FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of master's thesis

Title:	Fraudulent behavior detection using graph-based anomaly detection approaches
Student:	Bc. Ondřej Šofr
Supervisor:	Ing. Jaroslav Kuchař, Ph.D.
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2023/2024

Instructions

Instructions:
1) Review approaches and methods used for graph-based anomaly detection.
2) Get familiar with the field of fraudulent behavior detection. Examine several subdomains of this field (for example fraudulent product reviews, financial frauds or disinformation) and analyze their key differences. Analyze the applicability of methods studied in the previous part.
3) Propose a new method or a modification of existing methods that can improve detection in this task.
4) Thoroughly examine its performance (e.g accuracy or types of anomalies that can be detected) using suitable datasets (for example YelpCHI for reviews, Elliptic Dataset for transactions or MuMiN for disinformation detection) and compare it with existing methods.
5) Discuss the results and outline aspects that might be improved in further work.

Electronically approved by Ing. Magda Friedjungová, Ph.D. on 16 September 2022 in Prague.

Master's thesis

FRAUDULENT BEHAVIOR DETECTION USING GRAPH-BASED ANOMALY DETECTION APPROACHES

Bc. Ondřej Šofr

Faculty of Information Technology
Department of Applied Mathematics
Supervisor: Ing. Jaroslav Kuchař, Ph.D.
January 5, 2023

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis: Šofr Ondřej. *Fraudulent behavior detection using graph-based anomaly detection approaches.* Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2023.

Contents

List of Figures

List of Tables

I would like to thank my supervisor Ing. Jaroslav Kuchař, Ph.D. for his guidance and for giving me the opportunity to work on this interesting topic.

I would also like to thank all members of my family for their endless support throughout my studies.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60 (1) of the Act.

In Prague on January 5, 2023 ....

Abstract

This thesis deals with the field of anomaly detection. Traditional and modern approaches are described and analyzed, especially the methods of graph-based anomaly detection and their ability to utilize the relational information in the data. The main focus is the processing and modeling of user actions in order to detect their fraudulent behavior in several domains. Prediction accuracy and efficiency of solutions are evaluated with the emphasis on the usability in practice. This thesis contains experimental results of presented methods tested on real-world data containing posts extracted from social networks. The task is to detect misinformation claims using the structure of related discussion threads. It is shown that there is a connection between anomaly detection and misinformation detection and that it is possible to achieve accurate results using these approaches.

Keywords anomaly detection, graph-based data representation

Abstrakt

Tato práce se zabývá problematikou detekce anomálií. Jsou zde představeny přístupy využívané v současnosti, zejména metody pracující nad grafovou reprezentací dat, u nichž je analyzována jejich schopnost využívat informaci o vztazích mezi entitami. Hlavním zaměřením práce je popis uživatelského chování a detekce podvodného jednání osob v několika doménách. Kvalita předpovědí a efektivita daných řešení je porovnána, s důrazem na použití v praktických úlohách. Práce obsahuje experimentální výsledky představených metod, testovaných na datasetu příspěvků získaných ze sociálních sítí. Úlohou je zde detekovat dezinformace a lživá tvrzení s využitím struktury navazujících diskuzních vláken. Je ukázáno, že existuje vztah mezi obecnou úlohou detekce anomálií a detekcí lživých tvrzení a také, že je možné dosáhnout kvalitních výsledků s pomocí zkoumaných přístupů.

Klíčová slova detekce anomálií, grafová reprezentace dat

List of Abbreviations

AUC	Area under curve
AP	Average precision
GAE	Graph autoencoder
LOF	Local outlier factor
ML	Machine learning
ROC	Receiver operating characteristic

Chapter 1
Introduction

Representing data as a graph has many advantages over other approaches in domains where there are numerous interactions between various entities. In many problems linked to activity in networks, such entities can be users and objects of their interest such as companies, products, websites or news articles. Unfortunately, it is common to encounter malicious behavior of users in many domains. One example might be public websites containing reviews of products or services – fraudsters often create multiple fake user accounts and write purposely biased reviews as their goal is either to improve the rating of their own product or to harm the reputation of others'. Another example is using social networks to deliberately spread disinformation news. Several types of this illicit behavior can be spotted using the techniques of graph-based anomaly detection, which is the main topic of this study.

Chapter 2
Related work

This chapter summarizes the state of the art of undergoing research. While the anomaly detection domain has had solid foundations dating back to the previous century, only in the last decade has it attracted the attention of many researchers working in many seemingly unrelated fields.

2.1 Anomaly detection in graph-based data

Anomaly detection is a field of study with a long history, which focuses on detecting observations that deviate significantly from the set of other collected observations. In its basic form it is considered to be an unsupervised machine learning task, as the data is usually analyzed without any additional labels. Originally studied as a means of data preprocessing, it has since become crucial for many other domains.

There are several approaches, amongst which the most commonly used is multi-dimensional anomaly detection. The observations are described as multi-dimensional vectors and thus can be depicted as points in space (typically high-dimensional). This task is often referred to as *outlier detection* and it has been thoroughly studied. An example of a commonly used method is *Local outlier factor* (LOF) (proposed in [1]), which is based on the assumption that the outliers can be detected by the low density of other observations around them. As such, LOF utilizes distance in multi-dimensional space between points of data. Although the multi-dimensional approach might bring good results, it has its limitations in processing data that are relational by its nature and cannot be easily described as vectors of features.

2.1.1 Graph-based data representation

To accurately detect anomalies in various domains, in which the data has relational structure, graph-based data representation exhibits several advantages. By representing entities as nodes and links between them as edges, the inter-dependencies are correctly captured. As the nodes and edges in graphs can have various attributes, no information about them is lost either.

What is even more important is the change in the meaning of *neighborhood*. In multi-dimensional representation, the neighborhood of a data point is usually the set of several other points that are closest by some distance metric – yet this often means that it measures the similarity but not the *connection* between entities. On the other hand, graph-based representation supports constructing the neighborhood as a set of other nodes reachable by a path of a certain length, which has a more intuitive explanation and suits several tasks better. Fraud detection, where the structure of connections is very important, happens to be one of them (this claim is supported in detail in [2]).

Another advantage is that it can be more difficult for malicious entities to mask their behavior and avoid suspicion when the detection algorithms use graph data. It is usually easy for fraudsters to observe typical attributes of a normal user, for example numerical or text values used to customize user profile or log-in times, and mimic them. However, without knowing the complete structure of a graph of interactions, it is difficult for them to produce a camouflaged behavior to avoid detection.

2.1.2 Methods

Graph anomalies can be categorized based on the type in relation to the graph structure. It is common to make distinction between vector anomalies, edge anomalies and subgraph anomalies as each category can be detected by different variants of methods.

For a basic outline of methods in respect to the type of algorithm used, I will use classification as proposed in [2], which is a very detailed survey of this field. The authors divide methods by its approach into:

- *Feature-based methods* – for each node, a variety of attributes are computed from its neighborhood (for example degree of the node or number of cliques in the neighborhood) and used for traditional multi-dimensional anomaly detection

- *Proximity-based methods* – these methods exploit graph structure to measure closeness, as attributes are generated by some metrics that take the structure of the graph into account, such as PageRank score or pairwise SimRank score

- *Community methods* – each node is assigned to a community and anomalies are detected in subsequent phase as nodes that could be easily considered to belong to another community or nodes that significantly differ from other nodes in their community

- *Decomposition methods* – using matrix decomposition (typically utilizing modified adjacency matrix) obtain a set of characteristics such as eigenvalues and eigenvectors; what is often computed is the reconstruction error resulting from decomposition into smaller-dimensional matrices

- *Relational Learning methods* – using a subset of data previously categorized into one of two classes (anomalous or normal), select the proper class for the rest of data; therefore the task is transferred into the domain of supervised or semi-supervised learning

Shortly after this survey had been written, deep learning methods have started to attract a massive attention of researchers, resulting in numerous other approaches to appear. The focus of the research community has shifted significantly to these new methods related to graph neural networks (as documented in [3]). Nowadays, main study directions include (with many more emerging):

- *Embedding methods* – for each node, a vector or sequence of vectors is created and used as an input to a neural network; embeddings can also be calculated using deep learning methods such as graph convolutional networks (GCNs)

- *Autoencoder methods* – graph is compressed to a lower-dimensional description and reconstructed by an autoencoder neural network; anomalies can be then detected by analyzing a reconstruction error in selected parts of the graph

2.1.3 Challenges

There are two main difficulties involving the evaluation of methods that make this field of study challenging. First of all, anomaly detection is an unsupervised task in which finding previously unseen patterns is highly valued. This means that method evaluation can either be done by manual expert assessment of found anomalies or by transforming the task into a standard supervised classification task where labels denoting anomaly and non-anomaly classes are provided. As the first one is nearly impossible for large data, the second one is the only suitable possibility. However, this is hindered by the fact that the anomaly class cannot be easily described (usually there are different kinds of anomalies mixed in the data) and that such class tends to have much less observations because the anomalies are quite rare. Different methods also focus on different types of anomalies and should not be compared solely by their accuracy, as is discussed in [4]. Secondly, there are confidentiality issues preventing companies from offering high-quality datasets aimed at fraud detection to the public. This means that research is conducted on datasets that are either synthetic, small or incomplete. A more thorough discussion of this topic can be found in [3], in addition to methods of preprocessing such data.

2.1.4 Utilization of temporal dimension

The temporal information can be found in multiple forms in data. Usually it describes a moment in time when a certain activity happened, its duration or the order of such activities. There are two basic ways of representing it.

The first one is to store the timestamps of activities as an attribute of edges corresponding to these activities. This is a straightforward method, which does not really change the structure of the so-called *static graph*. However, it makes it an attributed graph, which means that not every method can be used to process it and even methods suited to attributed graphs might not be capable of fully utilizing the temporal information in this way. A more basic approach might be to create a set of additional edges without attributes, in situations where there is some kind of temporal relationship between nodes. This would however alter the structure of the graph.

The second one is to construct a sequence of graphs. In this sequence, every graph has the same set of nodes, yet only the activities observed during the corresponding time interval (each graph in sequence has a different one, typically with the same length as others) result in creation of an edge. Such a graph is called a *dynamic graph* and generally cannot be processed by the same set of methods as a static graph. The definition of the problem is also altered as not only node, edge and subgraph anomalies can be the targeted types, but also *events* and *changes* can be detected. Both are terms for a time step in which the dynamic graph shows abnormal properties, but they are slightly different. Events are time steps during which the graph looks significantly different from the previous and following time steps – meaning that the graph returns to a normal state after an event. Changes are time steps during which the graph looks different from the previous steps, but it retains such structure for the following time steps.

2.2 Fraud detection domains

2.2.1 Fake reviews

There are many websites and applications that let users write reviews of products, organizations or services. A typical fraudulent behavior is to write fake reviews. Most of the time, a user or a group of users try to deliberately improve the rating of a product in order to increase its popularity; or on the contrary to decrease the popularity of other products by making them look worse. This is often a coordinated effort of a high number of fake user profiles because several reviews are needed in order to make a significant impact. Malicious users also often mask their behavior by writing numerous ordinary reviews (typically positive reviews of well-known popular products).

Approaches to this problem can be divided (according to [2]) into three categories – behavioral analysis, language analysis and relational analysis. The third is the most relevant for this project, for which two traditional methods are mentioned.

Algorithm proposed in [5] (2011) assigns trustiness score for all of reviewers, reviews and product. Scores are subsequently recalculated as each of these three types of score is dependent upon others – for example the trustiness of a product is calculated by a function of trustiness of all reviews of it. This method to some extent resembles HITS, which is an algorithm for determining the authoritativeness of websites. Convergence of the solution is not guaranteed and the algorithm cannot utilize additional information about entities aside from the structure of the graph.

