UNIVERSITY
OF TECHNOLOGY
SYDNEY

Misbehaviour Detection Algorithms and Application in Social Networks

by Jun Yin

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Guandong Xu

University of Technology Sydney
Faculty of Engineering and Information Technology

the September 2021

Certificate of Original Authorship

I, *Jun Yin* declare that this thesis, is submitted in fulfilment of the requirements for the award of *Doctor of Philosophy*, in the *School of Computer Science*, *Faculty of Engineering and Information Technology* at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:

SIGNATURE: Signature removed prior to publication.

[Jun Yin]

DATE: 10th September, 2021

DEDICATION

To myself . . .

Acknowledgments

First of all, I would like to thank Prof. Guandong Xu, my principal supervisor, for his guidance, suggestion, and support throughout my doctoral program course at the University of Technology Sydney. Without his professional guidance and support, this work would not have been achieved.

I am also grateful to my previous co-supervisor Dr. Shaowu Liu for his constructive suggestions on my research work. I would like to pass my gratitude to my current co-supervisor and friend Dr. Qian Li, for her selfless help and suggestions on my work.

In addition, a thank you to Prof. Zhiang Wu of Nanjing Audit University, who introduced me to social network behaviour analysis, and whose enthusiasm for the research work had lasting effect.

Last but not least, I am thankful to my beloved husband, Wei Yuan, and my baby boy Adam J. Yuan, for their love and encouragement which helps me finish this thesis. I am also thankful to my parents and my parents in law for their support throughout my research studies.

Jun Yin
Sydney, Australia
September, 2021

Abstract

The liberty to contribute content freely has encouraged malicious users to exploit the social platforms (i.e., social networks and e-commerce platforms) for their benefits. Spammers, rumours, and some other unexpected activities are almost an appendage to all social platforms that disrupt the network order. We summarize these unexpected activities as misbehaviours in social platforms. To detect such social platform misbehaviours, machine learning is an expected method where modelling and algorithms are two significant elements. Such an interesting topic that has application prospects and research value has attracted the attention of many researchers, and some results have also been put forward in the literature.

In terms of spammer detection, because of the rich data types of e-commerce platform, such as score, comment content, and comment time, the mainstream detection methods rely on the above data to construct features on e-commerce platforms. For example, text-based features, behaviour features etc. in conjunction with some supervised learning algorithms like Naive Bayes, Decision trees etc. are the most frequently used combinations for spammer detection in e-commerce platforms. However, social networks are based on interaction data and are relatively deficient in data types, thus, spammer detection in social networks requires a detection framework that relies on relational data but is independent of content data. Along this line, the existing research attempts to define complex network features (e.g., degree, K-Core, PageRank, connected component, etc.) and interactive sequence-based features. Nevertheless, the deep semantic information hidden in the multi-relational networks has not been fully utilized.

Furthermore, rumour as another type of misbehaviours in social networks has been run through the whole evolutionary history of mankind. People maliciously disseminate rumours to increase awareness, slander others or cause panic, etc. To eliminate this issue, many researchers resort to detecting rumours in social networks. However, rumour detection is not sufficient to eliminate the negative impact, which also requires official institutions to provide the refutations. In practice, the number of rumours in social networks is too large, there is no need to refute some rumours with little or no concern. Therefore, an evaluation of the impact of the rumours in advance is essential.

To address the aforementioned research problems, a few approaches are proposed in the works introduced in this thesis.

- Based on the non-content data, we fully excavate the deep semantic information hidden in the heterogeneous network and define a series of user behaviour features using relational network data for spammer detection.

- Based on the graph embedding method, we propose a “Send-Receive” Role Separable Graph-Embedding Model (*RS-GEM*) to extract and fuse the hidden features of heterogeneous relations in multi-relational social networks to detect spammers.

- Inspired by deep sequential networks, we propose a “Multi-level Dependency Model” (*MDM*), which exploits user’s behaviours in terms of long-term and short-term dependency from both individual-level and union-level to detect multi-relational social spammers.

- Before a rumour has an impact on social networks, we need to assess the possible impact it may have. Therefore, we devise a rumour influence prediction model *RISM* (Rumour Impact on Social Media) based on a popular rumour intensity formula to predict the impact of a newborn rumour.

Last but not least, since the global outbreak of COVID-19 in early 2019, COVID-19-related topics have become hot spots on social networking platforms. At the end of this thesis, we shall also analyze the COVID-19-related tweets on Twitter and get a preliminary understanding of the public’s focus and sentiment trends during the pandemic.

Keywords: Multi-relational Social Network; Behaviour Analysis; Spammer Detection; Feature Construction; Rumour Analysis; Sentiment Analysis

LIST OF PUBLICATIONS

RELATED TO THE THESIS :

1. Yin, J., Zhou, Z., Liu, S., Wu, Z., & Xu, G. (2018, June). Social Spammer Detection: A Multi-Relational Embedding Approach. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 615-627). Springer, Cham.

