction = 'mean'} \\ \text{sum}(L) & \text{if reduction = 'sum'} \end{cases}
\end{gathered}
\tag{23}
$$

gdzie:

x - wyjście z sieci, które stanowi wejście do funkcji celu,

y - rzeczywiste klasy z danych,

N - liczba próbek w pojedynczym wsadzie danych (ang. batch),

w - tensor wag o rozmiarze równym liczbie próbek znajdujących się w rozważanym wsadzie. Jest to parametr opcjonalny, służący do skalowania wartości funkcji straty,

σ - funkcja sigmoidalna, która jest zdefiniowana jako:

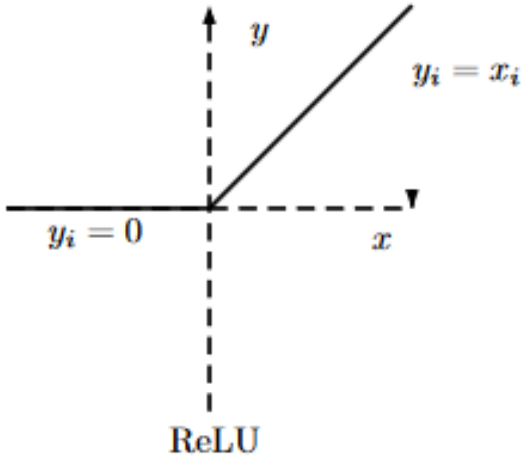
$$
\text{Sigmoid}(x) = \sigma(x) = \frac{1}{1+exp(-x)} \tag{24}
$$

Funkcje aktywacji w sieciach neuronowych pozwalają na wprowadzenie nieliniowości do procesu uczenia. Dzięki temu nawet skomplikowane, wielomianowe funkcje opisujące dane wejściowe mogą zostać wychwycone przez sieć, co nie byłoby możliwe w przypadku liniowym. Tak więc są niezbędne w procesie uczenia modelu bardziej złożonych funkcji [82].

W niniejszej pracy została wykorzystana poprawiona funkcja liniowa (ang. rectified linear unit, w skrócie ReLU) jest zdefiniowana matematycznie w następujący sposób:

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{25}$$

Jak widać, jeśli wejście do funkcji aktywacji ma wartość ujemną lub równą zeru, to zwracana jest aktywacja równa zero, w przeciwnym przypadku jest zwracana wartość wejściowa. Jest to funkcja z dwiema częściami liniowymi [83].



Rysunek 5.4. Funkcja aktywacji ReLU [84]

5.3.6. Zbadane podejścia do uczenia modeli

Zgodnie z założeniami opisanymi w 5.1 modele były trenowane z ograniczeniami zarówno czasowymi i ilościowymi w postaci redukcji liczby tweetów i retweetów branych pod uwagę w danej kaskadzie oraz na pełnych danych, aby zobaczyć jak wczesna detekcja wpływa na jakość predykcji.

W celu znalezienia architektury zapewniającej najlepszą jakość klasyfikacji zostały zbudowane trzy typy modeli, w oparciu o różne warstwy opisane w podrozdziale 2.1.

W każdym przypadku w architekturze można wyróżnić powtarzalne bloki składające się z tych samych następujących po sobie warstw. Wszystkie modele zbudowane są z serii bloków, przeplatanych warstwami łączącymi opisanymi w podrozdziale 2.1.4. Po wyjściu z ostatniego bloku reprezentacje wierzchołków trafiają do warstwy odczytu (ang.

readout layer). Następuje w niej przekształcenie reprezentacji z poziomu wierzchołków na poziom grafu przy pomocy funkcji agregujących, takich jak na przykład średnia, suma czy też mechanizm atencji [85]. Dopiero taka reprezentacja grafu jest podawana do bloku odpowiedzialnego za klasyfikację. W ostatniej warstwie znajduje się tylko jeden neuron, ponieważ funkcja kosztu to BCEWithLogitsLoss.

5.3.6.1. Algorytm GCNConv Pojedynczy blok dla tej architektury składa się z następujących warstw w kolejności: GCNConv 2.1.1, normalizacji wsadu oraz funkcji aktywacji ReLU.

Liczba neuronów ukrytych wybierana jest zarówno dla wejścia jak i wyjścia z warstwy GCNConv. Na wejściu do pierwszej warstwy w architekturze jest ona równa liczbie cech znajdujących się w zbiorze danych. Na wyjściu jest to liczba atrybutów nowej reprezentacji wektorowej wierzchołków (ang. node embeddings). Dane wyjściowe z pojedynczego bloku stanowią wejście do warstwy łączącej, w tym przypadku TopKPooling. Po przejściu przez wszystkie bloki otrzymana reprezentacja wierzchołków trafia do warstwy odczytu w której funkcjami agregującymi są global max pooling oraz global average pooling. Wartości tych funkcji są ze sobą sumowane i następnie podawane do bloku klasyfikacyjnego, który składa się z dwóch warstw liniowych przeplatanych funkcją aktywacji ReLU oraz dropoutem. Ostatnią warstwą modelu jest warstwa liniowa, z której wyjściem jest jeden neuron.

5.3.6.2. Algorytm GATConv Architektura zawierająca tę warstwę została zbudowana z takich samych elementów jak dla modelu GCNConv. Jednak obliczenia zostały na wczesnym etapie przerwane i porzucone przez brak poprawy jakości predykcji względem pierwszego modelu oraz przez ich czasochłonność.

5.3.6.3. Algorytm GraphSAGE Pojedynczy blok w tym modelu jest zbudowany z następujących po sobie warstw w kolejności: GraphSAGE 2.1.3, normalizacji wsadu oraz funkcji aktywacji ReLU. Liczba neuronów ukrytych na wejściu do pierwszej warstwy, tak jak w przypadku architektury GCNConv, jest równa liczbie atrybutów znajdujących się w zbiorze danych. Wyjściowa liczba neuronów ukrytych jest równa liczbie cech nowej reprezentacji wektorowej wierzchołków. Wyjście z bloku staje się wejściem do funkcji łączącej DiffPool, która przygotowuje reprezentacje grafu zawierającą 25% liczby wejściowych do tej warstwy wierzchołków. Segmenty składające się z dwóch bloków GraphSAGE oraz funkcji łączącej są powtarzane w tym przypadku dwa razy. Następnie reprezentacje wierzchołków są wejściem do warstwy odczytu. Wyjściem z warstwy odczytu jest reprezentacja grafu, która podawana jest to bloku klasyfikacyjnego składającego się z warstwy liniowej, funkcji aktywacji oraz ostatniej liniowej warstwy modelu.

6. Ocena uzyskanych wyników

Zgodnie z założeniami opisanymi w podrozdziale 5.1, modele były trenowane na pełnych kaskadach, oraz na częściowo zredukowanych odpowiednikach, tak, aby zbadać możliwości wczesnej detekcji fake newsów. Pojedynczą próbką wejściową do modelu są wszystkie kaskady dotyczące tego samego artykułu, połączone za pomocą głównego węzła, tak jak zostało to przedstawione na rysunku 3.3. Zagadnienie jest binarne, tak więc model uczy się poprawnego klasyfikowania artykułów na prawdziwe oraz fałszywe.

Został zastosowany predefiniowany podział zbiorów danych na zestaw treningowy, walidacyjny oraz testowy, aby mieć pewność, że jakość predykcji danego modelu jest niezależna od wylosowanego w danym momencie zestawu próbek. Podział nastąpił w taki sposób, aby w każdym zbiorze był taki sam rozkład klas. Takie same kroki zostały zastosowane niezależnie od tego, które podejście było testowane ani od tego w jaki sposób zostały przygotowane dane wejściowe.
Ograniczenia nałożone na zbiór danych, tak jak zostało to opisane w podrozdziale 5.1, to:

- W ujęciu czasowym:
  — pierwszych 5 min, 30 min, 1, 2, 4, 8, 12, 24 godziny oraz cała kaskada,
- w ujęciu ilościowym:
  — pierwszych 5, 10, 20, 30, 40, 50, 100 postów oraz cała kaskada.

Trening dla każdej architektury jest powtarzany 5 razy, a następnie jest wyciągana średnia wyników. Miara jakości predykcji klasyfikacji to miara F1 opisana w podrozdziale 2.4.

6.1. Modele klasyczne

Jest to podejście bazujące na artykule [13] opisanym w podrozdziale 5.2. Przedstawione niżej wyniki zostały uzyskane dla architektur opisanych w rozdziale 2.2, oraz metod zespołowych przedstawionych w podrozdziale 2.3. W tym drugim przypadku nie były wybierane nowe podzbiory klasyfikatorów do każdego treningu ponieważ takie obliczenia dla wszystkich możliwych kombinacji architektur były zbyt czasochłonne. Każdy zespół, w którym jest możliwość wyboru modeli bazowych został zbudowany ze wszystkich dostępnych klasyfikatorów.

Aby móc ocenić wpływ poszczególnych kategorii cech na uzyskiwane wyniki, treningi były przeprowadzane na następujących kombinacjach atrybutów:

Tabela 6.1. Wybór cech dla każdego treningu.

Lp	Cechy strukturalne	Cechy temporalne	Nowe cechy
1	+		
2		+	

3			+
4	+	+	
5	+		+
6		+	+
7	+	+	+

Uzyskane wyniki będą prezentowane w tabelach, w których zostaną pogrupowane i przedstawione w podziale na konkretne kombinacje cech na wejściu do modelu. Każdy wiersz będzie zawierał najwyższą uzyskaną miarę F1 dla konkretnej kombinacji atrybutów przy danym rodzaju ograniczenia. Dodatkowo, w celach referencyjnych zaprezentowane zostaną wynik modelu do którego wejściem były nieograniczone w żaden sposób całe kaskady.

6.1.1. Dane Politifact przygotowane wg. metody dołączania do oryginalnego tweeta

Pierwszym rozważanym zbiorem danych jest ten, w którym jeżeli nie ma możliwości bezpośredniego określenia potencjalnego źródła retweeta, to jest on przypisywany oryginalnemu tweetowi. Metoda rekonstrukcji danych została opisana w podrozdziale 3.7.1.

6.1.1.1. Zadanie wyboru najlepszego klasyfikatora

Tabela 6.2. Wybór najlepszego klasyfikatora - zbiór Politifact zrekonstruowany metodą opisaną w podrozdziale 3.7.1.

Lp	Cechy struktu-ralne	Cechy temporalne	Nowe Cechy	Przycięcie	Model	Wartość miary F1
1	+			24 godziny	k-NN	**0,742**
2	+			5 postów	SVM	0,721
3	+			brak	naiwny klas. Bayesa	0,704
4		+		12 godzin	naiwny klas. Bayesa	0,771
5		+		30 postów	regresja logistyczna	**0,778**
6		+		brak	Adaboost	0,768
7			+	24 godziny	naiwny klas. Bayesa	0,817
8			+	100 postów	AdaBoost	**0,846**
9			+	brak	Las losowy	0,821
10	+	+		12 godzin	regresja logistyczna	0,744
11	+	+		5 postów	SVM	**0,793**

12	+	+		brak	Las losowy	0,788
13	+		+	8 godzin	AdaBoost	0,800
14	+		+	100 postów	AdaBoost	**0,847**
15	+		+	brak	Las losowy	0,841
16		+	+	24 godziny	naiwny klas. Bayesa	0,800
17		+	+	100 postów	AdaBoost	0,862
18		+	+	brak	Las losowy	**0,878**
19	+	+	+	12 godzin	AdaBoost	0,800
20	+	+	+	100 postów	regresja logistyczna	0,859
21	+	+	+	brak	Las losowy	**0,878**

Analizując uzyskane wyniki można zauważyć, że w sytuacjach, w których wejściem do modelu jest tylko jeden rodzaj cech, to radzą sobie one lepiej na kaskadach, które są w pewien sposób ograniczone. Spośród cech strukturalnych, temporalnych oraz nowo zaproponowanych w tej pracy najlepsze wyniki są uzyskiwane dla tych ostatnich, ale za to na najdłuższym przedziale czasowym jak i największym ilościowym.