Algorithm *FraudEagle*, proposed in [6] in 2013, also uses iterative computation. It operates on bipartite graphs containing users and products. For both of these types of nodes, trustiness is calculated using *Loopy Belief Propagation* in order to spot anomalous users that tend to rate bad products high and vice versa. In the second phase, an induced subgraph is created over the set of these users and products reviewed by at least one of them. Then, a clustering algorithm is run on this subgraph to detect organized groups of fraudsters and validate the results of the first phase.

In recent years, the attention has shifted to methods of deep learning to solving previously omitted challenges. An example of this is [7], an article from 2020 focused on overcoming the active defense of fraudsters against detection by traditional methods. This so-called camouflage is based on mimicking the activity of regular users and creating positive feedback for attacking profiles, either using other fraudster profiles or accounts stolen from honest users. Against this camouflage, the researchers propose a new method *CARE-GNN (CAmouflage REsistant GNN)*, which overcomes this challenge and outperforms all older methods. The algorithm starts by building a node embedding using several phases, during which a neural network chooses from all neighbors a set of nodes with similar behavior and ignores suspicious connections. Node classification is then applied over this embedding.

Modern approaches to graph autoencoders are used in [8], an article published this year. Aside from an encoding part, two decoding submodules are trained, one specializing in detecting honest users and the other one focusing on malevolent users. This method has been named CGNN (*competitive graph neural networks*).

2.2.2 Online auctions and marketplaces

A problem with a significant economical importance is how to detect fraudulent behavior at online marketplaces such as *eBay* or *Amazon*. At these websites, users can buy products from others in two ways – either they accept an offer and make a trade immediately or bid in an auction, which ends after a certain time and the winner makes the trade. In the first case, the most common malicious behavior is not delivering a paid order by the seller, who gets the money and ceases to respond. In the second case, there can be a high amount of fake bids made by a group of fraudsters to drive up the price (who might even fake the trade, if necessary, and repeat the auction).

Earlier approaches to this problem utilizes more basic methods such as *2-Level Fraud Spotting* or *NetProbe*, proposed in 2006 and 2007 in [9] and [10] respectively. These are based on heuristic calculation of base value for the trustiness of each user and subsequent iterative propagation of these values by *Loopy Belief Propagation*. An advanced variant utilizing the same basic principles is proposed in [11] (written in 2014).

I have not found any article written in the last three years, which illustrates that this domain has slowly lost its appeal and not much research is conducted in it. That also means that no deep learning methods were proposed specifically for it.

2.2.3 Telecommunication frauds and spam

This domain deals with communication, especially messages, e-mails and telephone calls. The goal is to detect users that abuse such services by sending unwanted messages or for other fraudulent behavior like telephone scams. All subcategories have their own features, for example spam e-mails are hardly ever targeted to specific recipients. However, telephone scams are much more often conducted by an organized group of fraudsters and their victim is usually chosen purposely. The possible detection methods are also different as spam e-mails are often detectable using linguistic analysis only. Nevertheless, representing the network of messages or calls as a graph enables the usage of additional methods.

An article [12], written in 2020, focuses on a type of telephone scams where fraudsters persuade victims to send money to them. A new way of creating node embedding sequences is proposed, named *FraudWalk*. It is based on the *DeepWalk* algorithm, but the temporal information is taken into account during the random walk sampling. Only when the sampled edge denotes a call made in the time interval of $[t - \delta_t, t + \delta_t]$ (where t is the time of last call in sequence and δ_t is a parameter) is the current walk appended with the edge, otherwise it ends. This builds upon the assumption that fraudulent calls to a victim are made shortly after each other, even if coming from different telephone numbers. Such embedding is then combined with other information and the final classification is made by modified LSTM recurrent neural networks. An attention mechanism is also implemented to correctly determine which part of the sequence is the most important.

2.2.4 Financial frauds

Every kind of financial transactions can be easily represented as a graph, which is why it is commonly used to detect various types of financial frauds, from which the most notable are:

- Money laundering is a technique used by criminals to legalize money gained through criminal behavior. Typically, such money is transferred through a chain of accounts to make the process difficult to track and uncover.
- Traders (for example at stock market) might make a large number of purposeless transactions amongst between them to manipulate the market or reduce its stability
- Fraudsters can gain access to financial accounts of victims and make transactions to withdraw money from it. This type, called *cross-channel fraud*, is a major concern for banks.
- Insurance frauds such as that a group of people makes a high number of insurance contracts and then fakes incidents.
- Companies create a chain of contracts with faked costs to avoid paying customs and taxes.

Article [13] (written in 2012) proposes two methods for detecting market-manipulating groups of traders. Both are focused on detecting communities of users that exhibit signs of this behavior such as high number of internal transactions or high difference in incoming and outcoming transaction amounts.

A method named *coDetect* is described in [14] (2018), which can utilize both the structure of the transaction network and the attributes of entities in it. Decomposition algorithms are used in both cases and the resulting reconstruction error can be easily used to detect participating entities. The method works well for several fraud schemes such as money laundering, cross-channel frauds and insurance frauds.

Survey [15] (written in 2019) compares the traditional machine learning algorithms such as *Random Forest* and MLP to graph convolutional neural networks with respect to accuracy of detection of financial frauds over a graph of Bitcoin transactions.

2.2.5 Misinformation and disinformation

The goal of disinformation detection is to flag messages or posts on social media that contain unsupported claims or are deliberately created to confuse people. Aside from linguistic analysis of messages, it is possible to process the structure of the networks and utilize the information about publishers, senders and recipients that interact with such posts.

Detection of disinformation is an actively studied domain and the number of research articles has been growing significantly during the last five years, which means that most of the methods in use utilize deep learning. Older and more traditional approaches (especially statistical and distance methods) are reviewed and compared in a survey [16] written in 2019. This work also focuses on processing dynamically growing graphs.

An article [17] (2017) discusses the utilization of several types of information to detect potentially disinformation posts. A method named *CSI (Capture-Score-Integrate)* is proposed. The first part of this model creates node embedding of an article based on its text using *word2vec* algorithm and temporal information regarding its views by users by modified LSTM recurrent neural network. The second part of this model performs a SVD decomposition and dimensionality reduction of weighted adjacency matrix, which for each pair of users tracks the number of articles both reacted to. Final vector of attributes for each user is then processed by feed-forward NN and the results are used for classification.

Several other articles dealing with deep learning methods and temporal dimension are for example [18] (2021) which proposes the method *Temporally Evolving Graph Neural Network for Fake News Detection (TGNF)* and [19] (written in 2022), proposing *Factual News Graph (FANG)* framework.

Chapter 3

Availability of datasets

This chapter summarizes the state of the art involving publicly available datasets. As the typical usage of anomaly detection algorithm for fraud detection deals with credential information about users or other information that might be seen as sensitive from companies' perspective, obtaining reasonable data for research purposes can be an obstacle. Nevertheless, several promising datasets have been constructed. Their properties are studied in this chapter.

3.1 Overall situation

There are two main difficulties involving the evaluation of methods that make this field of study challenging. First of all, anomaly detection is an unsupervised task in which finding previously unseen patterns is highly valued. This means that method evaluation can either be done by manual expert assessment of found anomalies or by transforming the task into a standard supervised classification task where labels denoting anomaly and non-anomaly classes are provided. As the first one is nearly impossible for large data, the second one is the only suitable possibility. However, this is hindered by the fact that the anomaly class cannot be easily described (usually there are different kinds of anomalies mixed in the data) and that such class tends to have much less observations because the anomalies are quite rare. Different methods also focus on different types of anomalies and should not be compared solely by their accuracy (as is discussed in [4]).

Secondly, there are confidentiality issues preventing companies from offering high-quality datasets aimed at fraud detection to the public. This means that research is conducted on datasets that are either synthetic, small or incomplete (concluded in [3]). More often than not, modifying an existing dataset created for other purposes or fields is necessary. For example recommender systems datasets can be used for fake review detection whenever those contain a sufficient structure of users and reviews. Of course no labels as to who is a fraudster could be expected, so this information must be either calculated from other attributes or computed by some other established method, which creates labels that are considered *near-ground truth* at best. Despite all of these challenges, there are several public datasets that are suitable for further research.

3.2 Datasets for misinformation detection

As there are many possible tasks solvable by the graph anomaly detection approaches, it is out of the scope of this thesis to focus on all of them. Instead, the field of misinformation detection will be the most thoroughly studied.

3.2.1 Overview

As the result of the significant increase in popularity as a research topic in the last ten years, several research groups have proposed datasets (or methodology for their construction) aiming to provide a way of comparing methods and help future research in this field. Most of the datasets contain pieces of information in the form of articles or short statements (those are commonly referred to as *claims* in the context of fact-checking) that are considered either truthful or not, with many of them being truthful only partially. The intended task is to classify which of them belong to which category. What separates individual datasets is the method of collecting data and the structure and amount of additional information connected with these statements. For example, authors and publishing websites are typically taken into account regarding online articles. When relational information (such as who sent a message containing dubious statements to whom) are incorporated, the data can be modeled as a graph.

3.2.2 Standard datasets

In this section, several datasets are introduced. While all of them have different structure and properties, there are a number of key concepts shared by them. All of them contain claims made by real speakers and at least some veracity labels are provided by their authors. There is always an entity (typically statement, social network post or user) that is to be classified either as truthful or malicious.

3.2.2.1 Emergent

Historically, misinformation detection evolved from the task of automated fact checking. Probably the first public dataset focused on this topic was constructed by Vlachos and Riedel [20] in 2014. The intent of their work was originally to help automate the process of assessment of truthfulness of political claims, which is a tedious task normally performed by journalists who have to carefully evaluate individual parts of the claim and compare them to reliable sources of information. The output is usually a verdict supported by analysis of the reasoning behind it. This dataset is very small by today's standards as it contains only 106 statements, each accompanied by a verdict denoting truthfulness, using a five-point scale. The researchers gathered those statements and verdicts from two fact-checking websites (Channel 4 and PolitiFact) covering public life and politics in the US and UK. Because of the structure of the dataset, consisting only of textual representation of statements and categorical verdicts, it is almost impossible to utilize methods other than natural language processing. The authors discuss that additional sources of information not incorporated in the dataset are necessary for any classifier to have reasonable results.

The same research group published another similar dataset named *Emergent* [21] in 2016. This time, they gathered 300 claims with a total of 2,595 news articles, each of them having a supporting, refuting or just reporting stance on one claim. Again, only textual representations of claims and article headlines were used and NLP methods were recommended for this task by the authors.

3.2.2.2 Liar

Significantly larger and structurally richer is the *Liar* [22] dataset, proposed by Wang in 2017. It consists of approximately 12,800 statements, all of them collected and carefully assessed by the PolitiFact website. Statements are accompanied by several metadata labels, such as subject of the claim, context, venue and basic information about the speaker. The truthfulness rating of statements falls into one of six categories and their distribution can be considered well-balanced. Thematically, the dataset is focused on the political situation in the US with several prominent politicians being the authors of more than 100 statements. Yet, a significant portion of statements come from non-politicians or at least people affiliated with neither major political party. There are several relations that could even be utilized in graph representation.

3.2.2.3 PHEME

PHEME (often called PHEME5 to distinguish between its versions) is a dataset presented in an article by Zubiaga et al. [23] (written in 2016), which is focused on rumor detection. That is a task closely related to misinformation detection as the goal is to label statements that are unverified and might not be true. The typical use case presented in the aforementioned article is to check messages and reports made shortly after a major real-world event and to warn users which of those information might be only a rumor. There is a subtle difference between misinformation and rumor as a rumor might turn out to be true even though it is not possible to verify it at the moment of publishing. To properly test their approaches, the researchers collected a huge amount of tweets that were written immediately after five distinctive events that were shocking to the public and also surrounded by many unverified speculations – for example Charlie Hebdo office shooting and Germanwings plane crash in March 2015. Using several criteria, they sampled a subset of the tweets and provided them to a group of journalists who were closely monitoring those events. Inspecting the tweets, they provided manual annotations as to whether the content of tweets could have been verified or not before publishing. In total there are 5,802 tweets in the dataset, of which 34 % were deemed rumors. Aside from their textual representation, the dataset contains metainformation and a collection of replies for each of these tweets (although the classifiers proposed in the article do not utilize the replies).