2. Yin, J., Liu, S., Li, Q., & Xu, G. (2019, October). Prediction and Analysis of Rumour's Impact on Social Media. In 2019 International Conference on Behavioral, Economic, Socio-cultural Computing (BESC) (pp. 1-6). IEEE.

3. Yin, J., Li, Q., Liu, S., Wu, Z., & Xu, G. (2020, November). Leveraging Multi-level Dependency of Relational Sequences for Social Spammer Detection. Neurocomputing, vol. 428, pp. 130-141.

4. Yin, J., Li, Q., Liu, S., & Xu, G. Social Network Analysis of Twitter User's Behaviour during COVID-19 Pandemic. Prepared to be submitted as a Journal Paper.

OTHERS :

5. Zhou, Z., Liu S., Xu, G., Xie, X., Yin, J., Li, Y., & Zhang, W. (2018, June).Knowledge-Based Recommendation with Hierarchical Collaborative Embedding. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 222-234). Springer, Cham.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Social network is a vital part of information propagation. As social network becoming a more effective platform, it also greatly benefits both large organization and individuals in the process. The potential benefits have led to the widespread manufacture and dissemination of false comments and fraudulent information, summarizing as *misbehaviours* in social networks. Misbehaviours can be active, like spams, which are known as unwanted activities, study has shown that roughly 83% of social networks' users have received more than one unwelcome friend request or message [74]. In addition to spams, rumours are another typical representation of misbehaviours, knowing as a piece of information being circulated around in the public while its veracity and authenticity are yet to be verified [98], and it might be passive. Regardless of whether they are active or passive, these users and the misbehaviours conducted by them are putting sustainable development environment of online social networks at a great threat. Hence, it is significant to detect misbehaviours, and prevent them from continuing undermine the order of social networks.

The nature of detection issues have been clearly described as a classification problem [29] taking advantages of machine learning methods. Technically, most of the detection methods first try to define a set of features for indicating the abnormal characteristics, and modelling them in a mathematical way, and hence design a beneficial machine learning algorithms to train a classifier [56, 57]. It is noteworthy that features extracted for each target can be very different. Identifying the right features and modelling them in an appropriate way significantly improved the detection performance, and it is both data-specific and task-specific. That is, a right feature should be computable on the available data and it should also be qualified for the specific detection task. Besides, a beneficial algorithm should be designed for prepared model as well, which can be generally concluded as supervised, unsupervised and semi-supervised algorithms. This thesis will explore both modelling scheme and algorithm, focusing on two applications of social network misbehaviour detection: spammer detection and rumour analysis (especially in Chinese social networks).

In terms of spammer detection, according to the analysis on Twitter spam [32], malicious content is usually wrapped by a suspicious URL. Once users click on these URLs, they will be taken to some malicious pages, such as phishing websites, adult websites, etc. So, content containing URLs can be a resultful feature for Twitter. Along this line, in the literature, an extensive body of research has been devoted to construct features for various kinds of spams, such as email spam [65], web spam [19, 95], review/reviewer spam on e-commerce sites [29, 84] and social network spam [44, 74] by fully exploiting the metadata such as rating, timestamps and review text. Nevertheless, with the rapid development of social networks, the types of relations in social networks are becoming more and more complex. On the contrary, text information is much less than before and it is relatively short and refined. Thus, the previous analysis of semantics and timestamps of comments cannot detect the spammers in the social networks accurately, for the reason that it might be effortless for sophisticated spammers to pass these kinds of filters and pretend themselves like normal users. Network-topological and sequence-based modelling method were proposed to meet this problem, as the network-topological structure of social network users are harder to manipulate. However, omissions still exist, i.e., the interaction information between different relations has been ignored, the hidden information behind relations haven't been fully explored etc. Hence, social spammer detection calls for the relation-dependent but content-independent modelling method.

Further to spammer detection, in the rumour classification process, it usually begin with recognising that a piece of information is not confirmed and ends by determining the estimated veracity value of that piece of information (i.e., veracity classification). The entire process from rumour detection to veracity classification is containing the processes of rumour tracking and stance classification in the middle [98]. The first step is much more similar with spammer detection, while the main focus is on the text content. For example, a rumour detection could have a stream of social media posts as its input, and a binary classifier would subsequently determine whether the post is a rumour or a non-rumour. The component then output the same stream of posts with labels on each post that indicating whether the post is rumour or non-rumour. However, as social network is increasingly being used as source of information and news, with its immoderate nature, it leads to the surging of number of rumours online. Hence, verifying the authenticity of each rumour from social network is not feasible. Besides, from the perspective of refutation, paying too much attention on little or even harmless rumours is not cost-effective. Therefore we need to have filters to alleviated this issue before the next step. For rumours with a higher impact on social media, we need to pay much more attention to check their authenticity, and post the refutations accordingly. On the contrary, it might not be necessary to check its authenticity specifically. Meanwhile, in Chinese social networks, users communicate in Chinese, which is slightly different from English in the analysis process, especially in the process of word segmentation. Because the Chinese system is more complicated, while the English expression is relatively more straightforward.