W przypadku połączenia cech strukturalnych oraz czasowych, już dla kaskad składających się tylko z pięciu elementów miara F1 osiąga wartość ok. 0,8. Można zauważyć, że najlepsze wyniki wcale nie są uzyskiwane na danych zawierających wszystkie możliwe rodzaje cech, tylko dla zestawienia atrybutów temporalnych oraz nowych cech, gdzie miara F1 przy ograniczeniu w postaci 100 pierwszych postów dla pojedynczej kaskady jest równa 0,862. Można zauważyć, że jest to wynik niższy niż możliwy do uzyskania na pełnych kaskadach, lecz różnica wynosi jedynie 1,6 punktu procentowego.

Klasyfikatorem, który występuje najczęściej w powyższej tabeli w przypadku przyciętych ilościowo kaskad jest AdaBoost, który nie był brany pod uwagę w oryginalnym artykule, tylko został zaproponowany do użycia w tej pracy. Z kolei dla danych ograniczonych czasowo gaussowski naiwny klasyfikator Bayesa pozwalał na uzyskanie najwyższych wyników.

Klasyfikator, który uzyskał najwyższą wartość miary F1 dla ograniczonego przedziału to algorytm Adaboost trenowany na kombinacji danych czasowych oraz nowych. Uzyskany wynik to 0,862 dla kaskad przyciętych do pierwszych 100 postów. Zgodnie z podejściem stosowanym przez autorów artykułu [13] została policzona ważność cech przy użyciu klasyfikatora Extra Trees wykorzystującego współczynnik Giniego (ang. gini impurity). Poniżej znajdują się trzy najbardziej wpływowe cechy w kolejności od najważniejszej:

- informacja o tym, czy konto danego użytkownika jest zweryfikowane czy nie (uśredniona dla całej próbki),
- liczba list do których należy użytkownik (uśredniona dla całej próbki),
- różnica czasu pomiędzy pierwszym tweetem a ostatnim retweetem.

6.1.1.2. Głosowanie większościowe

Tabela 6.3. Głosowanie większościowe - zbiór Politifact zrekonstruowany metodą opisaną w podrozdziale 3.7.1.

Lp	Cechy strukturalne	Cechy temporalne	Nowe Cechy	Przycięcie	Wartość miary F1
1	+			24 godziny	**0,752**
2	+			5 postów	0,691
3	+			brak	0,713
4		+		24 godziny	**0,765**
5		+		30 postów	0,750
6		+		brak	0,756
7			+	8 godzin	0,838
8			+	100 postów	**0,840**
9			+	brak	0,803
10	+	+		24 godziny	0,757
11	+	+		30 postów	0,775
12	+	+		brak	**0,783**
13	+		+	8 godzin	0,829
14	+		+	100 postów	**0,855**
15	+		+	brak	0,830
16		+	+	24 godziny	0,839
17		+	+	40 postów	**0,846**
18		+	+	brak	0,837
19	+	+	+	8 godzin	0,857
20	+	+	+	100 postów	0,872
21	+	+	+	brak	**0,874**

Wyraźnie rysuje się tendencja w której również w głosowaniu większościowym dla pojedynczych atrybutów uzyskiwane wyniki są wyższe z ograniczeniami ilościowymi i czasowymi, niż na pełnych kaskadach. Można także zauważyć, że najwyższą wartość miary F1 prezentują modele wytrenowane na nowych cechach zaproponowanych w tej pracy.

Spośród dostępnych kombinacji dwóch rodzajów atrybutów stanowiących wejścia do modeli najlepszy wynik został uzyskany na danych zawierających cechy strukturalne oraz nowe cechy przy ograniczeniu kaskad do maksymalnie 100 elementów. Warto jednak zwrócić uwagę, że niewiele gorzej wypada zestawienie atrybutów temporalnych z nowymi cechami, gdzie wynik jest niższy jedynie o niecały punkt procentowy, a najwyższa wartość miary F1 jest osiągana już przy 40 postach w kaskadzie, czyli na o wiele wcześniejszym

etapie rozprzestrzeniania się informacji w sieci, co jest korzystne z perspektywy jak najwcześniejszej detekcji fałszywych wiadomości.

Globalnie najlepszy wynik jest osiągany przy połączeniu wszystkich możliwych cech, co może sugerować, że taki rodzaj zespołowego uczenia potrzebuje jak najszerszego zakresu różnych atrybutów do uzyskania optymalnej jakości predykcji, ponieważ każdy pojedynczy klasyfikator może skupić uwagę na innym ich połączeniu.

W porównaniu z tabelą 6.2 można zauważyć, że głosowanie większościowe pozwala na osiągnięcie zbliżonych wyników do tych z pierwszego podejścia. Powodem może być na przykład to, że tylko mały podzbiór klasyfikatorów uzyskuje wysoką jakość predykcji, w związku z czym gorzej radzące sobie klasyfikatory wpływają negatywnie na wartości miary F1.

6.1.1.3. Generalizacja stosowa

Tabela 6.4. Generalizacja stosowa - zbiór Politifact zrekonstruowany metodą opisaną w podrozdziale 3.7.1.

Lp	Cechy strukturalne	Cechy temporalne	Nowe Cechy	Przycięcie	Wartość miary F1
1	+			24 godziny	**0,704**
2	+			5 postów	0,691
3	+			brak	0,675
4		+		24 godziny	0,748
5		+		30 postów	0,736
6		+		brak	**0,779**
7			+	24 godziny	0,812
8			+	100 postów	**0,840**
9			+	brak	0,818
10	+	+		24 godziny	0,745
11	+	+		100 postów	0,756
12	+	+		brak	**0,818**
13	+		+	24 godziny	0,825
14	+		+	40 postów	**0,862**
15	+		+	brak	0,842
16		+	+	8 godzin	0,87
17		+	+	100 postów	0,857
18		+	+	brak	**0,882**
19	+	+	+	24 godziny	0,809

20	+	+	+	100 postów	**0,871**
21	+	+	+	brak	0,871

Porównując powyższą tabelę z wynikami zawartymi w tabelach 6.3 oraz 6.2 widoczna jest poprawa w kilku przypadkach. Przede wszystkim dla połączenia atrybutów strukturalnych oraz nowych cech uzyskany wynik ma najwyższą wartość przy najbardziej restrykcyjnym ograniczeniu ilościowym - pierwszych 40 postów. Dodatkowo dla zestawienia cech temporalnych z nowymi również uzyskujemy bardzo wysoką wartość miary F1 równą 0,87 i to już dla kaskad ograniczonych tylko do pierwszych 8h propagacji. W analizowanym podejściu najlepszy wynik dla danych zawierających wszystkie dostępne cechy jest wyższy niż dla klasyfikatora zaprezentowanego w tabeli 6.2 oraz porównywalny do wartości miary F1 uzyskanej dla głosowania większościowego. W pozostałych przypadkach zaobserwowane wyniki są zbliżone z pozostałymi podejściami.

6.1.1.4. Dynamiczne dobieranie zespołu

Tabela 6.5. Dynamiczne dobieranie zespołu - zbiór Politifact zrekonstruowany metodą opisaną w podrozdziale 3.7.1.

Lp	Cechy strukturalne	Cechy temporalne	Nowe Cechy	Przycięcie	Wartość miary F1
1	+			24 godziny	**0,696**
2	+			5 postów	0,686
3	+			brak	0,617
4		+		24 godziny	0,71
5		+		40 postów	0,665
6		+		brak	**0,734**
7			+	12 godzin	0,79
8			+	100 postów	**0,798**
9			+	brak	0,764
10	+	+		24 godziny	0,734
11	+	+		100 postów	0,686
12	+	+		brak	**0,762**
13	+		+	24 godziny	**0,803**
14	+		+	40 postów	0,788
15	+		+	brak	0,786
16		+	+	24 godziny	**0,81**
17		+	+	100 postów	0,803

18		+	+	brak	0,799
19	+	+	+	24 godziny	**0,823**
20	+	+	+	40 postów	0,765
21	+	+	+	brak	0,817

Wyniki zaprezentowane w powyższej tabeli pozwalają na zauważenie, że to podejście nie prowadzi do uzyskania lepszej jakości klasyfikacji dla żadnej z analizowanych kombinacji cech.

6.1.2. Dane Politifact przygotowane wg. metody dołączania do posta najpopularniejszego użytkownika

W tym podrozdziale zbiór danych różni się od poprzednio analizowanego w obszarze postów dla których nie da się wytypować źródła. Jeżeli dany użytkownik udostępniający retweeta nie jest powiązany obserwowaniem z żadnym z kont, które wcześniej udostępniły dany post, to zakłada się, że źródłem tego retweeta jest tweet albo retweet pochodzący od użytkownika, który ma największą liczbę obserwujących. Metoda rekonstrukcji danych została opisana w podrozdziale 3.7.2.

6.1.2.1. Zadanie wyboru najlepszego klasyfikatora

Tabela 6.6. Wybór najlepszego klasyfikatora - zbiór Politifact zrekonstruowany metodą opisaną w podrozdziale 3.7.2.

Lp	Cechy strukturalne	Cechy temporalne	Nowe Cechy	Przycięcie	Model	Wartość miary F1
1	+			24 godziny	Adaboost	**0,766**
2	+			5 postów	Las losowy	0,725
3	+			brak	naiwny klas. Bayesa	0,708
4		+		12 godzin	SVM	0,771
5		+		30 postów	regresja logistyczna	0,767
6		+		brak	Adaboost	**0,779**
7			+	24 godziny	naiwny klas. Bayesa	0,83
8			+	100 postów	Las losowy	**0,840**
9			+	brak	Las losowy	0,821
10	+	+		12 godzin	SVM	0,765
11	+	+		5 postów	SVM	0,78
12	+	+		brak	Adaboost	**0,792**

13	+		+	2 godziny	regresja logistyczna	0,8
14	+		+	100 postów	Las losowy	0,832
15	+		+	brak	Las losowy	**0,841**
16		+	+	24 godziny	naiwny klas. Bayesa	0,833
17		+	+	100 postów	Las losowy	0,843
18		+	+	brak	Las losowy	**0,872**
19	+	+	+	12 godzin	regresja logistyczna	0,819
20	+	+	+	100 postów	AdaBoost	0,859
21	+	+	+	brak	Las losowy	**0,863**

Porównując powyższą tabelę 6.6 z tabelą 6.2 można zauważyć, że w większości przypadków uzyskane wyniki mają porównywalne wartości z różnicami oscylującymi w okolicy jednego punktu procentowego. Jedynie dla połączenia atrybutów temporalnych oraz nowych cech przy ograniczeniu czasowym, w analizowanym podejściu wynik jest wyższy o około 3 punkty procentowe.

Warto też zwrócić uwagę, że w przypadku kombinacji atrybutów strukturalnych oraz nowych cech w podejściu czasowym uzyskany wynik jest równy temu z tabeli 6.2, ale na znacznie mniejszych kaskadach, bo zawierających posty jedynie z pierwszych dwóch godzin. Najczęściej pojawiającym się klasyfikatorem w powyższej tabeli jest las losowy.

Klasyfikatorem, który pozwolił na uzyskanie najwyższej wartości miary F1 dla przedziału zawierającego pewnego rodzaju ograniczenie jest ponownie algorytm Adaboost, przy czym w tym przypadku wejścia zawierały wszystkie dostępne kategorie cech. Uzyskany wynik to 0,859 dla kaskad przyciętych do pierwszych 100 postów.