In 2018, the same group of authors presented an extended version of the dataset named *PHEME9* in [24]. Tweets related to four topics were added and a new level of annotations was constructed – for each tweet that was deemed to be a rumor, journalists added a label whether the rumor later proved to be true, false or whether it remained unverified. The distribution of each label is different for each event, which is said by the authors to provide an interesting challenge for the construction of detection models.

3.2.2.4 MultiFC

MultiFC (the name connects the terms multi-domain and fact-checking) was proposed in 2019 in [25]. It is a large dataset of claims and statements constructed in a rather standard manner. The researchers gathered a list of 28 renowned fact-checking websites and collected a total amount of 34,918 claims by crawling them. The structure of these samples differ for each website, yet most of the claims are accompanied by rich metadata containing the speaker, tags, the fact-checking person, several temporal information and a veracity label with textual analysis. In addition to that, each claim was used verbatim as a query to the Google Search API and 10 most relevant results were saved (in the form of full web pages with various metadata). These web pages serve the role of context information or evidence for each of the claims. The researchers also performed named entity recognition and entity linking to find the topic of claims. The two most frequently mentioned entities are the United States and former US president Barack Obama, which shows that most of the claims are related to US politics.

Although this dataset is sufficiently large and robust, the way of presenting the veracity labels could be considered an obstacle. Each crawled website has its own methodology for rating the claims and both the wording and even the number of possible verdicts (ranging from 2 to 27) is different amongst them. Yet, the labels are stored in textual representation and the authors of the dataset made little effort to map those labels to a common veracity scale.

3.2.2.5 Fake News Net

Fake News Net is rather an ongoing data preprocessing project than a single dataset. The first versions of the resulting dataset were presented in 2017 in an article [26]. Because of the fact-checking sources and year of publishing, those two versions are usually referred to as *FNNBuzzFeed17* and *FNNPolitifact17* in other articles. What made them innovative is the high variety of information in them. The main entities are news articles, of which several hundred were collected and labeled for veracity. As the goal of authors was to evaluate a rather complex factorization algorithm with multiple submodules, three types of information were gathered - news content, social engagements (activity of Twitter users regarding posts about the articles) and publisher partisan (left-right political scale).

The most recent versions were published in 2020, as described in [27]. This time, fact-checking websites GossipCop.com and Politifact.com were used as a source of news articles labeled as fake; trusted news websites were used as source of real news to help make the dataset more balanced (as fact-checking websites are focused on debunking misinformations, these tend to be a majority of content). With this step completed, a large collection of Twitter posts containing the titles or URLs of these news articles was created. Response posts were gathered as well and for every user engaged in any way with those posts, their user profile information, their network of following and other metainformation was collected. The dataset contains a lot of spatiotemporal information as timestamps of user activities are stored and location of users are often visible from their profiles. The size of the data is outstanding compared to those mentioned earlier. There are over 345,000 users posting over 1.4 million tweets in the GossipCop part of the dataset.

3.2.2.6 Co-AID

Another worldwide event that has brought significant attention to misinformation detection was the beginning of Covid-19 pandemics at the beginning of 2020. There were many pieces of misinformation spreading on the internet and social media, speculating about the origin of the virus, promoting medically questionable treatments and falsely informing about the development of vaccines. In the span of the following two years, tens of datasets of various quality were created on this topic.

One of the earliest is *Co-AID* [28], published as a preprint in May 2020. Its goal is to cover both news articles and social media posts. For the former, articles from several reliable and checked medical sources (for example World Health Organization) were gathered and labeled as true, while articles highlighted and debunked by several fact-checking organizations were labeled as misinformation. The title, abstract, keywords and the full text are stored for each article. User engagement is also collected – tweets discussing said articles, as well as replies to them and metadata about their authors. In the second part of the dataset, several hundreds claims that were originally posted by users of social networks are collected and labeled for veracity by fact-checking websites. What is interesting is that in this part, misinformation makes up over 90 % of all claims, while in the articles part the situation is almost the opposite. Both parts of the dataset might thus be difficult to process using the same methods.

3.2.2.7 MM-COVID

Another dataset related to the ongoing pandemics is *MM-COVID* [29], preprinted in November 2020. Its construction is very similar to Co-AID – several reliable news sources are used to collect articles labeled as truthful and articles gathered at fact-checking websites are added with their veracity labels. These two steps are done to ensure that there are more truthful articles than fake ones, as collecting only from fact-checking websites would result in an imbalanced dataset with the majority of news being misinformation. The main contribution of this dataset is that it is *multilingual*, meaning that articles and all other textual information come in a variety of languages. This is achieved by collecting the articles from the International Fact-Checking

Network of Poynter Institute [30], unifying 96 organizations at the moment of data collection. From a large number of languages, six were chosen based on their frequency in articles - English, Spanish, Portugal, Hindi, French and Italian.

Using a multilingual dataset can be very beneficial for the research of misinformation spread. The authors of MM-COVID mention examples of false claims that appeared in one language and several days later their translations spread around the globe – which should be preventable by that time by correctly linking those claims together. Moreover, the ultimate goal of every misinformation classifier is to detect false news even in infrequently spoken languages, dialects and jargons. Traditional language models are usually tuned for formal English and thus cannot perform properly in this situation. Other methods (graph representation being amongst them) might bring more promising results.

This dataset also contains user social engagement information in a standard form (used also by other similar datasets). A query is made from each article title and relevant Twitter posts are gathered, alongside a subset of replies and user information about authors of the tweets. A rich temporal information collected regarding tweets about a certain claim can prove useful especially for misinformation spread modeling, as mentioned in the previous paragraph.

3.2.2.8 UPFD

User Preference-aware Fake News (UPFN) dataset, proposed in 2021 in [31], is based on the FakeNewsNet family of datasets. The methods presented and evaluated in the article focus on *endogenous* preferences of news consumers – utilizing a concept that users are far more likely to spread news that confirms their existing view (which is named *Confirmation Bias* in psychology [32]). As so, the researchers needed to gather a different kind of information than what usually is the backbone of similar datasets. Taking both versions of the 2020 FakeNewsNet dataset as a base, they collected additional information regarding previous activity of all users that are present in those datasets (meaning that the users engaged with any of the news articles, either posting related tweets or re-tweeting them). For each user, two hundred most recent tweets were fetched (if possible) and an aggregate representation of them was constructed using *word2vec* and *BERT* linguistic models.

The authors have also built a news propagation graph in an attempt to reconstruct how misinformation spread in the social network. Although it is not possible to determine what made each user retweet a certain post, the researchers followed reasonable heuristics. If multiple accounts that are followed by a certain user posted a news article before this user retweeted it, it is assumed that the user had seen it from the tweet posted the latest (earlier posts are not considered as much important for the user by the Twitter app filtering mechanism). If none of the followed accounts had spread the news, a non-followed account with the most followers (probably the largest authority source) is deemed as the source. This approach effectively creates a tree structure for every article, with the original source being the root.

Thanks to its rich structure, the UPFN dataset has been included in one of the most used deep learning graph libraries – PyTorch Geometric [33] as a benchmark dataset.

3.2.2.9 MuMiN dataset

MuMiN: A Large-Scale Multilingual Multimodal Fact-Checked Misinformation Social Network Dataset [34] is the most recent dataset, proposed by Nielsen and McConville at the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, which was held in July 2022. The goal of the authors is to provide the most complex dataset up to date, utilizing several advanced concepts. First of all, the dataset contains claims in 41 languages from all over the world, including smaller languages such as Czech, Kazakh, Oriya and Macedonian, making it truly multilingual. The most important language is English, having 45 % of claims in the MuMiN-medium version. The dataset is *multimodal*, meaning that there are claims related to various topics (politics, finance, healthcare, celebrities, sport, etc.), yet the analysis shows that most of the claims (approximately 50 %) are related to the Covid-19 pandemics.

Compared to other datasets, the construction process is also remarkably complex. First, a list of 115 fact-checking organizations from around the globe was gathered using Google FactCheck Tools API [35]. Each claim checked by those organizations was then collected and parsed into a universal format. To preprocess the textual veracity labels (fact-check verdicts) into only three categories (true claim, false claim, other), all labels were translated into English and a deep learning classifier was built using a manually classified part of the set. For each claim, all relevant Twitter posts made in the preceding three days before the claim was fact-checked were collected. Then, rich metadata and relational information regarding the tweets, users, mentioned articles and images were collected using a pipeline of several crawling processes.

There are three versions of the dataset (MuMiN-small, MuMiN-medium and MuMiN-large), which differ in the thresholds of the relevancy metrics – stricter values result in a smaller dataset with less noise. The authors claim that MuMiN is the largest misinformation dataset. Its properties are truly outstanding, with 21.6 million tweets from 2 million users in the MuMiN-large version. It contains 12,914 claims as the main entity upon which the misinformation classification is intended. This is more than in all other datasets except MultiFC and FakeNewsNet-GossipCop 2020 version, which have 34,918 and 22,140 respectively.

What is inconvenient for the usage of anomaly detection methods is the significant disbalance of classes. Unlike other datasets such as FNN and MM-Covid, the authors of MuMiN have collected the claims from fact-checking websites only. Because those websites focus on debunking misinformation news, there are much more articles labeled as such than those that are deemed truthful. In MuMiN-medium, the ratio of misinformation is 94.2 % of all claims. The authors discuss this fact and argue that it is better to not include claims from other sources, "as these will likely arise from different distribution then the rest of the dataset" [34]. Therefore, a special attention should be given to this fact when performing method evaluation using this dataset.

Table 3.1 Overview of available misinformation datasets

Dataset	Year	Main entities	Size	Meta info.	Relational info.
Vlachos 2014	2014	claims	106 claims	none	none
Emergent	2016	claims, articles	300 claims	none	none
Liar	2017	claims	12,800 cl.	some	some
PHEME5	2016	tweets	5,802 tw.	some	some
PHEME9	2018	tweets	6,425 tw.	some	some
MultiFC	2019	claims	34,918 cl.	rich	some
FNNPolitifact17	2017	articles, users	240 ar.	rich	rich
FNNGossipCop20	2020	articles, tweets	22,140 ar.	rich	very rich
Co-AID	2020	articles, tweets	3,769 ar.	some	rich
MM-COVID	2020	articles, tweets	11,173 ar.	rich	rich
UPFD-GossipCop	2021	articles, users	5,464 ar.	rich	very rich
MuMiN-medium	2022	claims, tweets	5,565 cl.	very rich	rich

Size is measured by the number of entities that are intended for veracity classification task. Meta information column denotes how much additional information about entities and their relationships are collected. Relational information column denotes how much information can be utilized for creating graph representation. Last two columns uses a scale of 4 values – none, some, rich and very rich. These values are assigned as my subjective opinion based on dataset analysis.

3.2.3 Special datasets

Aside from the mentioned datasets focused on verifying claims, statements or messages, there are several datasets with other goals. They can still be considered as part of misinformation detection, but their structure is different from the traditional ones.

3.2.3.1 FEVER

What is common for all the datasets mentioned so far in this chapter, is that they gather claims or statements that were made by real speakers (or at least social network users). A completely different approach was selected by the authors of *Fact Extraction and Verification* dataset (*FEVER*), presented in 2018 [36]. They took a large number of claim sentences from Wikipedia articles and manually altered them in a variety of ways, changing the meaning of a part of them. A group of annotators were to decide then, whether the claims are truthful or not, again using Wikipedia. The annotators were also to point out the sentences supporting or refuting the claim. As such, the authors have gathered an astonishing number of 185,445 labeled claims.