In view of the above problems, the work of this thesis will be carried out from two aspects: spammer detection and rumour analysis (especially in Chinese social networks), which will be introduced in details in the following chapters.

Since the global outbreak of COVID-19 in early 2019, COVID-19-related topics have become hot spots on social networking platforms. At the end of this thesis, we shall also analyze the COVID-19-related tweets on Twitter, and get a preliminary understanding of the public's focus and sentiment trends during the pandemic.

1.2 Research Objectives

The research goal of this article is to fully analyze the characteristics of misbehaviours in social networks and minimize its harm to social networks as much as possible. Specifically, We mainly analyze two kinds of misbehaviours in social networks: spammers and rumours.

For spammer detection in social networks, this thesis aims to achieve the following objectives.

- **Behaviour analysis and feature construction.** Fully exploring the deep semantic information hidden in the heterogeneous relation network, thereby defining a series of behaviour characteristic indicators based on the relation network data.

- **Graph-embedding based social spammer detection.** Considering the differences in the roles of normal users and spammers in different interaction relations (such as initiator and receiver) and the need for fusion of different relations, we shall utilize graph embedding method to help with extraction and fusion latent features in heterogeneous relational networks.

- **Sequence-based social spammer detection.** Comprehensively mining the abnormal characteristics of spammers in relational sequences in heterogeneous relational networks from the perspectives of both long-term and short-term.

As for rumour analysis in social networks, this thesis aims to achieve the following objectives.

- **Definition of rumour impact in social networks.** Based on the popular rumour intensity formula, we shall propose a definition for the impact of rumour in social networks, which is suitable for most mainstream social networks.

- **Predict the potential impact of a rumour in its early stage.** Proposing a new model that is applicable to most mainstream social networks, which can predict the potential influence of a newborn rumour in the early stage.

In terms of COVID-19 related analysis, we aim to get a preliminary understanding of the public's sentiment trending during the pandemic. We utilize social networks i.e., Twitter, as our research platforms to achieve the following objectives.

- **Changes in topics that public concern during the development of the pandemic.** Analyzing the COVID-19 related tweets in Twitter and uncovering the hot topics of public concern and analyze the changes of hot topics over time.

- **Sentiment trends toward COVID-19 related topics.** Utilizing machine learning method to analyze the sentiment of COVID-19 related tweets in Twitter and discovering the difference of sentiment across different topics.

1.3 Proposed Approaches

In order to achieve the research objectives mentioned in the last section, in works of this thesis, we propose the following categories of methods and techniques.

1.3.1 Proposed Approaches for Spammer Detection

The work of the first part of this thesis is based on a real-world social network data set from *Tagged.com*, which was released along with the publication of the literature at the top international conference SIGKDD-2015 [27]. It has the following characteristics and problems:

1. *Tagged.com* is a multi-relational social network that includes 7 types of different relations. However, each relation is given by a numerical name (i.e. 1, 2, 3, $\cdots$) instead of their corresponding semantic meanings (i.e., sending message, add friend, viewing profile, etc.). The work in [27] did not consider the difference in behaviour characteristics caused by the relation type, but simply extracted network-topological features on each relation.

2. This dataset gives each user a ground-truth label (normal user or spammer), which is urgently needed and rare precious information in the field of social spammer detection, which will provide convenient for evaluating the effectiveness of features and detection performance.

3. The *Tagged.com* dataset is extremely large, including about 860 million records from about 5.6 million users in 7 relations within 10 days, which is very difficult for the process of detection.

In view of the characteristics of the multi-relational social network *Tagged.com*, we propose the following approaches for spammer detection in multi-relational social networks.

- **Behaviour analysis and feature construction.** Based on the non-content data, we fully excavate the deep semantic information hidden in the heterogeneous network, and define a series of user behaviour features using relational network data. Since the dataset from *Tagged.com* does not publish the name of each relational attribute publicly, we first analyze the types of relations according to the data characteristics, and then deduce the name of the actual relation type corresponding to each attribute. Secondly, the behaviour patterns of spammers and normal users are compared and analyzed in each relation with the consideration of each relation type's meaning. Based on non-content information, such as active time, send/receive ratio, and the proportion of response after sending, a series of feature indicators have been given. Finally, we validate the performance of features with the help of user labels provided in the *Tagged.com* dataset.