Tak jak w przypadku analizy wyników zawartych w tabeli 6.2 została policzona ważność cech. Poniżej znajdują się trzy najbardziej wpływowe atrybuty w kolejności od najważniejszego:

- informacja o tym, czy konto danego użytkownika jest zweryfikowane czy nie (uśredniona dla całej próbki),
- liczba list do których należy użytkownik (uśredniona dla całej próbki),
- różnica czasu pomiędzy pierwszym i ostatnim tweetem,

Można zauważyć, że dwie pierwsze cechy pokrywają się z tymi z poprzedniej analizy.

6.1.2.2. Głosowanie większościowe

Tabela 6.7. Głosowanie większościowe - zbiór Politifact zrekonstruowany metodą opisaną w podrozdziale 3.7.2.

Lp	Cechy strukturalne	Cechy temporalne	Nowe Cechy	Przycięcie	Wartość miary F1
1	+			8 godzin	0,711
2	+			5 postów	**0,741**
3	+			brak	0,679
4		+		24 godziny	**0,78**
5		+		30 postów	0,777
6		+		brak	0,765
7			+	8 godzin	0,841
8			+	100 postów	**0,868**
9			+	brak	0,786
10	+	+		24 godziny	0,768
11	+	+		30 postów	0,761
12	+	+		brak	**0,79**
13	+		+	5 minut	0,842
14	+		+	100 postów	**0,88**
15	+		+	brak	0,806
16		+	+	24 godziny	0,821
17		+	+	100 postów	**0,87**
18		+	+	brak	0,848
19	+	+	+	8 godzin	**0,879**
20	+	+	+	100 postów	0,87
21	+	+	+	brak	0,857

Zestawiając powyższe wyniki z tabelą 6.3 można zauważyć, że przy bieżącym sposobie łączenia użytkowników w grafy uzyskane wartości miary F1 są w większości porównywalne, z niewielkimi wahaniami w obrębie kilku punktów procentowych.

Warto jednak zwrócić szczególną uwagę na przypadek kombinacji dwóch cech, a dokładniej atrybutów strukturalnych oraz nowych cech. Uzyskany wynik dla kaskad ograniczonych czasowo jest wyższy o około 1,3 punktu procentowego od tego z tabeli 6.3. Jednak wartość miary F1 równa 0,842 jest osiągana dla kaskad zawierających elementy udostępnione jedynie w ciągu pierwszych 5 minut, co jest bardzo korzystnym wynikiem z perspektywy wczesnej detekcji fake newsów.

6.1.2.3. Generalizacja stosowa

Tabela 6.8. Generalizacja stosowa - zbiór Politifact zrekonstruowany metodą opisaną w podrozdziale 3.7.2.

Lp	Cechy strukturalne	Cechy temporalne	Nowe Cechy	Przycięcie	Wartość miary F1
1	+			4 h	**0,743**
2	+			100 postów	0,695
3	+			brak	0,645
4		+		12 godzin	0,771
5		+		100 postów	0,721
6		+		brak	**0,783**
7			+	12 godzin	0,835
8			+	40 postów	**0,864**
9			+	brak	0,832
10	+	+		12 godzin	0,747
11	+	+		100 postów	0,734
12	+	+		brak	**0,787**
13	+		+	12 godzin	0,833
14	+		+	50 postów	**0,839**
15	+		+	brak	0,827
16		+	+	12 godzin	0,849
17		+	+	100 postów	**0,867**
18		+	+	brak	0,835
19	+	+	+	8 godzin	0,812
20	+	+	+	100 postów	0,855
21	+	+	+	brak	**0,871**

Porównując tabele 6.8 oraz 6.4 można zauważyć, że ponownie wyniki w większości przypadków są bardzo zbliżone. Zauważalnie lepsze wartości miary F1 zostały uzyskane dla pojedynczych typów atrybutów stanowiących wejścia do klasyfikatorów. W przypadku cech strukturalnych z ograniczeniem czasowym, jakość predykcji modelu jest wyższa o około 4 punkty procentowe i osiąga swoje maksimum już dla grafów przyciętych do pierwszych 4 godzin propagacji, gdzie w poprzednim podejściu następowało to dopiero po 24 godzinach. W przypadku cech temporalnych z ograniczeniem czasowym lepsze wyniki są osiągana już dla 12 godzin i są wyższe o około 2,3 punktu procentowego. Dla podejścia, w którym tylko nowe cechy stanowią wejście do modelu osiągana jest wyższa o

około 2,4 punktu procentowego wartość miary F1 już dla pierwszych 40 postów, gdzie w poprzednim podejściu najlepszy wynik był uzyskiwany dla 100 postów.

Również kiedy zestaw cech wejściowych zawiera wszystkie możliwe atrybuty uzyskanie miary F1 na podobnym poziomie jest możliwe dla kaskad ograniczonych czasowo do 12 godzin, zamiast 24.

6.1.2.4. Dynamiczne dobieranie zespołu

Tabela 6.9. Dynamiczne dobieranie zespołu - zbiór Politifact zrekonstruowany metodą opisaną w podrozdziale 3.7.2.

Lp	Cechy strukturalne	Cechy temporalne	Nowe Cechy	Przycięcie	Wartość miary F1
1	+			24 godziny	**0,69**
2	+			10 postów	0,683
3	+			brak	0,591
4		+		24 godziny	0,662
5		+		30 postów	0,696
6		+		brak	**0,742**
7			+	24 godzin	0,789
8			+	100 postów	**0,797**
9			+	brak	0,766
10	+	+		24 godziny	0,708
11	+	+		30 postów	0,687
12	+	+		brak	**0,749**
13	+		+	24 godziny	0,768
14	+		+	100 postów	0,778
15	+		+	brak	**0,787**
16		+	+	24 godziny	0,744
17		+	+	100 postów	**0,813**
18		+	+	brak	0,81
19	+	+	+	24 godziny	0,783
20	+	+	+	40 postów	0,799
21	+	+	+	brak	**0,813**

Ponownie, tak jak w poprzednim przypadku metoda dynamicznego doboru zepsołu nie prowadzi do poprawy jakości predykcji.

6.1.3. Dane Gossipcop przygotowane wg. metody dołączania do oryginalnego tweeta

Zbiór danych został przygotowany w taki sam sposób jak w podrozdziale 6.1.1, czyli w przypadku braku możliwości określenia potencjalnego źródła retweeta jest on dołączany do oryginalnego tweeta. Metoda rekonstrukcji danych została opisana w podrozdziale 3.7.1.

6.1.3.1. Zadanie wyboru najlepszego klasyfikatora

Tabela 6.10. Wybór najlepszego klasyfikatora - zbiór Gossipcop zrekonstruowany metodą opisaną w podrozdziale 3.7.1.

Lp	Cechy strukturalne	Cechy temporalne	Nowe Cechy	Przycięcie	Model	Wartość miary F1
1	+			24 godziny	Adaboost	0,828
2	+			5 postów	Drzewo decyzyjne	**0,871**
3	+			brak	Adaboost	0,816
4		+		24 godziny	Las losowy	0,825
5		+		5 postów	SVM	**0,869**
6		+		brak	Las losowy	0,85
7			+	24 godziny	SVM	0,954
8			+	100 postów	Las losowy	0,971
9			+	brak	Las losowy	**0,972**
10	+	+		24 godziny	Las losowy	0,855
11	+	+		5 postów	Adaboost	**0,884**
12	+	+		brak	Las losowy	0,847
13	+		+	24 godziny	SVM	0,957
14	+		+	100 postów	Las losowy	**0,971**
15	+		+	brak	Las losowy	0,971
16		+	+	24 godziny	SVM	0,955
17		+	+	100 postów	Las losowy	**0,968**
18		+	+	brak	Las losowy	0,968
19	+	+	+	24 godziny	SVM	0,958
20	+	+	+	100 postów	Las losowy	**0,968**
21	+	+	+	brak	Las losowy	0,967

Warto zwrócić uwagę na fakt, że klasyfikatory wytrenowane na ograniczonych ilościowo kaskadach dla cech strukturalnych, czasowych oraz połączenia obu osiągały najwyższe wartości miary F1 już po 5 pierwszych postach. Tak dobre wyniki dla najbardziej

restrykcyjnego ograniczenia są bardzo korzystne z perspektywy wczesnej detekcji fałszywych wiadomości.

Analizując powyższą tabelę można zauważyć, że w przypadku ograniczenia czasowego najwyższe wartości miar F1 były osiągane w maksymalnym rozważanym przedziale czasu (24 godziny). Jednak wyniki uzyskiwane dla bardziej restrykcyjnie ograniczonych czasowo danych nie odbiegały aż tak znacząco od tych zaprezentowanych powyżej. Przykładowo, dla zestawu atrybutów zawierającego jedynie nowe cechy już dla pierwszych 5 minut rozpowszechniania komunikatów gaussowski naiwny klasyfikator Bayesa pozwalał na osiągnięcie wartości miary F1 równej 0,82, a biorąc pod uwagę 8 pierwszych godzin propagacji algorytm Adaboost zwracał wyniki równe 0,9.

Dla zestawu danych zawierającego połączone atrybuty strukturalne z nowymi cechami po pierwszych dwóch godzinach udostępniania komunikatów miara F1 dla drzewa losowego wynosiła 0,837, a po pierwszych 4 godzinach dla algorytmu k najbliższych sąsiadów 0,866.

Rozważając wszystkie dostępne cechy, po dwóch godzinach drzewo decyzyjne zwracało miarę F1 równą 0,858.

Podobną analizę można przeprowadzić w przypadkach, w których najlepsze wyniki były osiągane dla maksymalnego ograniczenia ilościowego (100 postów). Dla nowych cech już na kaskadach przyciętych do 50 postów osiągana miara F1 była równa 0,964, a dla połączenia atrybutów czasowych oraz nowych cech algorytm Adaboost pozwalał na osiągnięcie wyników równych 0.963 dla pierwszych 40 komunikatów.

Można również generalnie zauważyć, że wyniki dla tego zbioru danych są w większości wyższe niż dla zestawu opartego o portal Politifact. Źródłem tych różnic może być większa ilość dostępnych próbek. Dodatkowo klasyfikatorem, który najczęściej pozwalał na osiągnięcie najwyższych wyników jest las losowy.

Najwyższa wartość miary F1 dla przyciętych kaskad została uzyskana dla lasu losowego, do którego wejściem były dane zawierające jedynie nowe cechy. Dla 100 pierwszych postów wynik jest równy 0,971. Zgodnie z podejściem stosowanym przez autorów artykułu [13] została policzona ważność cech przy użyciu klasyfikatora Extra Trees wykorzystującego współczynnik Giniego (ang. gini impurity). Poniżej znajdują się trzy najbardziej wpływowe cechy w kolejności od najważniejszej:

- współczynnik agregujący aktywność użytkownika w okresie poprzedzającym dodanie przez niego konkretnego posta - tweeta lub retweeta (uśredniony dla całej próbki),
- uśredniona reputacja konta,
- informacja o tym, czy konto danego użytkownika jest zweryfikowane czy nie (uśredniona dla całej próbki).

Dwie pierwsze cechy są nowymi atrybutami, pierwszy z nich jest współczynnikiem

stworzonym na potrzeby tej pracy, a drugi z nich został zaproponowany na podstawie przeglądu literatury dla podobnych zagadnień badawczych.

6.1.3.2. Głosowanie większościowe

Tabela 6.11. Głosowanie większościowe - zbiór Gossipcop zrekonstruowany metodą opisaną w podrozdziale 3.7.1.