Although this dataset in many ways resembles all the other misinformation datasets, its structure makes it more suitable for the task of verification against textual sources and related linguistic fields. There is also no relational data usable for graph modeling.

3.2.3.2 MediaEval

A rather different task was presented as a part of 2015 Benchmarking Initiative for Multimedia Evaluation (*MediaEval*) [37] . The goal of the *Verifying Multimedia Use* task is to uncover misinformation in the form of images. Given a Twitter post about a major world event accompanied by an image, a classifying model should decide whether the image correctly depicts the event, or is manipulated. The manipulation can be either a digital alteration of the image content or improper usage, such as using an older image depicting another event, or even using a picture originating from a movie or art installation, which has no connection to the event described in the tweet.

The dataset contains 361 images used in 12,040 tweets dealing with either world events (such as Boston Marathon bombing in 2013) or non-existencial events and hoaxes. Apart from the textual information, rich meta information was gathered for each tweet and its author – for example numbers of replies and mentions of the tweet and number of followers of the user. The images were also forensically analyzed and many features describing their technical properties, such as coefficients of the JPEG compression, were collected. This information could be used for the advanced analysis, whether the picture was digitally altered.

3.3 Datasets for other domains

There are many other applications of graph anomaly detection and their full review is out of the scope of this thesis. To illustrate the availability of datasets in other domains, several of them are shortly mentioned in this section.

3.3.1 Fraudulent reviews

The *Amazon Review Data* project [38], maintained by UC San Diego researchers, offers data from the Amazon online marketplace, focusing on user reviews. There are 233 milion reviews in total, each containing timestamp, rating on a scale from 1 to 5 stars, textual rating, evaluation of *helpfulness* of the review as rated by other users and meta-informations about the reviewed product. The dataset is structured by the product category, meaning that only a certain types of products can be used (for example *electronics* contains 21 million review). It is easy to create a graph representation of desired structure from this dataset and because of the timestamps a dynamic graph of arbitrary time granularity can be constructed as well. Another dataset from the same domain that is often used in research articles is *Yelp Open Dataset* [39], published by the Yelp platform used for reviewing hotels and restaurants. It consists of over 7 million reviews of 150 000 establishments. Several researchers have also used scraping of Yelp websites to obtain labels as there is an option to detect suspicious reviews provided. Such approach is used in the *YelpCHI* dataset [40].

3.3.2 Financial transactions

As for the domain of financial transactions, [14] utilizes a database of documents obtained during the Panama Papers and subsequent leakages, which is then preprocessed into a graph representation. The data is publicly available at the website of The International Consortium of Investigative Journalists [41], but it does not contain labels for categorizing entities. Another interesting dataset is also the *Eliptic Data set*, proposed in [15]. This is a database of Bitcoin blockchain transactions containing 200 000 anonymized accounts. Some of them are labeled honest (20 % from the total amount) and some are labeled illicit (2 %); the rest are unlabeled. Illicit entities are those that accepted money gained through crime, frauds, malware and ransomware attacks and also that participated in Ponzi schemes or funded terrorist organizations. Aside from the basic features describing the transactions, there are 166 attributes related to every account. Unfortunately, those values are not described in detail due to confidentiality.

Chapter 4

Graph anomaly detection experiments

The following pages try to answer the crucial question of this thesis: "Is it possible to utilize the methods of anomaly detection to spot misinformation on social networks? Or is this task too difficult for this approach alone?" In this chapter I thoroughly describe the data preprocessing steps and experimental setups using several model architectures. What is similar for all of them is that a single graph is created for each news article, forming a large collection. The anomaly score is then computed for each of these graphs.

4.1 Used data

The previous chapter provides an overview of the most frequently used datasets suited for misinformation detection tasks. Although there is a variety of well-prepared datasets, none of them can be considered truly dominant in the research in this field. There are multiple reasons for this. Firstly, the field is rapidly developing and there has not emerged a widely accepted benchmark authority (as is, for example the Open Graph Benchmark [42] in the domain of general graph machine learning) yet. Secondly, new algorithms that are presented usually excel under different conditions than others and their authors naturally tend to alter existing datasets (or even create new ones) to showcase their strengths.

To select a proper dataset for each task is a difficult challenge. For the purposes of this thesis, I have especially focused on two key properties of the datasets:

- *Is there a rich source of relational information in the dataset? Alternatively, is there enough information with the potential to be modeled as a part of graph representation.*

It would clearly make no sense to use a predominantly textual dataset intended to be used solely for the comparison of natural language processing models. There has to be enough relationship between entities (either as a part of the dataset, or derivable by standard methods) to justify the usage of graph models.

- *What was the process to determine the ratio of target classes (misinformation and truthful)? Does it follow reasonable assumptions and was the data gathered accordingly to reflect a real-life balance of such news?*

Class disbalance is a well studied topic in machine learning. Traditional approaches are thoroughly reviewed in [43] and [44]. Generally, unbalanced datasets are considered to be more difficult to process by many algorithms and special care has to be given to the choice of experimental setups and resulting metrics, because a large portion of them can bring misleading or uninterpretable results. On the other hand, anomaly detection is based on the assumption that anomalies are not frequent, so this aspect has to be addressed. It can also be expected that the ratio of classes will heavily impact the performance of some models. Because of this, it is necessary for the dataset to correctly depict the distribution of misinformation amongst all data.

4.1.1 Dataset selection

Taking these requirements into account, I have chosen the *Fake News Net* project as the primary source of data. The variants officially published in 2020 (*FNN Gossipcop* and *FNN Politifact*) are significantly rich in relational data. There are several entities, the most important of which are news articles, tweets and users, that are connected by several types of relationships and many more can be modeled from the provided metadata.

The authors were also aware of the fact that their main data collecting process – gathering news articles and claims with veracity verdicts from fact-checking websites – brings significant bias towards misinformations, as those websites are aimed to detect, flag and explain the fake claims and generally do not contain obviously true claims. To mitigate this bias, authors have also collected a large set of news articles from reliable sources. The final percentage of fake claims depends on the version of the dataset, as well on my preprocessing steps, but generally it is between 15 and 25 percent. This seems like a reasonable amount. Unfortunately, I have not been able to find a reliable analysis of how much news on the internet is to be considered misinformation. Several sociological studies try to find out how much people encounter such news, but there are two main problems – the results might be heavily influenced by the lack of ability to determine what is misinformation (certain news could be viewed differently by a average person and, for example, radical far-right or far-left supporter) and this number might not properly describe how much news is misleading, given that some might spread only in certain communities, which is common for news spread on social networks. Therefore I adopt the dataset authors' reasoning and consider this amount of misinformation as truthfully depicting reality.

The Fake News Net datasets are also suitable for this work in other aspects. They are large enough in terms of the number of samples and misinformation topics they cover. They have been used in several research articles and serve as a base for other extended datasets. The data are publicly available (although with several limitations and challenges, presented further in this chapter) and the authors provide a framework for downloading and extending the datasets (although I had to severely modify it, as described in later sections).

4.1.1.1 On selecting labeled datasets

A careful reader might wonder: "Why is it so important that the reviewed datasets have labels? It has been stated many times that anomaly detection is an unsupervised learning method and thus requires no labels and in fact it is suited for tasks where no labels are provided." This is a correct observation. All presented methods operate without any label information. Throughout the process of designing experiments I have been extremely careful not to utilize this information in any way, as this would go completely against the intended direction of the research. It is important to note that even a basic supervised learning model (for example a simple decision tree) is expected to bring good results and may outperform even the best unsupervised model. Nevertheless, under the harsh (yet very common) constraint that no labels are provided, such a model could not be trained and deployed.

However, the goal of this thesis is to present and evaluate several approaches and algorithms. Without the ground truth values that determine which news belongs to each class, the choice of evaluation metrics and comparison approaches is severely limited. In terms of unsupervised learning, several compression models can be compared by their average reconstruction error without the use of labels, but there is no guarantee that low value results in a good model for the task of misinformation detection. Similarly, clustering algorithms can be compared by various metrics measuring cohesion of resulting clusters, yet there might be no clear connection to the performance on binary classification tasks.

While using labeled datasets is relatively common in literature presenting anomaly detection methods, there are also alternative methods used for determining their quality. In several articles, the researchers have taken the most likely candidates for being anomalous and inspected them manually. This has enabled them to test their approaches on data that is difficult to label and potentially larger than any labeled dataset, yet it does not really show the performance of the models in general (for example how much anomalies are left undetected). Another method is to use a well established anomaly detection model and use its outputs as so-called *near-ground-truth*, which means assuming that these are the desired labels. This approach is generally not recommended as the model developed under these settings is evaluated for its ability to predict the base model outputs, rather than real labels.

All these points considered, selecting a fully labeled dataset is the best option for this thesis in order to properly evaluate the performance of presented methods. Using unlabeled or semi-labeled (where only a small portion of samples are labeled) data is a potential further research task, out of scope of this thesis.

4.1.2 Using Twitter data

What do most of the datasets presented in the previous chapter have in common, is that they analyze posts (called tweets) and other user activity performed at the social network Twitter. This is not surprising, given that Twitter has always been quite friendly to the data science community and has granted researchers several variants of elevated access to its data through the Twitter API. Nevertheless, it is not always easy to work with it, as I have experienced many times during the writing of this thesis. In this section, I will summarize its functionality and outline what impact its limitations have on the used datasets.

4.1.2.1 Twitter policy on data sharing

Twitter has established a robust set of policies for the handling of its data. That is understandable, considering that users' data protection is of high importance nowadays and extensive sharing of tweets and user profiles could be seen unfavorably by its customers (even though all this information can be gathered via web interference with enough effort).

The current Twitter policy ([45]) on content redistribution heavily impacts the format in which the related datasets are shared by researchers. It is not possible to any data in textual form as it is stated that *"If you provide Twitter Content to third parties, including downloadable datasets or via an API, you may only distribute Tweet IDs, Direct Message IDs, and/or User IDs (except as described below)."* in the aforementioned document. Because of this, the standard process for the construction of datasets is as follows:

- The authors construct a dataset that uses tweets or user profiles, or at least a subset of features of those.
- They share a set of Tweet IDs or User IDs that are needed for the dataset (a set of such IDs is often called *dehydrated tweets/users*). Tools for preprocessing this data to desired state are usually provided.

The typical usage of such dataset requires several steps for the person wanting to use the dataset:

- It is necessary to obtain the required level of access permissions from Twitter. This is discussed in the following section.

- The tweets or user profiles have to be downloaded using the Twitter API. IDs of these entities are passed to the API calls as arguments. This process is commonly called *rehydrating tweets/users.*
- The required preprocessing is to be made according to the dataset authors. This could be some kind of feature selection and transformation or more sophisticated steps like reconstructing the sequences of news spread through the network.

This Twitter policy ensures that only people authorized by Twitter are able to view and process the complete information used in each dataset. Also, Twitter can modify the availability of such information any time. Unfortunately, this has several consequences for the accessibility and quality of datasets.

4.1.3 Collecting data

After selecting the dataset, the resulting data had to be collected in accordance with the process outlined in the previous section. Despite the fact that Fake News Net is a widely used dataset, it was not as straightforward as I had hoped for. Unfortunately, the provided code for downloading and preprocessing Twitter data (published as a repository at [46]) had been written for the nowadays outdated version of Twitter API (version 1.1, described at [47]) and no update to it has been made since then.

The second challenge to overcome was the fact that due to the limited download rate, the process of downloading the whole dataset would take several years (by my approximations). The authors of the dataset were well aware of this and have recommended using a larger collection of Twitter accounts with academic access (in fact, they have provided a robust code for synchronizing the download process when using more of them). While this solution might work for larger research groups, where every member fulfills the requirements for having such an account, it is not a solution I could have used. I had to carefully select a subset of tweets to download, so I was able to obtain a complete distribution of samples of the dataset, while formulating the problem in a way consistent with the proposed use-cases of the algorithms.