Lp	Cechy strukturalne	Cechy temporalne	Nowe Cechy	Przycięcie	Wartość miary F1
1	+			24 godziny	0,824
2	+			5 postów	**0,869**
3	+			brak	0,814
4		+		24 godziny	0,82
5		+		5 postów	**0,864**
6		+		brak	0,793
7			+	24 godziny	0,957
8			+	100 postów	**0,962**
9			+	brak	0,959
10	+	+		24 godziny	0,842
11	+	+		5 postów	**0,879**
12	+	+		brak	0,845
13	+		+	24 godziny	0,957
14	+		+	100 postów	**0,963**
15	+		+	brak	0,954
16		+	+	24 godziny	0,945
17		+	+	40 postów	**0,962**
18		+	+	brak	0,957
19	+	+	+	24 godziny	0,955
20	+	+	+	100 postów	**0,963**
21	+	+	+	brak	0,955

W tym przypadku również można zauważyć, że przy ograniczeniu czasowym najwyższe wartości były osiągane dla przedziału zawierającego 24 godziny propagacji. Jednak biorąc pod uwagę jedynie nowe cechy, już po 4 godzinach udostępniania komunikatów miara F1 była równa 0,931.

W przypadku zestawienia atrybutów strukturalnych oraz nowych cech, po 12 godzinach miara F1 osiągała wartość 0,951. Dla cech temporalnych oraz nowych po 4 godzinach wyniki były równe 0,935.

Z kolei zestaw danych zawierający wszystkie możliwe cechy i ograniczony do pierwszych 8 godzin zwracał wartość miary F1 równą 0,933.

Rozważając ograniczenie ilościowe w przypadkach, w których najlepsze wyniki były osiągane na maksymalnym przedziale warto zauważyć, że na przykład dla nowych cech wartość miary F1 wynosiła 0,948 już dla pierwszych 10 postów, a dla 40 postów rosła do wartości 0,961.

Dane zawierające zestaw atrybutów strukturalnych połączonych z nowymi cechami pozwalał na osiągnięcie wyników rzędu 0,952 dla 20 pierwszych postów.

W każdym opisanym powyżej przypadku są to wartości wyższe niż dla rozważanego wcześniej zbioru danych portalu Politifact. W porównaniu z zagadnieniem wyboru najlepszego klasyfikatora rozważane podejście nie daje możliwości osiągnięcia znacznej poprawy wyników.

6.1.3.3. Generalizacja stosowa

Tabela 6.12. Generalizacja stosowa - zbiór Gossipcop zrekonstruowany metodą opisaną w podrozdziale 3.7.1.

Lp	Cechy strukturalne	Cechy temporalne	Nowe Cechy	Przycięcie	Wartość miary F1
1	+			24 godziny	0,832
2	+			5 postów	**0,871**
3	+			brak	0,818
4		+		24 godziny	0,821
5		+		5 postów	**0,868**
6		+		brak	0,844
7			+	24 godziny	0,958
8			+	100 postów	**0,973**
9			+	brak	0,973
10	+	+		24 godziny	0,859
11	+	+		10 postów	**0,882**
12	+	+		brak	0,851
13	+		+	24 godziny	0,955
14	+		+	100 postów	0,972
15	+		+	brak	**0,973**
16		+	+	24 godziny	0,953
17		+	+	100 postów	0,967
18		+	+	brak	0,966

19	+	+	+	24 godziny	0,95
20	+	+	+	100 postów	0,968
21	+	+	+	brak	**0,969**

Podobnie jak w poprzednich przypadkach, tutaj również najwyższe wartości miary F1 są osiągane dla maksymalnego przedziału czasowego. Jednak dla zestawu zawierającego jedynie nowe cechy już po pierwszych 4 godzinach propagacji wyniki są równe 0,933. Dla połączenia atrybutów strukturalnych oraz nowych cech również dla kaskad zawierających komunikaty pochodzące tylko i wyłącznie z pierwszych 4 godzin propagacji wartość miary F1 jest równa 0,936. Zestawienie atrybutów czasowych oraz nowych cech oferuje zbliżone wyniki dla tego samego przedziału czasu, wynoszące 0,934. Model wytrenowany na zestawie danych zawierającym wszystkie dostępne atrybuty osiąga wyniki 0,938 po 8 godzinach propagacji.

Rozważając ograniczenia ilościowe warto zauważyć, że w przypadku nowych cech model osiąga wartość miary F1 równą 0,938 dla kaskad składających się z pierwszych 5 postów, a przy 20 komunikatach wyniki rosną do 0,953. Podobne wartości są osiągane dla zestawienia atrybutów strukturalnych z nowymi cechami, oraz temporalnych również z nowymi cechami. Na tej podstawie można wyciągnąć wniosek, że dodawanie pozostałych typów atrybutów do nowych cech nie prowadzi do poprawy jakości klasyfikacji.

W porównaniu z wynikami zaprezentowanymi w tabelach 6.11 oraz 6.10 generalizacja stosowa nie pozwala na osiągnięcie znacząco wyższych wartości miary F1.

6.1.3.4. Dynamiczne dobieranie zespołu

Tabela 6.13. Dynamiczne dobieranie zespołu - zbiór Gossipcop zrekonstruowany metodą opisaną w podrozdziale 3.7.1.

Lp	Cechy strukturalne	Cechy temporalne	Nowe Cechy	Przycięcie	Wartość miary F1
1	+			24 godziny	0,785
2	+			5 postów	**0,871**
3	+			brak	0,733
4		+		24 godziny	0,807
5		+		5 postów	**0,85**
6		+		brak	0,813
7			+	24 godziny	0,892
8			+	50 postów	0,958
9			+	brak	**0,963**

10	+	+		24 godziny	0,813
11	+	+		5 postów	**0,854**
12	+	+		brak	0,824
13	+		+	24 godziny	0,922
14	+		+	100 postów	0,949
15	+		+	brak	**0,96**
16		+	+	12 godzin	0,922
17		+	+	30 postów	0,948
18		+	+	brak	0,955
19	+	+	+	8 godzin	0,919
20	+	+	+	50 postów	0,95
21	+	+	+	brak	**0,96**

Podobnie jak w przypadku analizy wyników dla zbiorów opartych o portal Politifact, tak również tutaj algorytm dynamicznego dobierania zespołu nie oferuje znaczącej poprawy wartości miary F1.

6.1.4. Dane Gossipcop przygotowane wg. metody dołączania do posta najpopularniejszego użytkownika

Zbiór danych został przygotowany w taki sam sposób jak w podrozdziale 6.1.2, czyli w przypadku braku możliwości określenia potencjalnego źródła retweeta jest on dołączany do retweeta pochodzącego od użytkownika, który ma największą liczbę obserwujących. Metoda rekonstrukcji danych została opisana w podrozdziale 3.7.2.

6.1.4.1. Zadanie wyboru najlepszego klasyfikatora

Tabela 6.14. Wybór najlepszego klasyfikatora - zbiór Gossipcop zrekonstruowany metodą opisaną w podrozdziale 3.7.2.

Lp	Cechy struktu-ralne	Cechy temporalne	Nowe Cechy	Przycięcie	Model	Wartość miary F1
1	+			24 godziny	Regresja logistyczna	0,835
2	+			5 postów	Adaboost	**0,871**
3	+			brak	AdaBoost	0,818
4		+		24 godziny	AdaBoost	0,824
5		+		5 postów	Adaboost	**0,875**
6		+		brak	Las losowy	0,849

7			+	24 godziny	Las losowy	0,96
8			+	100 postów	Las losowy	0,968
9			+	brak	Las losowy	**0,973**
10	+	+		24 godziny	Las losowy	0,861
11	+	+		5 postów	k-NN	**0,882**
12	+	+		brak	Las losowy	0,85
13	+		+	24 godziny	SVM	0,956
14	+		+	100 postów	Las losowy	**0,97**
15	+		+	brak	Las losowy	0,97
16		+	+	24 godziny	Las losowy	0,957
17		+	+	100 postów	Las losowy	0,967
18		+	+	brak	Las losowy	**0,969**
19	+	+	+	24 godziny	Las losowy	0,961
20	+	+	+	50 postów	Las losowy	0,968
21	+	+	+	brak	Las losowy	**0,969**

Tak jak w przypadku wyników z tabeli 6.10 w powyższej również można zaobserwować, że najlepsze wyniki dla ograniczenia czasowego były uzyskiwane po 24 godzinach rozpowszechniania komunikatów.

Jednak rozważając na przykład dane zawierające tylko nowe cechy po pierwszych 5 minutach gaussowski naiwny klasyfikator Bayesa osiąga wartość miary F1 wynoszącą 0,816, a po 8h algorytm AdaBoost zwraca wynik równy 0,916.

Drzewo decyzyjne wytrenowane na zestawie danych zawierającym połączone atrybuty temporalne z nowymi cechami zwraca wartości miary F1 0,853 po pierwszej godzinie propagacji, a po 12 godzinach dla maszyny wektorów nośnych ta wartość rośnie do 0,952 czyli wyniku jedynie o 0,2 punktu procentowego niższego niż dla najlepszego klasyfikatora z tabeli.

Podobnie jak dla ograniczeń czasowych, również najlepsze wyniki dla redukcji liczby postów były najczęściej osiągane dla maksymalnego ograniczenia w tym przedziale.

W przypadku nowych cech najwyższa wartość miary F1 została zwrócona przez las losowy dla 100 postów. Jednak już dla kaskad składających się z 40 komunikatów algorytm Adaboost pozwala na osiągnięcie wyników na poziomie 0,957, a dla 10 postów gaussowski naiwny klasyfikator Bayesa zwraca wartość 0,913.

Gdy brane są pod uwagę tylko cechy strukturalne i nowe cechy SVM dla pierwszych 50 postów zwraca miarę F1 równą 0,968, czyli tylko o 0,2 punktu procentowego niższą niż dla najlepszego klasyfikatora.

Porównując powyższe wyniki z tymi przedstawionymi w tabeli 6.10 można zauważyć,

że uzyskane wartości są bardzo zbliżone. Na tej podstawie pojawia się wniosek mówiący o tym, że obie metody przygotowania danych są równoważne.

Wartość miary F1 dla klasyfikatora charakteryzującego się najlepszą jakością predykcji na ograniczonych kaskadach wynosi 0,97. Tym modelem jest las losowy wytrenowany na danych zawierających atrybuty strukturalne oraz nowe cechy. Tak jak dla poprzednich analiz wyników dla zadania wyboru najlepszego klasyfikatora, tu również została policzona ważność cech przy użyciu klasyfikatora Extra Trees wykorzystującego współczynnik Giniego (ang. gini impurity). Poniżej znajdują się trzy najbardziej wpływowe cechy w kolejności od najważniejszej:

- współczynnik agregujący aktywność użytkownika w okresie poprzedzającym dodanie przez niego konkretnego posta - tweeta lub retweeta (uśredniony dla całej próbki),
- uśredniona reputacja konta,
- informacja o tym, czy konto danego użytkownika jest zweryfikowane czy nie.

Można zauważyć, że są one takie same jak dla drugiej metody rekonstrukcji danych.

6.1.4.2. Głosowanie większościowe

Tabela 6.15. Głosowanie większościowe - zbiór Gossipcop zrekonstruowany metodą opisaną w podrozdziale 3.7.2.

Lp	Cechy strukturalne	Cechy temporalne	Nowe Cechy	Przycięcie	Wartość miary F1
1	+			24 godziny	0,837
2	+			5 postów	**0,871**
3	+			brak	0,824
4		+		24 godziny	0,817
5		+		5 postów	**0,876**
6		+		brak	0,8
7			+	24 godziny	0,957
8			+	100 postów	**0,966**
9			+	brak	0,966
10	+	+		24 godziny	0,849
11	+	+		5 postów	**0,882**
12	+	+		brak	0,85
13	+		+	24 godziny	0,959
14	+		+	100 postów	**0,967**
15	+		+	brak	0,965

16		+	+	24 godziny	0,956
17		+	+	100 postów	0,957
18		+	+	brak	**0,965**
19	+	+	+	24 godziny	0,963
20	+	+	+	50 postów	0,957
21	+	+	+	brak	**0,963**

W zagadnieniu głosowania większościowego również można zauważyć, że przy ograniczeniu czasowym maksymalne wartości miary F1 były osiągane dla najdłuższego przedziału - zawierającego 24 godziny propagacji komunikatów. Poniżej zostaną przedstawione wybrane wyniki dla krótszych okresów czasu.