4.1.3.1 Sampling of the dataset

Having completely rewritten the download code, I have switched my focus into selecting a reasonable portion of the dataset to be downloaded. As I was unable to download all data because of the Twitter download limits (as described previously), I had to sample only a part of the dataset. Here, I will present three approaches to tackling this challenge.

It is important to note that using each of them, the same set of news articles is downloaded. News articles (corresponding to the claims, which were assessed by the fact-checking organizations) are the main entity to be classified and all of them should be downloaded in order to create comparable variants of the dataset. The only exception are those news articles without sufficient activity (set as at least 20 tweets discussing them, which is also described in later section 4.2 about data preprocessing). What could be changed is the amount of downloaded information for each of these news articles, namely the number of discussing Tweets, the number of related discussion threads connected to these tweets (or the depth of such threads) and the number of information about the users interacting with the tweets.

It is certainly best to have as much available data as possible to create an accurate model. However, in all thinkable applications related to misinformation detection, it is not the only desired attribute of the model. Maybe even more important is the ability to detect misinformation early, before it is spread to too many users, so the negative impact of it is mitigated. In terms of social networks, it is necessary to detect such content using only the first several tens or hundreds of tweets, by chronological ordering.

4.1.3.2 Chronological variant

I have used this reasoning to create the first schema of dataset sampling, naming it the *chronological variant*. For each news article, only the chronologically first 200 tweets related to a claim are downloaded. All information about such tweets are downloaded, as well as the user profiles of the authors of these tweets. No tweets replying to these tweets are downloaded, as the number of those is larger than even the Twitter academic access monthly limit for downloaded tweets.

This variant of the dataset captures the first mentions of a certain news in the social network. The amount of 200 tweets should be large enough to correctly depict the situation in the beginning of the spread of the information. It could be reasonable to expect a misinformation to have some signs of anomaly even on such a small set of related posts.

4.1.3.3 Compact variant

It would be ideal to collect the reply tweets as well. This is something I attempt to do by a variant referenced as the *compact variant*. Again focusing on the tweets from the beginning of the life-cycle of the news, this time I search the first 20 tweets with an interesting discussion revolving around them. The criteria for a tweet to be collected, along with all the reply tweets in the conversation (even all replies-to-replies and so on) was:

- The tweet has to have at least 1 reply recorded (even if it became unavailable later).
- The tweet cannot be a reply tweet itself.
- If the tweet has more than 300 replies, the tweet is collected along with the first 300 replies retrieved by the Twitter API.

The last condition comes from the fact that it is not possible to determine the number of all reply tweets in the whole conversation through the Twitter API (except by downloading them all). The replies are also returned in batches, ordered by latest first, unfortunately. From this incomplete information, only a portion of the discussion threads can be reconstructed, but no other option is feasible.

Overall, this variant collects information related to a much lower number of top-level tweets than the chronological variant, but the inclusion of replies makes it a much more condensed source of information. As a result, this variant takes much more time to be completely collected (several days for the FNN Gossipcop version).

4.1.3.4 Combined variant

This ultimate variant is the combination of the previous two. Any tweet that meets the criteria of one of the variants is included. This results in a variant that contains basic information about the first 200 tweets and also a rich information about several full conversation threads, balancing these two approaches. The technical structure of the download process allows the downloaded files to be combined into one directory structure without the need to fetch it from Twitter again, meaning that this variant is constructible in no additional time.

If not explicitly stated otherwise, this *combined variant* is the default variant used for modeling the graphs in all the experiments presented in this thesis. The visualization of these three variants can be found in figure 4.1.

4.2 Overview of experiments

Before describing the specific experiments, I would like to focus on certain aspects which they all have in common. It is necessary to correctly define the machine learning task and to design the common preprocessing steps and evaluation approaches.

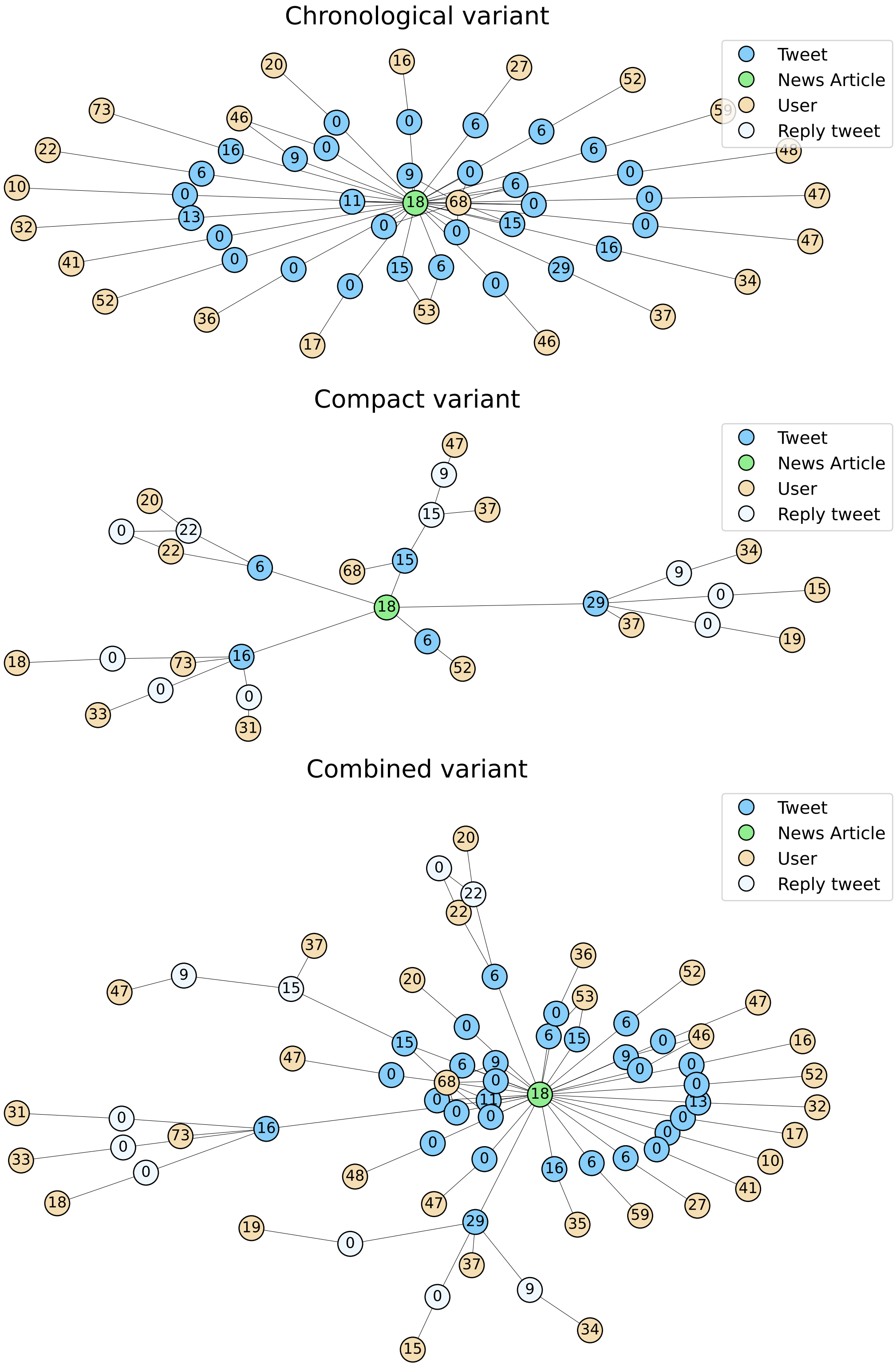


Figure 4.1 The visualization of the differences between dataset sampling variants using the same news article. The numbers assigned to nodes are its discussion importance features, computed and normalized over the whole dataset.

4.2.1 Transformation to a binary classification task

As described previously, anomaly detection is an unsupervised learning task. Throughout all the experiments, this is the key concept and all models will be constructed without the knowledge of any kind of label information. However, to ensure I have constructed them correctly and to select the values for their parameters, the evaluation of models, as well as the final testing used to determine the best models for this task, will utilize labels.

The results of anomaly detection models can vary in its structure, but in its general form, a real value denoting the /emphlevel of anomalousness (sometimes called *anomalousness score*) is assigned to each modeled entity. Some models (or some of their implementations) rather output a list of entities deemed to be anomalous in a more straightforward matter, but it is often possible to observe the anomalousness scores during their computations – for example the local outlier factor algorithm implemented in *scikit learn* [48] library outputs anomalous points, yet it is possible to get the computed outlier scores for each point. When such mapping of entities to anomalousness scores exists, it is possible to design a threshold value and consider all entities with higher values to be anomalous. In fact, this is a basic approach in machine learning to transform a task into a binary classification task.

In this chapter, I will try to map the anomalousness scores into the $[0; 1]$ range for convenience. By doing that, such a value can be intuitively thought of as the predicted *probability* of the entity being misinformation (however, this is only a simplification). In accordance with the meaning of limit values of this range, I have decided to set the misinformation class as the positive class in binary classification. This is consistent with the intended use case – to detect misinformation entities. This decision ensures that basic metrics related to the confusion matrix are no longer ambiguous – for example incorrectly predicting that (what is truly) factual information is misinformation is a *false positive* verdict (type 1 error).

4.2.2 Evaluation metrics

The goal of this thesis is to compare possible approaches and methods in general, rather than to fine tune their performance for a certain setting defined by specific business goals. Because of this, I will use two functions used for evaluation that iterate over all possible threshold values - receiver operating characteristic (ROC) and precision-recall curve. Both of these can be also reduced to one aggregate value, the area under curve (AUC), called average precision (AP) in the case of precision-recall curve.

4.2.3 Data splitting

A crucial step in machine learning experiments is designing the correct training and evaluation process. Specifically, it is necessary to ensure that the resulting evaluation is made using a set of previously unseen data in order to assess the generalization performance of a model. In the perspective of supervised learning, this results in the dogmatic *train-validation-test* splitting of the dataset, where the model is trained using the train data. Validation data is used for *model selection*, which is typically done by selecting the parameters describing the model, or the training process. Moreover, validation often helps in determining the correct length of the training period and therefore protects the model from being *underfitted* (not trained enough yet) or *overfitted* (too much specialized on inputted data without the generalization ability). Finally, the test set of data is used only after final model configuration selection to assess its overall performance in an evaluation completely independent of the training process.

For unsupervised learning techniques, the situation is not that different. The final testing should be performed on a separate set of data. However, what is meant by training and validation steps largely depends on the structure of the model in question, and is described in each experiment's section.

To have an unified testing set upon which all selected models will be evaluated and compared, I have split the Fake News Net Gossipcop dataset into two equally large sets – the training/validation set and the test set. 50 % of news articles were randomly sampled for each class (fake news and real news) and inserted to the test set. The statistics computed for both sets can be found in table 4.1. It shows that the sets are correctly split and these sets do not significantly vary in structure.

	train/validation set	test set
News articles	5 516	5 517
fake news	757	758
real news	4 759	4 759
Total number of tweets	338 960	338 510
Total number of users	331 292	335 540
Total number of replies	38 868	41 747

Table 4.1 Comparison of the train/validation and test sets using FNN Gossipcop dataset, showing they are reasonable balanced.

4.3 Baseline experiments

A standard process in machine learning tasks is to construct a set of *baseline* experiments. Using relatively simple and well known methods that are easily explainable, preliminary results are obtained, which can be later used for comparison while evaluating more sophisticated models. Baseline experiments also help to assess how difficult a task is and which preprocessing steps are needed for certain data.

In this section, several non-graph anomaly models are constructed from the information about the chronologically first related tweets.

4.3.1 Data preprocessing

For these basic experiments, only the chronological variant of the FNN Gossipcop dataset is used. This means that for each news article I have between 20 and 200 tweets, corresponding to the chronologically first mentions of a claim on the social network. For each of the tweets, there is a variety of complex attributes and metainformation – yet to make the model as simple as possible, I have chosen only to extract the information about the impact the tweet had on the users of the network. To approximate this value, I have defined an attribute *discussion importance*, computed as the sum of the number of retweets, likes, quotes and replies to the tweet. Each of these signifies a certain explicit action that other users can perform. What all of these actions have in common is that it shows that the user is somehow interacting with the news, although it is not easy to determine which of these actions signify a stronger interest than others (which is why all are given the same weight).

Using this approach, it is easy to construct a vector of such values for a news article, sorted by the time of publishment of the corresponding tweets. For news with less than 200 tweets (which would result in a shorter vector), the existing values are repeated as many times as needed until the length of the vector is 200.

However, there are two obvious problems with this approach:

- The distribution of the discussion importance feature is very uneven. Most of the time it is smaller than 10, but sometimes the number is significantly higher (maximum is 176 248 for a tweet posted by a major news publisher). It is to be expected that distance-based methods would not be effective using such values.

- There is significant noise in the data. The vectors of values can be seen as time sequences with a large number of peaks, as the values are rather independent of the neighbouring values. This is also not ideal for rather simple models.

To combat these aspects, two alternative approaches are proposed. The first one replaces the discussion importance with its logarithmic value (as depicted in the following equation).

$$Discussion\_importance\_log(tweet) = \ln(Discussion\_importance(tweet) + 1) \tag{4.1}$$

The second one uses this modification and also compresses the information in the vector by taking disjunct continuous subsequences (of a fixed length 20) and computing the sum of those values, effectively outputting vectors of length 10.

4.3.2 Construction of models

The model I have chosen is the local outlier factor (LOF). Its simplicity and a small number of well-interpretable parameters are the desired properties for the baseline experiments. It is also a deterministic algorithm that does not use randomized setups and so its results are robust and easily repeatable. The vectors for all news articles obtained in the previous steps are presented to the model and it calculates the LOF Score, which can be easily scaled into the $[0, 1]$ range as higher values of LOF score denote the anomalies.

4.3.3 Model settings

The only parameter that I have considered worth inspecting in detail is the number of nearest neighbours used for determining the local density around a data point. It is generally not easy to correctly determine this parameter, although it is recommended (in [49])to set it slightly larger than the expected size of coherent clusters appearing in the data . As this value is unknown to me, experimental evaluation is needed. I have used several values of this parameter for each of the data preprocessing variants.

The results are summarized in table 4.2. The small value of 1 has been proved the best for the first two models, which is a strong indicator that the dimension of the vector is too large for the number of graph samples (this is a common obstacle in machine learning, named *curse of dimensionality*). The value of 5 for the third model is a more reasonable parameter settings.

4.3.4 Results

Unsurprisingly, the best results have been achieved by the third variant of preprocessing with the condensed vector. From the two metrics in figure 4.2 it is possible to see that it has the best performance at the whole range of possible thresholds and its ROC AUC is 0.74 and average precision 0.25.

An interesting aspect of the results is that there is a sudden drop in precision right at the start of the precision-recall curve for all models. This indicates that the most anomalous graphs do not belong to the misinformation class. The precision rises after this point and maintains a stable performance for higher recall values.

The main conclusion of these experiments is that even naive models behave significantly differently from random classifiers and are able to outperform it in several metrics. This is a strong indication that anomaly detection can be used for the task of misinformation classification as the two classes can be at least partially separable by their anomaly scores. What remains to be found is whether using graph-based methods can further improve this.

	Baseline		Log Scaling		Condensed Vectors	
Param.	ROC AUC	AP	ROC AUC	AP	ROC AUC	AP
1	**0.599**	**0.156**	**0.679**	**0.193**	0.679	0.206
2	0.587	0.149	0.647	0.172	0.704	0.231
3	0.569	0.142	0.606	0.154	0.720	0.235
4	0.553	0.137	0.555	0.138	0.737	0.240
5	0.544	0.135	0.531	0.132	**0.741**	**0.245**
6	0.531	0.130	0.502	0.123	0.730	0.234
7	0.523	0.128	0.488	0.120	0.733	0.234
8	0.522	0.128	0.474	0.117	0.733	0.233

Table 4.2 The performance of baseline models depending on LOF parameter. The full results can be found in the appendix A.1.

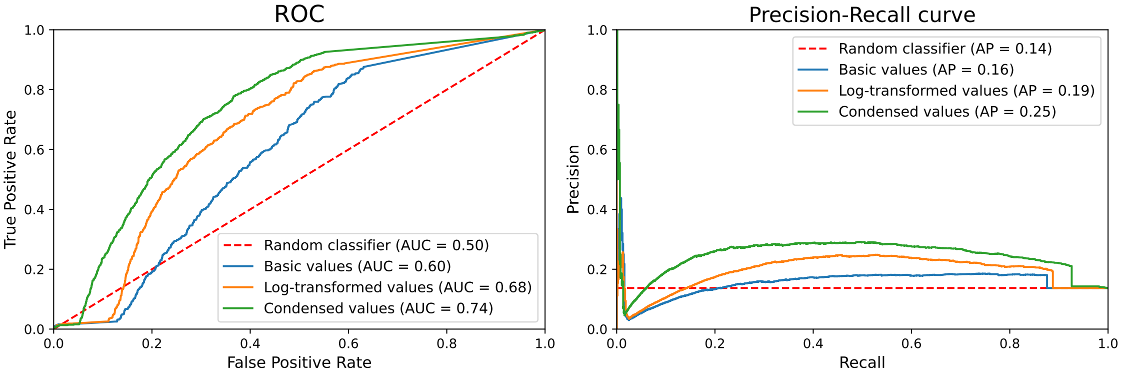


Figure 4.2 The comparison of best baseline models for different preprocessing. Number of neighbours used in LOF model is 1 for the *Basic values* and *Log-transformed values* variants and 5 for the *Condensed values* variant.

4.4 Graph embedding

This section describes the second possible model architecture, which uses two separate steps:

- The embedding for each graph is computed in the form of a vector denoting a point in latent space.
- A traditional outlier detection algorithm computes the anomaly scores using the gathered vectors.

While this approach might seem as suboptimal because the embedding is created by a model that might have other optimization criteria than the separability of the two classes, the significant advantage here is that the two steps are separate and each of them happen independently. Because of that, there are a large number of possible combinations of embedding models and outlier models which can be incorporated into such a process.

I have chosen the combination of the graph2vec embedding algorithm with the local outlier factor algorithm for the experiments in this section.

4.4.1 Graph modeling

The graph2vec algorithm computes embeddings using one discrete node feature. I have once again used the *discussion importance* for tweet nodes as defined in the baseline experiment section. As the algorithm requires that all the nodes in the graph have the same feature, I have modified its definition even for user and news article entities, as summarized in table 4.3.

Entity	Discussion importance definition	Minimum value	Maximum value
News article	Number of collected tweets	20	220
Tweet	Sum of numbers of retweets, replies, quotes and likes	0	176 248
User	Number of followers	0	107 508 063
Reply tweet	Sum of numbers of retweets, replies, quotes and likes	0	5 639

Table 4.3 The definition of discussion importance for different entities. All the metrics are related to Twitter.

As those features are not uniformly distributed, with a small amount of very large values, I have once again transformed these values using the logarithmic function and applied scaling into a $[0; 99]$ range for convenience and rounded these values to nearest integers. This scaling has been done independently for each of the entity types.

4.4.2 Basic graph

In the first embedding experiment, I have used the graph with the structure derived directly form the dataset. I have used the discussion importance feature computed as described before.

4.4.2.1 Parameter selection

The two parameters that have the potential to most significantly influence the results are the number of dimensions of the latent space (which is a parameter of graph2vec algorithm) and number of neighbours in LOF. As can be seen in table 4.4, from the perspective of average precision the best values are 16 dimensions and 5 neighbours, while ROC AUC favors larger values in both parameters. As the experiments did not cover all combinations and the graph2vec algorithm is partially randomized, this process might miss even better settings.

	Emb. size = 16		Emb. size = 32		Emb. size = 64		Emb. size = 128	
LOF param.	ROC AUC	AP	ROC AUC	AP	ROC AUC	AP	ROC AUC	AP
2	0.641	0.223	0.629	0.213	0.629	0.206	0.634	0.205
5	0.664	**0.230**	0.655	0.225	0.649	0.216	0.665	0.218
10	0.652	0.224	0.656	0.229	0.644	0.218	0.649	0.221
15	0.671	0.223	**0.672**	0.228	0.669	0.228	**0.672**	0.226

Table 4.4 The performance of graph2vec + LOF models using basic graph

4.4.3 Adding temporal dimension information

The structure of the graph, along with the discussion importance feature are only two parts of the potential information that can be found in the data. Every tweet gathered through the Twitter API also has several types of temporal information. In the next section, I will try to utilize the timestamps of tweets' publication.

One of the most popular options to do so in the graph research community (used for example in [7] and [12]) is to create edges connecting entities that can be considered close from a time perspective. I will try to utilize two variants of this approach in the following experiments.

The first one will be referred to as *time neighbours variant.* The intuition is that a tweet is likely to influence a certain number of following tweets. In this variant, each tweet is connected to a chosen number of tweets chronologically preceding and following it. For the experiments, this value is set to 3, meaning that each tweet in the middle of the sequence is connected to 6 tweets (3 before and 3 after) if such tweets exist.

The second can be described as *time window variant.* A tweet is connected to each tweet published in a certain time interval centered around the time of publication of said tweet. For the experiments, the value of 10 hours was selected. To prevent the creation of an exhaustive number of edges inside a cluster of tweets posted during a short period of time (creating a clique in the graph), this approach can be combined with the time neighbours variant. In such settings, the edge is created only if it fulfills both requirements (i.e. the degree of tweet nodes are limited).

The comparison of all variants can be found in 4.3.