W przypadku pojedynczych kategorii atrybutów wyniki wyższe niż 0,9 zostały uzyskane jedynie dla danych zawierających nowe cechy. Po pierwszych 5 minutach propagacji miara F1 wynosiła 0,86 i dla kaskad zawierających komunikaty pochodzące z okresu 4 godzin rosła do wartości 0,919.

Dla kombinacji atrybutów strukturalnych oraz nowych cech po pierwszych dwóch godzinach rozpowszechniania wiadomości miara F1 wynosiła 0,911. A w przypadku połączenia cech temporalnych z nowymi wyniki dla okresu pierwszych 30 minut są równe 0,89 i rosną do wartości 0,913 dla kaskad zawierających komunikaty powstałe w przeciągu pierwszej godziny.

W przypadku ograniczenia ilościowego również można zauważyć, że dla większości kombinacji atrybutów maksymalne wartości miary F1 są osiągane na najliczniejszym przedziale - 100 postach.

Model wytrenowany na danych zawierających kaskady ograniczone do 5 pierwszych oraz jedynie nowe cechy osiąga wyniki równe 0,932. Kombinacja atrybutów strukturalnych oraz nowych cech pozwala na osiągnięcie na tym przedziale wartości miary F1 równej 0,929, której wartość wzrasta do 0,945 już dla 10 postów. Podobne wartości są uzyskiwane dla połączenia atrybutów czasoprzestrzennych oraz nowych cech.

6.1.4.3. Generalizacja stosowa

Tabela 6.16. Generalizacja stosowa - zbiór Gossipcop zrekonstruowany metodą opisaną w podrozdziale 3.7.2.

Lp	Cechy strukturalne	Cechy temporalne	Nowe Cechy	Przycięcie	Wartość miary F1
1	+			24 godziny	0,842
2	+			5 postów	**0,869**
3	+			brak	0,819

4		+		24 godziny	0,827
5		+		10 postów	**0,873**
6		+		brak	0,856
7			+	24 godziny	0,958
8			+	100 postów	**0,97**
9			+	brak	0,964
10	+	+		24 godziny	0,864
11	+	+		5 postów	**0,882**
12	+	+		brak	0,853
13	+		+	24 godziny	0,961
14	+		+	100 postów	**0,971**
15	+		+	brak	0,965
16		+	+	24 godziny	0,955
17		+	+	100 postów	**0,969**
18		+	+	brak	0,965
19	+	+	+	24 godziny	0,958
20	+	+	+	100 postów	**0,97**
21	+	+	+	brak	0,96

Porównując powyższe wyniki z tymi przedstawionymi w tabeli 6.12 nie widać znaczącej poprawy wynikającej z faktu wykorzystania innej techniki rekonstrukcji danych.

6.1.4.4. Dynamiczne dobieranie zespołu

Tabela 6.17. Dynamiczne dobieranie zespołu - zbiór Gossipcop zrekonstruowany metodą opisaną w podrozdziale 3.7.2.

Lp	Cechy strukturalne	Cechy temporalne	Nowe Cechy	Przycięcie	Wartość miary F1
1	+			24 godziny	0,798
2	+			5 postów	**0,865**
3	+			brak	0,781
4		+		24 godziny	0,798
5		+		5 postów	**0,856**
6		+		brak	0,827
7			+	24 godziny	0,928
8			+	50 postów	0,954

9			+	brak	**0,963**
10	+	+		24 godziny	0,832
11	+	+		20 postów	**0,862**
12	+	+		brak	0,828
13	+		+	24 godziny	0,949
14	+		+	50 postów	0,956
15	+		+	brak	**0,959**
16		+	+	24 godziny	0,924
17		+	+	40 postów	0,948
18		+	+	brak	**0,962**
19	+	+	+	24 godziny	0,938
20	+	+	+	50 postów	**0,958**
21	+	+	+	brak	0,958

W porównaniu z tabelą 6.14 zawierającą wyniki dla zadania wyboru najlepszego klasyfikatora można zauważyć, że dynamiczne dobieranie zespołu nie przynosi poprawy dla kaskad ograniczonych czasowo. Biorąc pod uwagę zmniejszanie rozmiaru ilościowo można zauważyć, że największe wartości miar F1 były osiągane dla mniejszej liczby postów, średnio o połowę wcześniej.

6.2. Grafowe sieci neuronowe

Wyniki opisane poniżej zostały uzyskane dla podejścia bazującego na artykule [17], który został szerzej omówiony w podrozdziale 5.3.

Rezultaty obliczeń zostaną zaprezentowane tylko i wyłącznie dla jednej metody rekonstrukcji kaskad komunikacyjnych. Jest to następstwem wniosków wyciągniętych na podstawie poprzedniego podrozdziału, gdzie nie udało się zaobserwować znaczącej różnicy pomiędzy uzyskiwanymi wartościami miary F1 dla dwóch rozważanych podejść. Kaskady przygotowano według metody przedstawionej w artykule [13] i opisanej w podrozdziale 3.7.1.

Poniższe wyniki zostały uzyskane dla modeli, których architektury przedstawiono w podrozdziale 5.3.6. Dla każdej wartości ograniczenia czasowego oraz ilościowego został wytrenowany oddzielny model. Hiperparametry były dobierane drogą ich optymalizacji przy pomocy biblioteki optuna [76]. Ogólne informacje związane z ustawieniami treningu oraz przygotowaniem danych do postaci wejścia do modelu zostały opisane w 5.3.4 oraz 5.3.5. Dla zbioru Politifact wartości miar F1 przedstawione w poniższych tabelach są średnią wyniku z 5 treningów. Z kolei dla zbioru Gossipcop, przez wzgląd na czasochłonność pojedynczego treningu oraz ilość różnych podejść wymagających przetestowania, są to wyniki dla jednego, najlepszego modelu.

6.2.1. Dane Politifact przygotowane wg. metody dołączania do oryginalnego tweeta

Tabela 6.18. Porównanie wyników uzyskanych dla architektur opartych o algorytmy GCNConv oraz GraphSAGE - zbiór Politifact zrekonstruowany metodą opisaną w podrozdziale 3.7.1.

Lp	Przycięcie	GCNConv - wartość miary F1	GraphSAGE - wartość miary F1
1	5 minut	0,771	0,752
2	30 minut	0,772	0,755
3	1 godzina	0,855	
4	2 godziny	**0,895**	0,756
5	4 godziny	0,816	0,765
6	8 godzin	0,821	0,82
7	12 godzin	0,851	0,83
8	24 godziny	0,877	
9	5 postów	**0,894**	0,839
10	10 postów	0,881	0,824
11	20 postów	0,813	0,832
12	30 postów	0,838	0,841
13	40 postów	0,809	0,815
14	50 postów	0,818	0,822
15	100 postów	0,861	**0,876**

W powyższej tabeli zostało zawarte porównanie wyników uzyskanych dla dwóch architektur - jednej opartej o algorytm GCNConv i warstwę łączącą TopKPooling oraz drugiej bazującej na algorytmie GraphSAGE oraz warstwie łączącej DiffPool.

Zestawiając ze sobą wyniki dla kolejnych ograniczeń można zauważyć, że w przypadku, w którym kaskady przycinane są na podstawie czasu upływającego od momentu dodania pierwszego tweeta w danym temacie, modele bazujące na algorytmie GCNConv generalnie pozwalały na osiągnięcie lepszych wyników.

Najwyższa wartość miary F1 została uzyskana dla danych zawierających grafy ograniczone do dwóch pierwszych godzin propagacji i wynosi 0,895. Jest to wynik wyższy niż jakikolwiek inny uzyskany metodami klasycznymi dla ograniczenia czasowego na zbiorze Politifact, opisanymi w podrozdziale 6.1. Klasyfikatorem charakteryzującym się najlepszą jakością predykcji jest algorytm głosowania większościowego, który był trenowany na danych zawierających wszystkie dostępne cechy (tabela 6.7). Uzyskana miara F1 wynosi 0,879. Różnica nie jest bardzo duża (1,6 punktu procentowego) lecz rezultat w podejściu klasycznym został osiągnięty dopiero dla pierwszych 8 godzin propagacji, czyli na czterokrotnie dłuższym przedziale czasowym niż w przypadku modelu GCNConv. Biorąc pod

uwagę problem jak najwcześniejszej detekcji fałszywych informacji, model zbudowany w oparciu o grafowe sieci neuronowe wypada zdecydowanie korzystniej w tym zestawieniu.

Porównując wyniki dla obu modeli w przypadku ograniczenia ilościowego widać, że dla kaskad zawierających mniej elementów lepsze wyniki są uzyskiwane dla architektury GCNConv, przy czym już dla 20 postów wyższe wartości miary F1 zwracają modele wykorzystujące algorytm GraphSAGE. Ze względu na temat zagadnienia badawczego, w którym analizowane są możliwości jak najwcześniejszej detekcji fake newsów szczególnie interesujący jest wynik uzyskany dla pierwszych 5 postów przez model zbudowany z warstw GCNConv. Wartość miary F1 jest równa 0,894 i ponownie jest to wyższa wartość niż jakakolwiek inna uzyskana metodami klasycznymi dla ograniczenia ilościowego. Znów klasyfikatorem charakteryzującym się najlepszą jakością predykcji jest algorytm głosowania większościowego, tylko tym razem dla połączenia cech strukturalnych oraz nowych atrybutów (tabela 6.7), dla którego wartość miary F1 jest równa 0,88. Różnica między tymi wynikami ponownie jest niewielka, przy czym w podejściu klasycznym klasyfikator został wytrenowany na danych zawierających aż 100 pierwszych postów. W tym aspekcie można odnotować znaczącą przewagę modelu grafowego, ponieważ czas propagacji dla tak dużej wartości ograniczenia ilościowego może być bardzo długi, zwiększając szansę na to, że dana informacja zostanie zauważona przez popularnego użytkownika posiadającego wielu obserwatorów, co mogłoby skutkować wzrostem szybkości rozprzestrzeniania nieprawdziwej informacji w medium społecznościowym.

Warto również zwrócić uwagę na brak jednoznacznego wzrostu wartości miar F1 wraz z rozluźnianiem ograniczeń - to może sugerować albo nieoptymalne wartości hiperparametrów, a co za tym idzie utknięcie modelu w minimum lokalnym podczas treningu lub zbyt krótki czas uczenia - każda architektura była trenowana na maksymalnie 100 epokach.

6.2.2. Dane Gossipcop przygotowane wg. metody dołączania do oryginalnego tweeta

Tabela 6.19. Porównanie wyników uzyskanych dla architektur opartych o algorytmy GCNConv oraz GraphSAGE - zbiór Gossipcop zrekonstruowany metodą opisaną w podrozdziale 3.7.1.