4.4.3.1 Parameter selection

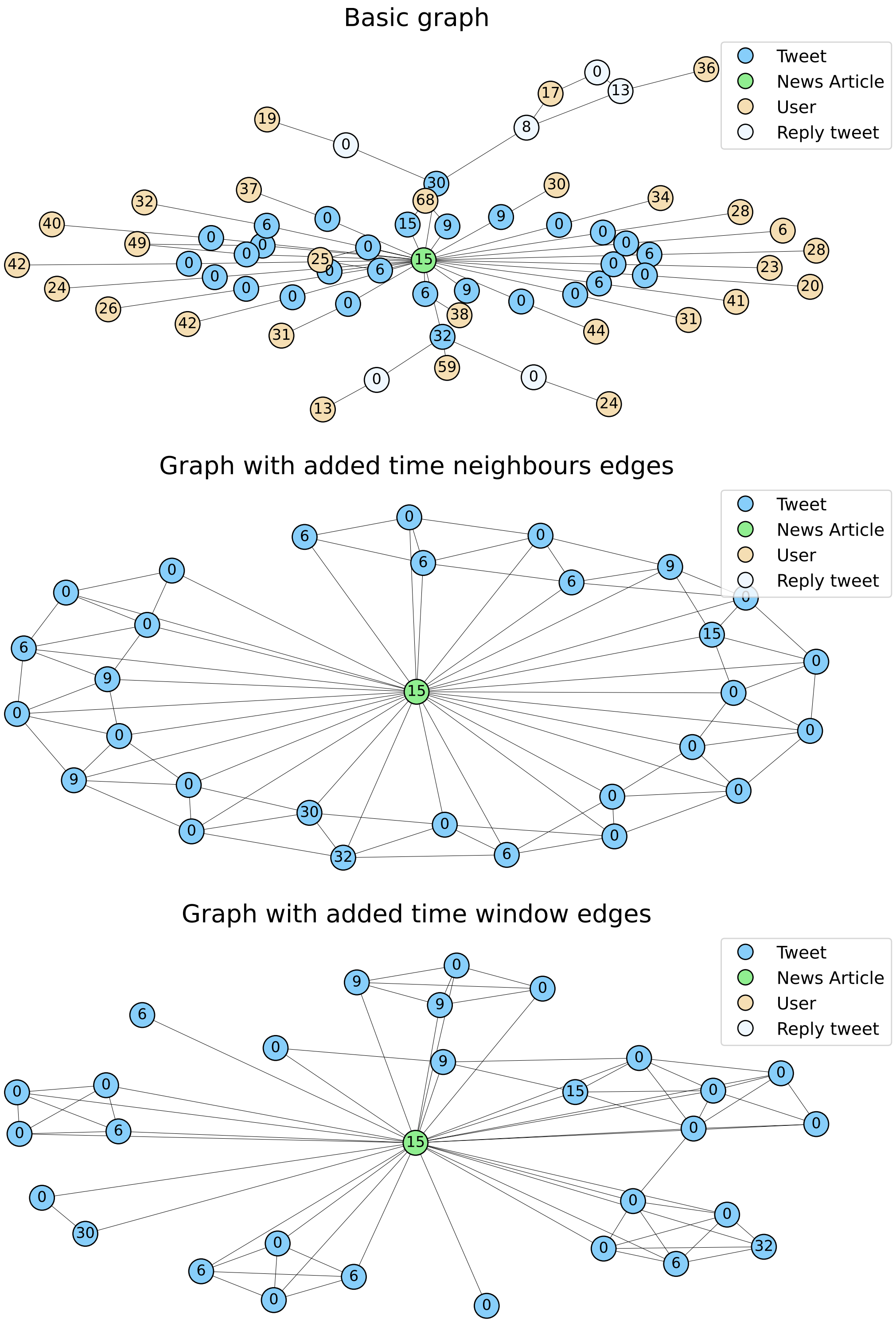
The results can be found int 4.5 and 4.6 respectively. What is interesting is that each modeling variant has significantly different requirements on the parameter values.

	Emb. size = 16		Emb. size = 32		Emb. size = 64		Emb. size = 128	
LOF param.	ROC AUC	AP	ROC AUC	AP	ROC AUC	AP	ROC AUC	AP
2	0.609	0.229	0.605	0.215	0.621	0.214	0.618	0.211
5	0.646	0.237	**0.663**	**0.240**	0.650	0.226	0.646	0.220
10	0.641	0.221	0.644	0.220	0.650	0.221	0.655	0.221
15	0.640	0.208	0.647	0.213	0.645	0.215	0.640	0.210

Table 4.5 The performance of graph2vec + LOF models using added time neighbours edges.

	Emb. size = 16		Emb. size = 32		Emb. size = 64		Emb. size = 128	
LOF param.	ROC AUC	AP	ROC AUC	AP	ROC AUC	AP	ROC AUC	AP
2	0.583	0.182	0.661	0.189	0.599	0.192	0.593	0.190
5	0.633	0.203	0.642	0.208	0.642	0.208	0.648	0.215
10	0.655	0.206	0.661	0.216	0.661	0.215	0.660	0.217
15	0.685	0.216	0.704	0.23	0.703	0.234	**0.711**	**0.238**

Table 4.6 The performance of graph2vec + LOF models using added time window edges.



Basic graph

Graph with added time neighbours edges

Graph with added time window edges

Figure 4.3 The visualization of graphs with added temporal information. The user and reply tweet nodes are omitted from the plots of the second and third graphs for clarity. The second graph contains edges connecting a tweet node with 2 tweets posted previously and subsequently (using parameter `time_neighbours=2` while building the graph). The third graph is constructed using edges between nodes corresponding to the tweets posted in the time window of 10 hours.

4.4.4 Results

The resulting plots can be seen in 4.4. The precision-recall curve shows that no model is clearly superior to another. Each variant of temporal information inclusion has a different range of peak performance.

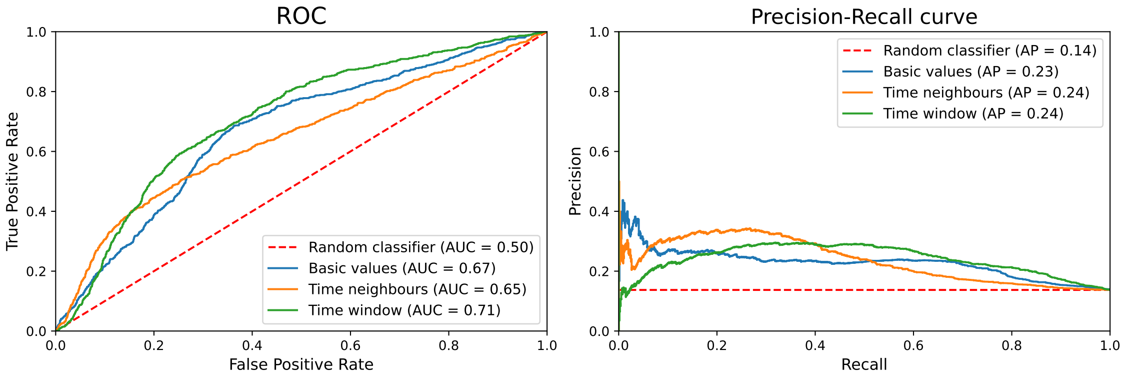


Figure 4.4 The comparison of best graph2vec+LOF models for different graph construction.

4.5 Graph Autoencodes

The most advanced models researched in this thesis are graph autoencoders. There are two variants:

- Traditional node feature graph autoencoder, utilizing several convolutional layers and compressing the node features information for each node. This is rather a simple autoencoder that does not compress the graph structure.
- A autoencoder based on the Dominant model [50], changed for the task of graph anomaly detection (as the original variant is suited for node anomaly detection). The reconstruction error is computed from both the features of nodes (the discussion importance as defined previously and type of the entity of the node) and from the adjacency matrix.

4.5.1 Results

The basic results of one run of each model can be seen in 4.5. The dominant-based model is overall more accurate than the basic graph autoencoder. The precision-recall visualization shows that these models are much better than the others previously mentioned in detecting the obvious misinformation (with high precision regarding the most anomalous samples).

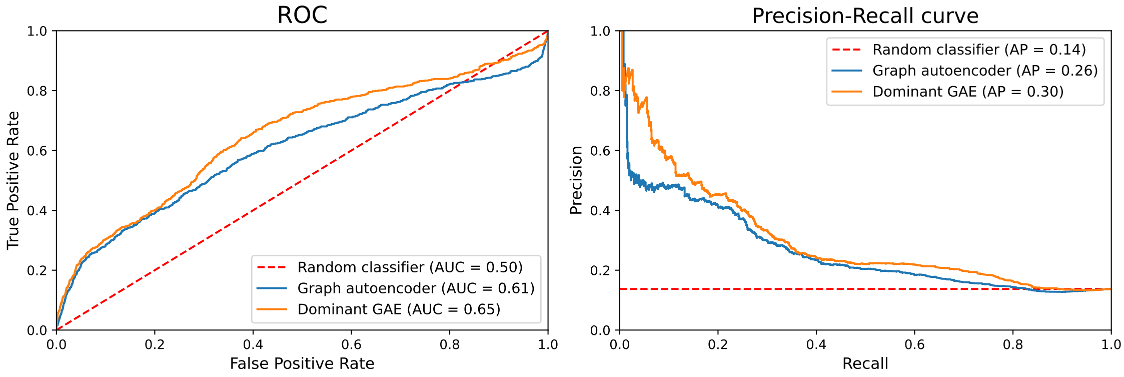


Figure 4.5 The comparison of graph autoencoder models.

Chapter 5

Conclusion

In this thesis, the field of anomaly detection was thoroughly investigated with the focus on graph-based anomaly detection. Several families of approaches on how to utilize relational and temporal information in data were analyzed and the current state of the art of the research in this field was summarized. In the third chapter, an exhaustive analysis of available datasets related to the domain of misinformation detection is proposed. A variety of approaches to the task of misinformation detection was evaluated in the experimental part of this thesis. On a large real-world dataset of message posts on social network Twitter, it has been proved that there is strong evidence that misinformations can be detected using anomaly detection methods. From several of them, graph-based approaches have proved to be accurate even on heavily imbalanced data.

I have provided a careful assessment of a wide field of research and performed a series of experiments to test the hypothesis that fraudulent behavior can be detected by anomaly detection. For this, I consider the goals of this thesis to be reasonably fulfilled.

5.1 Further work

Despite all the effort, there are several aspects that have not been properly studied in this thesis. Many of the experiments depend on precise parameter optimization, which is however a difficult process requiring a large amount of time and knowledge. For this part, I am well aware of the limitations of models presented in the experimental part.

Using only one family of datasets, this work would also benefit greatly by comparing the result using other data. In an ideal scenario, the experiments should be run on significantly larger datasets, potentially not containing labels, as that is the primary use case for using unsupervised learning methods. There has been an effort to obtain a large and very interesting private dataset of online discussions from one of the largest media companies in the Czech Republic, but despite a high effort of my supervisor, at the time submitting this thesis it has not been available for me to use it.

There are also several approaches used in related fields that could be incorporated into this research. For example, there is an observation that individual models presented in the experimental part are good for detecting certain types of anomalies, while failing to uncover another. I think that it is reasonable to assume that using the methods of model ensembling, the detection precision could be significantly improved by carefully combining several models. Unfortunately this is beyond the scope of this thesis.

Appendix A

Extended results of experiments

	Baseline		Log Scaling		Condensed Vectors	
Param.	ROC AUC	AP	ROC AUC	AP	ROC AUC	AP
1	**0.599**	**0.156**	**0.679**	**0.193**	0.679	0.206
2	0.587	0.149	0.647	0.172	0.704	0.231
3	0.569	0.142	0.606	0.154	0.720	0.235
4	0.553	0.137	0.555	0.138	0.737	0.240
5	0.544	0.135	0.531	0.132	**0.741**	**0.245**
6	0.531	0.130	0.502	0.123	0.730	0.234
7	0.523	0.128	0.488	0.120	0.733	0.234
8	0.522	0.128	0.474	0.117	0.733	0.233
9	0.517	0.127	0.465	0.116	0.727	0.226
10	0.513	0.126	0.453	0.113	0.719	0.223
15	0.496	0.122	0.426	0.109	0.706	0.217
20	0.485	0.119	0.409	0.106	0.688	0.206
25	0.480	0.118	0.397	0.104	0.692	0.218
30	0.478	0.118	0.392	0.104	0.687	0.219

Table A.1 The full results showing performance of baseline models depending on LOF parameter.

Bibliography

1. BREUNIG, Markus M.; KRIEGEL, Hans-Peter; NG, Raymond T.; SANDER, Jörg. LOF: Identifying Density-Based Local Outliers. *SIGMOD Rec.* 2000, vol. 29, no. 2, pp. 93–104. ISSN 0163-5808. Available from DOI: `10.1145/335191.335388`.

2. AKOGLU, Leman; TONG, Hanghang, et al. Graph based anomaly detection and description: a survey. *Data Mining and Knowledge Discovery.* 2015, vol. 29, no. 3, pp. 626–688. ISSN 1573-756X. Available from DOI: `10.1007/s10618-014-0365-y`.

3. MA, Xiaoxiao; WU, Jia, et al. A Comprehensive Survey on Graph Anomaly Detection with Deep Learning. *IEEE Transactions on Knowledge and Data Engineering.* 2021. ISSN 1558-2191. Available from DOI: `10.1109/TKDE.2021.3118815`.

4. RANSHOUS, Stephen; SHEN, Shitian, et al. Anomaly detection in dynamic networks: a survey. *WIREs Computational Statistics.* 2015, vol. 7, no. 3, pp. 223–247. ISSN 1939-0068. Available from DOI: `https://doi.org/10.1002/wics.1347`.