Lp	Przycięcie	GCNConv - wartość miary F1	GraphSAGE - wartość miary F1
1	5 minut	0,905	0,85
2	30 minut	0,84	0,764
3	1 godzina	0,852	0,804
4	2 godziny	0,834	
5	4 godziny	0,85	
6	8 godzin		
7	12 godzin	0,851	

8	24 godziny		
9	5 postów	0,888	
10	10 postów	0,920	
11	20 postów	0,91	
12	30 postów	0,903	
13	40 postów		0,894
14	50 postów		0,912
15	100 postów		0,924

Porównując powyższe wyniki, które zostały uzyskane na zbiorze Gossipcop z tymi dla zestawu danych Politifact zawartymi w tabeli 6.19 można zauważyć, że dla kaskad przyciętych czasowo do pierwszych 5 minut propagacji komunikatów rezultaty w bieżącym podejściu są wyższe o około 13 punktów procentowych dla modelu zbudowanego w oparciu o algorytm GCNConv. Z kolei dla 30 pierwszych minut rozprzestrzeniania się informacji ta przewaga maleje do około 7 punktów procentowych. Biorąc pod uwagę różnicę w licznościach obu zbiorów, poprawa w drugim przypadku nie jest bardzo duża, szczególnie, że czas treningu modeli dla zestawu danych Gossipcop jest znacznie dłuższy niż dla Politifact.

Najwyższa zwrócona wartość miary F1 przez model niebazujący na sieciach neuronowych, do którego wejście stanowiły grafy ze zbioru Gossipcop ograniczone do pierwszych 5 minut propagacji komunikatów jest równa 0,86 i została uzyskana przez algorytm głosowania większościowego (wyniki znajdują się w tabeli 6.15) dla zestawu danych zawierającego jedynie nowe atrybuty. Ponownie, rezultat uzyskany przy pomocy modelu opartego o grafowe sieci neuronowe jest wyższy, tym razem o około 4 punkty procentowe.

Podobnie jak dla zbioru danych Politifact, w tym wypadku również można zauważyć, że wartości miary F1 wahają się mimo coraz mniej restrykcyjnych ograniczeń - zarówno ilościowych jak i czasowych.

7. Wnioski

Analizując rezultaty uzyskane dla podejścia wykorzystującego klasyczne modele uczenia maszynowego, które zostały zaprezentowane w podrozdziale 6.1, można dojść do kilku wniosków. Porównując wyniki dla obu metod rekonstrukcji kaskad komunikacyjnych w obrębie jednego zbioru danych można zauważyć, że są one równoważne, a dokładniej nie da się bezpośrednio wyodrębnić jednego sposobu przygotowania danych, który pozwalałby na uzyskanie znacznie lepszych wyników. Te obserwacje są takie same dla obu zbiorów danych.

Porównując wyniki uzyskane przez pojedyncze klasyfikatory oraz metody zespołowe widać, że metoda dynamicznego dobierania zespołu prowadziła najczęściej do uzyskania niższych, lub zbliżonych wartości miary F1 niż dla pozostałych klasyfikatorów. W dalszych badaniach należałoby rozważyć zmianę wewnętrznego algorytmu z KNORA-Eliminate na KNORA-Union i porównać uzyskane wartości. Możliwym wytłumaczeniem niskiej jakości predykcji metody dynamicznego tworzenia zespołu jest wybór zbyt restrykcyjnego algorytmu. Różnice pomiędzy KNORA-Eliminate oraz KNORA-Union zostały opisane w podrozdziale 2.3.3.

Warto zauważyć, że dla zbioru Politifact najwyższe wartości miary F1 były uzyskiwane metodą głosowania większościowego, a z kolei dla zbioru Gossipcop porównywalne wyniki były możliwe do osiągnięcia zarówno generalizacją stosową jak i pojedynczymi klasyfikatorami.

Biorąc pod uwagę wyniki dla poszczególnych kategorii atrybutów można zauważyć, że dla modeli wytrenowanych na danych wejściowych zawierających tylko jeden typ atrybutów, najwyższe wyniki są uzyskiwane dla nowych cech. Wartości miary F1 dla pozostałych kategorii atrybutów są w większości przypadków zbliżone między sobą, przy czym różnica między ich wynikami a tymi dla nowych cech potrafi wynosić nawet około 10 punktów procentowych, niezależnie od metody uczenia ani rozważanego zbioru. Jest to szczególnie ciekawa obserwacja biorąc pod uwagę, że nowy zestaw atrybutów zawiera cechy wynikające z analizy całej sieci społecznościowej (pagerank, lokalny stopień gronowania itd.) jak również atrybuty pochodzące z zawartości profili użytkowników. Analiza kombinacji dwóch z trzech cech pozwala na wyciągnięcie wniosku, że połączenie którejkolwiek cechy z nowymi atrybutami prowadzi do wzrostu jakości predykcji względem modeli wytrenowanych na zestawieniu cech strukturalnych oraz temporalnych.

Należy również zwrócić uwagę na fakt, że dla zbioru danych bazującego na portalu Gossipcop wyniki były wyższe, nawet do 10 punktów procentowych, niż dla zestawu danych zbudowanego w oparciu o stronę Politifact. Prawdopodobnie jest to bezpośrednim następstwem różnic w liczności obu zbiorów. Nawet po zbalansowaniu liczby próbek w klasach w zbiorze Gossipcop, Politifact zawiera jedynie około 7% przykładów tego zestawu danych. Jednak większa liczność zbioru przekłada się również na dłuższy czas treningu algorytmu. Jest to pewnego rodzaju kompromis, w którym należy rozważyć który aspekt jest

ważniejszy - jakość predykcji czy szybkość treningu. Większa liczność zbioru Gossipcop pozwoliła na uzyskanie ciekawych wyników dla wczesnej detekcji fałszywych wiadomości, szczególnie dla algorytmu głosowania większościowego dla zbioru zawierającego jedynie nowe cechy. Już na kaskadach zawierających 10 postów wartość miary F1 była równa 0,948, a dla przypadku generalizacji stosowej dla zestawu danych składającego się z przyciętych kaskad do 5 pierwszych komunikatów i tych samych cech wyniki były równe 0,938. Z kolei dla zbioru Politifact przy pomocy głosowania większościowego oraz identycznego zestawu atrybutów uzyskano wartość miary F1 równą 0,842 już dla pierwszych 5 minut propagacji komunikatów.

W przypadku treningów pojedynczych klasyfikatorów, wyliczana była ważność cech dla modelu, który pozwalał na uzyskanie najwyższej miary F1. W przypadku zbioru danych Politifact najwyższa jakość predykcji była uzyskiwana dla kombinacji cech temporalnych oraz nowych lub wszystkich dostępnych cech. W obu przypadkach trzy najbardziej wpływowe atrybuty pochodziły z zestawu nowych cech. Najbardziej wpływową cechą była informacja o tym, czy dane konto jest zweryfikowane. Dla zbioru Gossipcop najlepsze wyniki zostały uzyskane w jednym przypadku dla nowych cech oraz w drugim dla połączenia nowych atrybutów oraz cech strukturalnych. Najbardziej wpływową cechą był współczynnik agregujący aktywność użytkownika w okresie poprzedzającym dodanie przez niego posta, kolejną była reputacja konta wyliczana na podstawie liczby osób obserwujących oraz obserwowanych. To oznacza, że analiza sieci społecznościowej pod kątem nowych cech jest obiecującym kierunkiem badań, jednak wymaga dużych zasobów pamięciowych.

Analizując wyniki dla poszczególnych podejść można dojść do wniosku, że ograniczenie ilościowe prowadzi do wyższych wyników. Jednak w szczególności dla maksymalnej wartości przedziału - 100 postów - może to oznaczać bardzo długi czas propagacji informacji w sieci, co zwiększa szansę na dotarcie tych komunikatów do wpływowych użytkowników w sieci.

Przechodząc do interpretacji wyników uzyskanych dla grafowych sieci neuronowych, opisanych w podrozdziale 6.2 należy zwrócić uwagę na kwestię rozmiaru wektora atrybutów. Te modele były trenowane przy wykorzystaniu mniejszej liczby cech - 15 dla pojedynczego węzła - w porównaniu z modelami klasycznymi, opisanymi w podrozdziale 6.1, które posiadały 7 atrybutów strukturalnych, 8 temporalnych oraz 20 nowych. Dodatkowo, w przypadku sieci grafowych większość tych cech była wyznaczana na podstawie informacji zawartych w profilach kont lub były to współczynniki obliczane bazując na sieci społecznościowej, gdzie w obu przypadkach nie były dostępne pełne dane. Fragmentaryczność zestawów informacji wynika z bardzo czasochłonnego procesu ich pobierania z API Twittera oraz ograniczeń pamięciowych na wykorzystanym urządzeniu. Z kolei zarówno atrybuty strukturalne jak i temporalne były wyznaczane na podstawie informacji takich jak struktura kaskady komunikacyjnej czy też znaczniki czasowe kolejnych postów, które to dane były w całości pobrane i dostępne. Mniejsza całkowita liczba atrybutów wynika z

konieczności wyznaczenia tych wartości dla każdego pojedynczego węzła znajdującego się w grafie wejściowym do grafowej sieci neuronowej, gdzie dla modeli klasycznych wyliczana była wartość średnia z wszystkich węzłów.

Ograniczeniem, które mogło również mieć negatywny wpływ na uzyskane wyniki była stosunkowo niewielka ilość danych, szczególnie dla portalu Politifact. Modele neuronowe zbudowane z tak skomplikowanych warstw zazwyczaj posiadają dużą liczbę parametrów, których wartości są wyznaczane podczas treningu, co z jednej strony pozwala na dopasowanie się do skomplikowanych funkcji opisujących dane, a z drugiej nakłada wymaganie na rozmiar zbioru treningowego, który powinien składać się z o wiele większej liczby próbek niż na przykład dla prostych algorytmów niebazujących na sieciach neuronowych. Mimo to, grafowe sieci neuronowe, a w szczególności modele zbudowane w oparciu o algorytm GCNConv, pozwoliły na osiągnięcie najwyższych wartości miar F1 dla zbioru Politifact. Biorąc pod uwagę ograniczenie czasowe już dla danych zawierających jedynie komunikaty powstałe w ciągu pierwszych dwóch godzin propagacji wartość miary F1 była równa 0,895. Z kolei dla grafów przyciętych do pierwszych 5 postów wynik tej miary wynosił 0,894. Szczególnie druga wartość jest ciekawym rezultatem patrząc z perspektywy wczesnej detekcji fałszywych wiadomości, ze względu na stosunkowo wysoki wynik uzyskany dla bardzo restrykcyjnego ograniczenia.

Można zauważyć, że wraz z rozluźnianiem ograniczeń zarówno czasowych jak i ilościowych wartość miary F1 fluktuuje, zamiast zachowywać spodziewaną tendencję wzrostową. Może to być związane na przykład z nieoptymalnymi wartościami hiperparametrów, powodującymi zatrzymanie uczenia w minimum lokalnym. Ze względu na długi czas trenowania pojedynczego modelu przestrzeń hiperparametrów mogła nie zostać zbadana w wystarczającym stopniu. Innym wytłumaczeniem może być za krótki czas uczenia, wynikający ze zbyt małej odgórnie zadanej liczby epok, w trakcie których model jest trenowany. Może to prowadzić do wymuszonego zatrzymania procesu uczenia w momencie, w którym wartość funkcji straty wciąż zdąża do swojej minimalnej wartości, a parametry modelu są aktualizowane. Dla obu zbiorów maksymalna liczba epok była równa 100, a jej wybór został podyktowany kompromisem między jakością predykcji a czasem pojedynczego treningu.

Porównując dwie przetestowane architektury: GCNConv oraz GraphSAGE widać, że w przypadku ograniczenia ilościowego pierwsza z nich pozwala na uzyskanie wyższych wartości miar F1 dla danych zawierających bardziej restrykcyjnie przycięte kaskady komunikacyjne, szczególnie dla pierwszych 5 oraz 10 postów. Z kolei model oparty o algorytm GraphSAGE oraz warstwę DiffPool charakteryzuje się lepszą jakością predykcji na danych zawierających 20 lub więcej komunikatów. Może to świadczyć o tym, że ta druga architektura posiada więcej parametrów, których wartości są aktualizowane podczas treningu, czyli jest bardziej skomplikowana. Patrząc z kolei z perspektywy ograniczenia czasowego

można zauważyć, że generalnie model GCNConv pozwala na osiągnięcie wyższych miar F1.

Podobnie jak w przypadku algorytmów niebazujących na grafowych sieciach neuronowych, analiza uzyskanych rezultatów prowadzi do konkluzji, że modele wytrenowane na próbkach ograniczonych ilościowo zwracają generalnie wyższe wartości miary F1. Zbiory danych zawierające grafy wejściowe przycięte na podstawie czasu upływającego od opublikowania pierwszego posta są zdecydowanie mniej liczne, co jest następstwem faktu, że w większości przypadków pierwsze udostępnienia nie pojawiają się w tak krótkim odstępie czasu jak na przykład 5 lub 30 minut. Poddając analizie nieograniczone w żaden sposób próbki można było zauważyć, że moment inicjalizacji propagacji w medium społecznościowym następował zazwyczaj później, poza zakresem do którego przycinane były kaskady.

8. Zakończenie

W obecnych czasach, w których ponad połowa populacji świata ma dostęp do mediów społecznościowych i aktywnie z nich korzysta, coraz większym problemem staje się zjawisko celowej dezinformacji. Jej wpływ na kształtowanie opinii publicznej można było zaobserwować na przykład podczas wyborów prezydenckich w USA w 2016 roku. Ze względu na rosnącą skłonność do ufania informacjom pozyskanym z serwisów społecznościowych, szczególnie w grupie młodych dorosłych, pojawia się potrzeba weryfikacji komunikatów w czasie rzeczywistym, zanim trafią do szerokiego grona odbiorców. Jest to o tyle istotne, że wraz z rozprzestrzenianiem fałszywych informacji, rośnie szansa na to, że więcej ludzi im zaufa, a badania dowodzą, iż trudno jest skorygować postrzeganie danego tematu, nawet jeśli jest ono mylne.

W celu zaadresowania tego problemu w niniejszej pracy została zbadana możliwość detekcji fałszywych komunikatów we wczesnej fazie ich rozprzestrzeniania na Twitterze. Przeanalizowane modele zostały podzielone na dwie grupy: klasyczne algorytmy niebazujące na sieciach neuronowych, oraz grafowe sieci neuronowe. Pod uwagę były brane cechy strukturalne i temporalne obliczane dla całych kaskad komunikacyjnych oraz zaproponowany został zestaw nowych atrybutów wyodrębnionych z profili użytkowników oraz sieci połączeń pomiędzy nimi. W toku badań udowodniono, że nowe cechy pozytywnie wpływają na jakość predykcji klasyfikatorów. Mimo iż nie były brane pod uwagę informacje wynikające z analizy semantycznej oraz składniowej tekstu, w podejściu z klasycznymi algorytmami uczenia maszynowego uzyskano wysokie wartości miary F1: dla zbioru Politifact oraz danych przyciętych do pierwszych 5 minut propagacji komunikatów równą 0,842, a dla zbioru Gossipcop i próbek zawierających jedynie pierwszych 5 postów wynoszącą 0,938. Z kolei dla grafowych sieci neuronowych na zbiorze Politifact osiągnięto wartość miary F1 wynoszącą 0,894 dla danych ograniczonych do pierwszych 5 postów.

Dla algorytmów niebazujących na sieciach neuronowych wyznaczono ważność atrybutów. Cechą, która miała największy wpływ na uzyskane wyniki dla zbioru Politifact była informacja o tym, czy konto danego użytkownika udostępniającego tweet lub retweet jest zweryfikowane a dla zbioru Gossipcop współczynnik agregujący aktywność użytkownika w okresie poprzedzającym dodanie przez niego posta. Ta druga cecha jest nowym współczynnikiem, zaproponowanym w ramach tej pracy.

Ze względu na specyfikę dostępu do zbioru danych, a w szczególności skomplikowanego procesu jego pobrania oraz konieczności rekonstrukcji sieci społecznościowej, w ten sposób przygotowane modele uczenia maszynowego mogą zostać wykorzystane raczej w zastosowaniu operatora danego serwisu internetowego niż w indywidualnym. Szczególnie biorąc pod uwagę bardzo wczesną detekcję fałszywych informacji (po pierwszych 5 minutach, albo na podstawie pierwszych 5 postów) pobranie wymaganego zestawu danych z API Twittera mogłoby być zbyt czasochłonne dla indywidualnych jednostek. Z drugiej strony status medium społecznościowego, w którym udostępniane są głównie wiarygodne

informacje na pewno przełożyłby się na wyższe zainteresowanie użytkowników, a co za tym idzie większą popularność danego portalu.

Niniejsza praca nie wyczerpuje jednak analizowanego tematu pozostawiając miejsce na dalsze badania. W kolejnych krokach należałoby przede wszystkim skupić się na zbiorze danych, a konkretnie na pobraniu z API Twittera pełnego zestawu informacji o profilach użytkowników oraz dotyczących relacji obserwowania pomiędzy nimi. Następnie, dysponując większymi zasobami pamięciowymi, należałoby odtworzyć całą sieć społecznościową. Biorąc pod uwagę algorytmy uczenia maszynowego dla metod zespołowych zaproponowanych w podejściu klasycznym należałoby przetestować różne kombinacje podstawowych klasyfikatorów, co mogłoby pozwolić na wyłonienie takiej, która pozwalałaby na zwiększenie jakości predykcji. W ujęciu grafowych sieci neuronowych należałoby przetestować więcej różnych architektur - zarówno kombinacji warstw opisanych w niniejszej pracy jak również zbadać możliwe zastosowania nowych algorytmów. Dodatkowo, należałoby poświęcić więcej zasobów na znalezienie optymalnych zestawów hiperparametrów dla modeli służących do wczesnej detekcji fake newsów. Ze względu na czasochłonność tego procesu oraz mnogość rozważanych przypadków, ten temat nie został wyczerpany.

Bibliografia

[1] Wong, Belle, "Top Social Media Statistics And Trends Of 2023", Dostęp zdalny (08.2023): https://www.forbes.com/advisor/business/social-media-statistics/, 2023.

[2] A. Bovet i H. Makse, "Influence of fake news in Twitter during the 2016 US presidential election", Nature Communications, t. 10, 2019. DOI: 10.1038/s41467-018-07761-2.

[3] K. Shu, D. Mahudeswaran, S. Wang, D. Lee i H. Liu, "FakeNewsNet: A Data Repository with News Content, Social Context and Dynamic Information for Studying Fake News on Social Media", CoRR, t. abs/1809.01286, 2018. adr.: http://arxiv.org/abs/1809.01286.

[4] Liedke, Jacob and Gottfried, Jeffrey, "U.S. adults under 30 now trust information from social media almost as much as from national news outlets", Dostęp zdalny (08.2023): https://www.pewresearch.org/short-reads/2022/10/27/u-s-adults-under-30-now-trust-information-from-social-media-almost-as-much-as-from-national-news-outlets/, 2022.

[5] "Social Media and News Fact Sheet", Dostęp zdalny (08.2023): https://www.pewresearch.org/journalism/fact-sheet/social-media-and-news-fact-sheet/, 2022.

[6] V. Balakrishnan, W. Z. Ng, M. C. Soo, G. J. Han i C. J. Lee, "Infodemic and fake news – A comprehensive overview of its global magnitude during the COVID-19 pandemic in 2021: A scoping review", International Journal of Disaster Risk Reduction, t. 78, s. 103 144, 2022. DOI: 10.1016/j.ijdrr.2022.103144. PMID: 35791376.

[7] L. Ha, L. A. Perez i R. Ray, "Mapping Recent Development in Scholarship on Fake News and Misinformation, 2008 to 2017: Disciplinary Contribution, Topics, and Impact:" American Behavioral Scientist, s. 000 276 421 986 940, 2019. adr.: https://api.semanticscholar.org/CorpusID:203111220.

[8] K. Shu, H. R. Bernard i H. Liu, "Studying Fake News via Network Analysis: Detection and Mitigation", CoRR, t. abs/1804.10233, 2018. arXiv: 1804.10233.

[9] F. Monti, F. Frasca, D. Eynard, D. Mannion i M. M. Bronstein, "Fake News Detection on Social Media using Geometric Deep Learning", CoRR, t. abs/1902.06673, 2019. arXiv: 1902.06673.

[10] T. Li, Y. Sun i e. a. Shang-ling Hsu, "Fake News Detection with Heterogeneous Transformer", 2022. arXiv: 2205.03100 [cs.SI].

[11] S. Vosoughi, D. Roy i S. Aral, "The spread of true and false news online", Science, t. 359, s. 1146–1151, 2018. DOI: 10.1126/science.aap9559.

[12] K. Shu, A. Sliva, S. Wang, J. Tang i H. Liu, "Fake News Detection on Social Media: A Data Mining Perspective", CoRR, t. abs/1708.01967, 2017. arXiv: 1708.01967.

[13] K. Shu, D. Mahudeswaran, S. Wang i H. Liu, "Hierarchical Propagation Networks for Fake News Detection: Investigation and Exploitation", CoRR, t. abs/1903.09196, 2019. arXiv: 1903.09196.

[14] S. Aphiwongsophon i P. Chongstitvatana, "Detecting Fake News with Machine Learning Method", w ECTI-CON 2018, 2018, s. 528–531. DOI: 10.1109/ECTICon.2018.8620051.

[15] M. I. Uddin, I. Ahmad, M. Yousaf, S. Yousaf i M. O. Ahmad, "Fake News Detection Using Machine Learning Ensemble Methods", Complexity, t. 2020, s. 8 885 861, 2020, ISSN: 1076-2787. DOI: 10.1155/2020/8885861.

[16] J. Thorne, M. Chen, G. Myrianthous, J. Pu, X. Wang i A. Vlachos, "Fake news stance detection using stacked ensemble of classifiers", w NLPmJ@EMNLP, 2017. adr.: https://api.semanticscholar.org/CorpusID:38349306.

[17] Y. Han, S. Karunasekera i C. Leckie, "Graph Neural Networks with Continual Learning for Fake News Detection from Social Media", CoRR, t. abs/2007.03316, 2020. arXiv: 2007.03316. adr.: https://arxiv.org/abs/2007.03316.

[18] P. Saikia, K. Gundale, A. Jain, D. Jadeja, H. Patel i M. Roy, "Modelling Social Context for Fake News Detection: A Graph Neural Network Based Approach", 2022. arXiv: 2207.13500 [cs.SI].

[19] M. Dhawan i S. Sharma, "GAME-ON: Graph Attention Network based Multimodal Fusion for Fake News Detection", 2022. arXiv: 2202.12478 [cs.MM].

[20] J. Zhou, G. Cui, S. Hu i in., "Graph Neural Networks: A Review of Methods and Applications", 2021. arXiv: 1812.08434 [cs.LG].

[21] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang i P. S. Yu, "A Comprehensive Survey on Graph Neural Networks", IEEE Trans Neural Netw Learn Syst, t. 32, s. 4–24, 2019.

[22] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner i G. Monfardini, "The Graph Neural Network Model", IEEE Transactions on Neural Networks, t. 20, nr. 1, s. 61–80, 2009. DOI: 10.1109/TNN.2008.2005605.

[23] T. N. Kipf i M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks", CoRR, t. abs/1609.02907, 2016. arXiv: 1609.02907.

[24] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò i Y. Bengio, "Graph Attention Networks", 2018. arXiv: 1710.10903 [stat.ML].

[25] W. L. Hamilton, R. Ying i J. Leskovec, "Inductive Representation Learning on Large Graphs", CoRR, t. abs/1706.02216, 2017. arXiv: 1706.02216.

[26] D. Grattarola, D. Zambon, F. M. Bianchi i C. Alippi, "Understanding Pooling in Graph Neural Networks", CoRR, t. abs/2110.05292, 2021. arXiv: 2110.05292.