5. WANG, Guan; XIE, Sihong, et al. Review Graph Based Online Store Review Spammer Detection. In: 2011, pp. 1242–1247. Available from DOI: `10.1109/ICDM.2011.124`.

6. AKOGLU, Leman; CHANDY, Rishi, et al. Opinion Fraud Detection in Online Reviews by Network Effects. *Proceedings of the 7th International Conference on Weblogs and Social Media, ICWSM 2013.* 2013, vol. 7, pp. 2–11. Available from DOI: `10.1609/icwsm.v7i1.14380`.

7. DOU, Yingtong; LIU, Zhiwei, et al. Enhancing Graph Neural Network-Based Fraud Detectors against Camouflaged Fraudsters. In: *Proceedings of the 29th ACM International Conference on Information and Knowledge Management.* Virtual Event, Ireland: Association for Computing Machinery, 2020, pp. 315–324. CIKM '20. ISBN 9781450368599. Available from DOI: `10.1145/3340531.3411903`.

8. ZHANG, Ge; LI, Zhao, et al. EFraudCom: An E-Commerce Fraud Detection System via Competitive Graph Neural Networks. *ACM Trans. Inf. Syst.* 2022, vol. 40, no. 3. ISSN 1046-8188. Available from DOI: `10.1145/3474379`.

9. CHAU, Duen Horng; PANDIT, Shashank, et al. Detecting Fraudulent Personalities in Networks of Online Auctioneers. In: *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases.* Berlin, Germany: Springer-Verlag, 2006, pp. 103–114. ECMLPKDD'06. ISBN 3540453741. Available from DOI: `10.1007/11871637_14`.

10. PANDIT, Shashank; CHAU, Duen Horng, et al. Netprobe: A fast and scalable system for fraud detection in online auction networks. In: 2007. Available from DOI: `10.1145/1242572.1242600`.

11. TSANG, Sidney; KOH, Yun Sing, et al. SPAN: Finding collaborative frauds in online auctions. *Knowledge-Based Systems*. 2014, vol. 71, pp. 389–408. ISSN 0950-7051. Available from DOI: `https://doi.org/10.1016/j.knosys.2014.08.016`.

12. LIU, Guannan; GUO, Jia, et al. Fraud detection via behavioral sequence embedding. *Knowledge and Information Systems*. 2020, vol. 62, no. 7, pp. 2685–2708. ISSN 0219-3116. Available from DOI: `10.1007/s10115-019-01433-3`.

13. LI, Zhongmou; XIONG, Hui, et al. Mining blackhole and volcano patterns in directed graphs: a general approach. *Data Mining and Knowledge Discovery*. 2012, vol. 25, no. 3, pp. 577–602. ISSN 1573-756X. Available from DOI: `10.1007/s10618-012-0255-0`.

14. HUANG, Dongxu; MU, Dejun, et al. CoDetect: Financial Fraud Detection With Anomaly Feature Detection. *IEEE Access*. 2018, vol. 6, pp. 19161–19174. Available from DOI: `10.1109/ACCESS.2018.2816564`.

15. WEBER, Mark; DOMENICONI, Giacomo, et al. *Anti-Money Laundering in Bitcoin: Experimenting with Graph Convolutional Networks for Financial Forensics*. arXiv, 2019. Available from DOI: `10.48550/ARXIV.1908.02591`.

16. TAM, Nguyen Thanh; WEIDLICH, Matthias, et al. From Anomaly Detection to Rumour Detection Using Data Streams of Social Platforms. *Proc. VLDB Endow.* 2019, vol. 12, no. 9, pp. 1016–1029. ISSN 2150-8097. Available from DOI: `10.14778/3329772.3329778`.

17. RUCHANSKY, Natali; SEO, Sungyong, et al. CSI: A Hybrid Deep Model for Fake News Detection. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. Singapore, Singapore: Association for Computing Machinery, 2017, pp. 797–806. CIKM '17. ISBN 9781450349185. Available from DOI: `10.1145/3132847.3132877`.

18. SONG, Chenguang; SHU, Kai, et al. Temporally evolving graph neural network for fake news detection. *Information Processing Management*. 2021, vol. 58, no. 6, p. 102712. ISSN 0306-4573. Available from DOI: `https://doi.org/10.1016/j.ipm.2021.102712`.

19. NGUYEN, Van-Hoang; SUGIYAMA, Kazunari, et al. FANG: Leveraging Social Context for Fake News Detection Using Graph Representation. *Commun. ACM*. 2022, vol. 65, no. 4, pp. 124–132. ISSN 0001-0782. Available from DOI: `10.1145/3517214`.

20. VLACHOS, Andreas; RIEDEL, Sebastian. Fact Checking: Task definition and dataset construction. In: *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*. Baltimore, MD, USA: Association for Computational Linguistics, 2014, pp. 18–22. Available from DOI: `10.3115/v1/W14-2508`.

21. FERREIRA, William; VLACHOS, Andreas. Emergent: a novel data-set for stance classification. In: *Proceedings of NAACL-HLT 2016*. Association for Computational Linguistics, 2016, pp. 1163–1168. Available from DOI: `10.18653/v1/N16-1138`.

22. WANG, William. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 422–426. Available from DOI: `10.18653/v1/P17-2067`.

23. ZUBIAGA, Arkaitz; LIAKATA, Maria, et al. Learning Reporting Dynamics during Breaking News for Rumour Detection in Social Media. *CoRR*. 2016, vol. abs/1610.07363. Available from arXiv: `1610.07363`.

24. KOCHKINA, Elena; LIAKATA, Maria, et al. All-in-one: Multi-task Learning for Rumour Verification. *CoRR*. 2018, vol. abs/1806.03713. Available from arXiv: `1806.03713`.

25. AUGENSTEIN, Isabelle; LIOMA, Christina, et al. MultiFC: A Real-World Multi-Domain Dataset for Evidence-Based Fact Checking of Claims. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 4685–4697. Available from DOI: 10.18653/v1/D19-1475.

26. SHU, Kai; WANG, Suhang, et al. Exploiting Tri-Relationship for Fake News Detection. *CoRR*. 2017, vol. abs/1712.07709. Available from arXiv: 1712.07709.

27. SHU, Kai; MAHUDESWARAN, Deepak; WANG, Suhang; LEE, Dongwon; LIU, Huan. FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information for Studying Fake News on Social Media. *Big Data*. 2020, vol. 8, pp. 171–188. Available from DOI: 10.1089/big.2020.0062.

28. CUI, Limeng; LEE, Dongwon. CoAID: COVID-19 Healthcare Misinformation Dataset. *CoRR*. 2020, vol. abs/2006.00885. Available from arXiv: 2006.00885.

29. LI, Yichuan; JIANG, Bohan, et al. MM-COVID: A Multilingual and Multimodal Data Repository for Combating COVID-19 Disinformation. *CoRR*. 2020, vol. abs/2011.04088. Available from arXiv: 2011.04088.

30. POYNTER INSTITUTE. *International Fact-Checking Network* [https://www.poynter.org/ifcn/]. 2022. [Online; accessed 30-October-2022].

31. DOU, Yingtong; SHU, Kai, et al. User Preference-Aware Fake News Detection. In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Virtual Event, Canada: Association for Computing Machinery, 2021, pp. 2051–2055. SIGIR '21. ISBN 9781450380379. Available from DOI: 10.1145/3404835.3462990.

32. NICKERSON, Raymond. Confirmation Bias: A Ubiquitous Phenomenon in Many Guises. *Review of General Psychology*. 1998, vol. 2, pp. 175–220. Available from DOI: 10.1037/1089-2680.2.2.175.

33. PYTORCH GEOMETRIC. *PyG is the ultimate library for Graph Neural Networks, built upon PyTorch.* [https://www.pyg.org/]. 2022. [Online; accessed 30-October-2022].

34. NIELSEN, Dan S.; MCCONVILLE, Ryan. MuMiN: A Large-Scale Multilingual Multimodal Fact-Checked Misinformation Social Network Dataset. In: *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Madrid, Spain: Association for Computing Machinery, 2022, pp. 3141–3153. SIGIR '22. ISBN 9781450387323. Available from DOI: 10.1145/3477495.3531744.

35. GOOGLE LLC. *Google Fact Check Tool APIs* [https://toolbox.google.com/factcheck/apis]. 2022. [Online; accessed 6-November-2022].

36. THORNE, James; VLACHOS, Andreas, et al. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, 2018, pp. 809–819. Available from DOI: 10.18653/v1/N18-1074.

37. BOIDIDOU, Christina; ANDREADOU, Katerina, et al. Verifying Multimedia Use at MediaEval 2015 In MediaEval Benchmarking Initiative for Multimedia Evaluation. In: *MediaEval 2015 Workshop*. 2015. Available also from: https://iris.unitn.it/bitstream/11572/121886/2/Verif2015.pdf.

38. NI, JIANMO. *Amazon Review Data* [https://nijianmo.github.io/amazon/index.html]. 2018. [Online; accessed 2-September-2022].

39. YELP.COM. *Yelp Open Dataset* [https://www.yelp.com/dataset]. 2022. [Online; accessed 2-September-2022].

40. OUTLIER DETECTION DATASETS. *YelpCHI dataset* [http://odds.cs.stonybrook.edu/yelpchi-dataset/]. 2022. [Online; accessed 2-September-2022].

41. INTERNATIONAL CONSORTIUM OF INVESTIGATIVE JOURNALISTS. *Offshore Leaks Database* [https://offshoreleaks.icij.org/]. 2022. [Online; accessed 2-September-2022].

42. HU, Weihua; FEY, Matthias, et al. Open Graph Benchmark: Datasets for Machine Learning on Graphs. *CoRR*. 2020, vol. abs/2005.00687. Available from arXiv: 2005.00687.

43. GUO, Xinjian; YIN, Yilong; DONG, Cailing; YANG, Gongping; ZHOU, Guangtong. On the class imbalance problem. In: *2008 Fourth international conference on natural computation*. IEEE, 2008, vol. 4, pp. 192–201. Available from DOI: 10.1109/ICNC.2008.871.

44. LONGADGE, Rushi; DONGRE, Snehalata. Class imbalance problem in data mining review. *International Journal of Computer Science and Network*. 2013, vol. 2. ISSN 2277-5420. Available also from: http://ijcsn.org/IJCSN-2013/2-1/IJCSN-2013-2-1-58.pdf.

45. TWITTER, INC. *Developer Agreement and Policy* [https://developer.twitter.com/en/developer-terms/agreement-and-policy]. 2023. [Online; accessed 1-January-2023].

46. SHU, Kai; MAHUDESWARAN, Deepak. *Fake News Net* [https://github.com/KaiDMML/FakeNewsNet]. 2023. [Online; accessed 1-January-2023].

47. TWITTER, INC. *Standard v1.1* [https://developer.twitter.com/en/docs/twitter-api/v1]. 2023. [Online; accessed 1-January-2023].

48. PEDREGOSA, Fabian; VAROQUAUX, Gael, et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011, vol. 12, no. 85, pp. 2825–2830. Available also from: http://jmlr.org/papers/v12/pedregosa11a.html.

49. SCIKIT-LEARN. *Outlier detection with Local Outlier Factor (LOF)* [https://scikit-learn.org/stable/auto_examples/neighbors/plot_lof_outlier_detection.html]. 2023. [Online; accessed 3-January-2023].

50. DING, Kaize; LI, Jundong, et al. Deep Anomaly Detection on Attributed Networks. In: 2019, pp. 594–602. ISBN 978-1-61197-567-3. Available from DOI: 10.1137/1.9781611975673.67.

Contents of attached storage medium

- `readme.txt` ... overview of the contents of storage medium
- `src`
  - `experiments` ... source codes of the implementation of the experiments
  - `thesis` ... thesis text source code in LaTeX format
- `text` ... the text of the thesis
  - `thesis.pdf` ... text of the thesis in the PDF format