[27] C. Liu, Y. Zhan, J. Wu i in., "Graph Pooling for Graph Neural Networks: Progress, Challenges, and Opportunities", 2023. arXiv: 2204.07321 [cs.LG].

[28] R. Ying, J. You, C. Morris, X. Ren, W. L. Hamilton i J. Leskovec, "Hierarchical Graph Representation Learning with Differentiable Pooling", CoRR, t. abs/1806.08804, 2018. arXiv: 1806.08804.

[29] A. Daigavane, B. Ravindran i G. Aggarwal, "Understanding Convolutions on Graphs", Distill, 2021, https://distill.pub/2021/understanding-gnns. DOI: 10.23915/distill.00032.

[30] H. Gao i S. Ji, "Graph U-Nets", CoRR, t. abs/1905.05178, 2019. arXiv: 1905.05178.

[31] J. Lee, I. Lee i J. Kang, "Self-Attention Graph Pooling", CoRR, t. abs/1904.08082, 2019. arXiv: 1904.08082.

[32] P. Cichosz, Wykłady do przedmiotu "Metody Odkrywania Wiedzy", 2020.

[33] S. Shalev-Shwartz i S. Ben-David, "Understanding Machine Learning: From Theory to Algorithms", 2014.

[34] G. H. John i P. Langley, "Estimating Continuous Distributions in Bayesian Classifiers", CoRR, t. abs/1302.4964, 2013. arXiv: 1302.4964.

[35] P. Cunningham i S. J. Delany, "k-Nearest Neighbour Classifiers: 2nd Edition (with Python examples)", CoRR, t. abs/2004.04523, 2020. arXiv: 2004.04523.

[36] Y. Freund i R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting", Journal of Computer and System Sciences, t. 55, nr. 1, s. 119–139, 1997, ISSN: 0022-0000. DOI: https://doi.org/10.1006/jcss.1997.1504.

[37] L. Kuncheva, "Combining Pattern Classifiers: Methods and Algorithms: Second Edition", w 2014, t. 47, s. 113–116. DOI: 10.1002/0471660264.

[38] D. Wolpert, "Stacked Generalization", Neural Networks, t. 5, s. 241–259, 1992. DOI: 10.1016/S0893-6080(05)80023-1.

[39] Dostęp zdalny (08.2023): https://rasbt.github.io/mlxtend/user_guide/classifier/StackingClassifier/.

[40] S. Džeroski i B. Ženko, "Is Combining Classifiers with Stacking Better than Selecting the Best One?", Machine Learning, t. 54, s. 255–273, 2004. DOI: 10.1023/B:MACH.0000015881.36452.6e.

[41] A. Ko i A. Jr, "From dynamic classifier selection to dynamic ensemble selection", Pattern Recognition, t. 41, s. 1718–1731, 2008. DOI: 10.1016/j.patcog.2007.10.015.

[42] Dostęp zdalny (08.2023): https://research.aimultiple.com/machine-learning-accuracy/.

[43] A. Hirunyawanakul, N. Kerdprasop i K. Kerdprasop, "A Novel Heuristic Method for Misclassification Cost Tuning in Imbalanced Data", Int J Mach Learn Comput, t. 8, s. 565–570, 2018. DOI: 10.18178/ijmlc.2018.8.6.746.

[44] D. M. W. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation", ArXiv, t. abs/2010.16061, 2011.

[45] Y. Sasaki, "The truth of the F-measure", Teach Tutor Mater, 2007.

[46] M. Sokolova, N. Japkowicz i S. Szpakowicz, "Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation", t. Vol. 4304, 2006, s. 1015–1021, ISBN: 978-3-540-49787-5. DOI: 10.1007/11941439_114.

[47] S. Dixon, "Number of Twitter users worldwide from 2019 to 2024", Dostęp zdalny (08.2023): https://www.statista.com/statistics/303681/twitter-users-worldwide/, 2022.

[48] Z. Kechen, H. Shen i H. Zhang, "User spread influence measurement in microblog", Multimedia Tools and Applications, t. 76, 2017. DOI: 10.1007/s11042-016-3818-z.

[49] K. Shu i D. Mahudeswaran, FakeNewsNet, Dostęp zdalny (14.03.2019): https://github.com/KaiDMML/FakeNewsNet, 2019.

[50] M. Paraschiv, N. Salamanos, C. Iordanou, N. Laoutaris i M. Sirivianos, "A Unified Graph-Based Approach to Disinformation Detection using Contextual and Semantic Relations", CoRR, t. abs/2109.11781, 2021. arXiv: 2109.11781.

[51] H. Liang, "Broadcast Versus Viral Spreading: The Structure of Diffusion Cascades and Selective Sharing on Social Media", Journal of Communication, t. 68, nr. 3, s. 525–546, 2018, ISSN: 0021-9916. DOI: 10.1093/joc/jqy006.

[52] Y. Dou, K. Shu, C. Xia, P. S. Yu i L. Sun, "User Preference-aware Fake News Detection", CoRR, t. abs/2104.12259, 2021. arXiv: 2104.12259.

[53] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu i M. Sun, "Graph Neural Networks: A Review of Methods and Applications", CoRR, t. abs/1812.08434, 2018. arXiv: 1812.08434.

[54] M. E. J. Newman, "Mixing patterns in networks", Phys. Rev. E, t. 67, s. 026 126, 2 2003. DOI: 10.1103/PhysRevE.67.026126.

[55] Y. Liu i Y.-F. Wu, "Early Detection of Fake News on Social Media Through Propagation Path Classification with Recurrent and Convolutional Networks", Proc. Innov. Appl. Artif, t. 32, nr. 1, 2018. DOI: 10.1609/aaai.v32i1.11268.

[56] L. E. Boehm, "The Validity Effect: A Search for Mediating Variables", PSPB, t. 20, nr. 3, s. 285–293, 1994. DOI: 10.1177/0146167294203006.

[57] J. De keersmaecker i A. Roets, "'Fake news': Incorrect, but hard to correct. The role of cognitive ability on the impact of false information on social impressions", Intelligence, t. 65, s. 107–110, 2017, ISSN: 0160-2896. DOI: https://doi.org/10.1016/j.intell.2017.10.005.

[58] C. Yuan, Q. Ma, W. Zhou, J. Han i S. Hu, "Early Detection of Fake News by Utilizing the Credibility of News, Publishers, and Users Based on Weakly Supervised Learning", CoRR, t. abs/2012.04233, 2020. arXiv: 2012.04233.

[59] H. Jia, Wang, H. i X. Zhang, "Early detection of rumors based on source tweet-word graph attention networks", PloS one, t. 17, nr. 7, 2022.

[60] A. Silva, Y. Han, L. Luo i S. Karunasekera, "Propagation2Vec: Embedding partial propagation networks for explainable fake news early detection", Inf. Process. Manag., t. 58, nr. 5, s. 102 618, 2021. DOI: 10.1016/j.ipm.2021.102618.

[61] A. Zubiaga, M. Liakata i R. Procter, "Learning Reporting Dynamics during Breaking News for Rumour Detection in Social Media", CoRR, t. abs/1610.07363, 2016. arXiv: 1610.07363.

[62] S. Feng, H. Wan, N. Wang, J. Li i M. Luo, "TwiBot-20: A Comprehensive Twitter Bot Detection Benchmark", CoRR, t. abs/2106.13088, 2021. arXiv: 2106.13088.

[63] T. Ahmad, M. S. Faisal, A. Rizwan, R. Alkanhel, P. W. Khan i A. Muthanna, "Efficient Fake News Detection Mechanism Using Enhanced Deep Learning Model", Applied Sciences, t. 12, nr. 3, 2022, ISSN: 2076-3417. DOI: 10.3390/app12031743.

[64] Z. Zhao, J. Zhao, Y. Sano i in., "Fake news propagate differently from real news even at early stages of spreading", 2019. arXiv: 1803.03443 [physics.soc-ph].

[65] M. Meyers, G. Weiss i G. Spanakis, "Fake News Detection on Twitter Using Propagation Structures", w Disinformation in Open Online Media, Cham: Springer International Publishing, 2020, s. 138–158, ISBN: 978-3-030-61841-4.

[66] S. Goel, A. Anderson, J. Hofman i D. Watts, "The Structural Virality of Online Diffusion", Management Science, t. 62, s. 150 722 112 809 007, 2015. DOI: 10.1287/mnsc.2015.2158.

[67] Dostęp zdalny (08.2023): https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.link_analysis.pagerank_alg.pagerank.html.

[68] L. Page, S. Brin, R. Motwani i T. Winograd, "The PageRank Citation Ranking : Bringing Order to the Web", w The Web Conference, 1999.

[69] Dostęp zdalny (08.2023): https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.cluster.clustering.html.

[70] G. Fagiolo, "Clustering in complex directed networks", Physical Review E, t. 76, nr. 2, 2007.

[71] S. Bhat i M. Abulaish, "Community-Based Features for Identifying Spammers in Online Social Networks", 2013. DOI: 10.1145/2492517.2492567.

[72] C. E. Shannon, "A mathematical theory of communication", Bell Syst. Tech. J., t. 27, s. 623–656, 1948. adr.: https://api.semanticscholar.org/CorpusID:55379485.

[73] D. Felmlee, C. McMillan i R. Whitaker, "Dyads, triads, and tetrads: a multivariate simulation approach to uncovering network motifs in social graphs", Appl. Netw. Sci., t. 6, 2021. DOI: 10.1007/s41109-021-00403-5.

[74] Dostęp zdalny (08.2023): https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/object-model/user.

[75] E. Rossi, H. Kenlay, M. I. Gorinova, B. P. Chamberlain, X. Dong i M. M. Bronstein, "On the Unreasonable Effectiveness of Feature propagation in Learning on Graphs with Missing Node Features", CoRR, t. abs/2111.12128, 2021. arXiv: 2111.12128.

[76] Dostęp zdalny (08.2023): https://optuna.readthedocs.io/en/stable/.

[77] Y. Wu, L. Liu, J. Bae i in., "Demystifying Learning Rate Polices for High Accuracy Training of Deep Neural Networks", CoRR, t. abs/1908.06477, 2019. arXiv: 1908.06477.

[78] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever i R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting", J. Mach. Learn. Res.,

t. 15, s. 1929–1958, 2014. adr.: https://api.semanticscholar.org/CorpusID:6844431.

[79] S. Ioffe i C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", CoRR, t. abs/1502.03167, 2015. arXiv: 1502.03167.

[80] K. Zhou, Q. Song, X. Huang, D. Zha, N. Zou i X. Hu, "Multi-Channel Graph Convolutional Networks", CoRR, t. abs/1912.08306, 2019. arXiv: 1912.08306.

[81] Dostęp zdalny (08.2023): https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html.

[82] S. Sharma, S. Sharma i A. Athaiya, "ACTIVATION FUNCTIONS IN NEURAL NETWORKS", International Journal of Engineering Applied Sciences and Technology, t. 04, s. 310–316, 2020. DOI: 10.33564/IJEAST.2020.v04i12.054.

[83] I. Goodfellow, Y. Bengio i A. Courville, Deep Learning. MIT Press, 2016, http://www.deeplearningbook.org.

[84] B. Xu, N. Wang, T. Chen i M. Li, "Empirical Evaluation of Rectified Activations in Convolutional Network", CoRR, t. abs/1505.00853, 2015. arXiv: 1505.00853.

[85] D. Buterez, J. P. Janet, S. J. Kiddle, D. Oglic i P. Liò, "Graph Neural Networks with Adaptive Readouts", 2022. arXiv: 2211.04952 [cs.LG].

Wykaz symboli i skrótów

EiTI – Wydział Elektroniki i Technik Informacyjnych
PW – Politechnika Warszawska

Spis rysunków

Spis tabel