- **Graph-embedding based social spammer detection.** We propose a “Send-Receive” Role Separable Graph-Embedding Model (*RS-GEM*) to extract and fuse the hidden features of heterogeneous relations. First, we build a graph in a shared embedding space, where nodes represent for users and edges represent for relations between users. Second, the number of interactions between the sending and receiving users is extracted as interaction vectors. Third, the sending user feature matrix and receiving user feature matrix are constructed, and the user-user interaction vector is represented by dot product. The difference between these two vectors is used to fit the probability matrix decomposition model, and the constraint conditions are added to prevent the overfitting problem in the optimization process. Finally, the hidden features of each user in multi-relational social networks are obtained through multi relational mosaic. Cross validation results show that the latent features extracted by *RS-GEM* contribute significantly in the area of multi-relational social network spammer detection.

- **Sequence-based social spammer detection.** We propose a novel *multilevel dependency model* (*MDM*) that exploits user behaviour for social network spammer detection in terms of long-term and short-term dependencies. In particular, in the case of short-term dependencies, we explore both individual-level and union-level perspectives. Individual-level dependencies only consider user’s individual recent behaviours that may trigger subsequent behaviours. Comparatively, union-level dependencies take into account the collective influence between the relational unions involved in a user’s short-term behavioural sequence. Our proposed *MDM* is able to uncover deeper information hidden behind user relational sequences, thus improving the performance of spammer detection in multi-relational social networks.

1.3.2 Proposed Approaches for Rumour Analysis

More proposed methods of rumour detecting are emerging recently, meanwhile, rumours in social networks also grows rapidly. By only identifying rumours does not resolve the negative impact on the public, official refutations are required. However, refuting every rumour from social network is not necessary and impossible as the number of rumours is humongous. Therefore, we address the two challenges shown as below.

1. How to define the impact of rumours on social networks?

2. How to predict the possible impact of rumour at its early stage?

According to the challenges, we shall propose the following approaches.

- **Measuring rumour impact.** Based on the rumour intensity equation proposed by Chorus ($R = I * A/C$), we define rumour impact score that is adaptable on most social networks. Utilizing the content text in conjunction with the statistical information, we give each element (i.e., Importance, Ambiguity and Public Critical Ability.) of the rumour intensity equation a detailed definition.

- **Content-based feature extraction.** Our aim is to predict the impact of the rumour at the very beginnings of its appearance. However, when a rumour first appears, usually there is no other related attributes apart from its content. Therefore, the features are extracted based only from its content. In specific, there are two parts of features we extracted from the content, part one, *TF-IDF*, which is mostly used in text mining, and part two, *Word to Vector*, which represents the semantic information hidden in the text.

1.3.3 Proposed Approaches for COVID-19 Related Analysis

We shall use the dataset collected from Twitter to analysis the discussions on Twitter related to COVID-19 and to investigate public's sentiments toward COVID-19 during different period of time. The research questions that we aim to address can be summarized as below.

1. How has the public's focus on COVID-19-related topics changed over time?

2. What is the difference between the public's sentiment trends toward COVID-19-related topics?

3. Has the trend of public sentiment on COVID-19-related topics changed over time?

According to the research questions, we shall take the existing machine learning method, i.e., TF-IDF, LDA and the help of Valence Aware Dictionary and Sentiment Reasoner (VADER) to analyse the public's sentiment towards COVID-19 over time.

1.4 Thesis Outlines

An overview outlines of this thesis is presented as follows.

Chapter 2. A literature review is given in this chapter to show the background of the misbehaviour analysis in social networks. This chapter mainly consists of two aspect of misbehaviours, namely "spammers" and "rumours".

In terms of spammer detection, we introduce from e-commerce platforms to social networks and make a comparison between the spammer detection in these two mainstream media networks.

As for rumour analysis, four steps of rumour classification are illustrated, i.e., rumour detection, rumour tracking, stance classification and veracity classification. Then, we introduce the research on rumour detection in details, as the rumour detection is the first and non-ignorable process in all four steps. During the literature review, we find that there is quite a few research on the impact of rumours, though the research on rumour detection is in full swing. And this kind of thinking triggers our work on the analysis of rumour impact.

Part I Spammer Detection. This part consists of three chapters (Chapter 3 to Chapter 5), which shall propose three different kinds of methods for spammer detection in multi-relational social networks.

Chapter 3. Data description and behaviour analysis. Firstly, we introduce a real-world dataset from *Tagged.com*, which will be used for our following works on spammer detection in multi-relational social networks. Then, according to the statistic analysis of each relation in the dataset, we speculate the name of the specific relation type corresponding to the numerical relation ID. Subsequently, we carry out relational semantic mining based on this dataset, analyze and compare the behavioural differences between spammers and normal users in each relation, and propose a series of behaviour characteristic indicators based on the relational network. Finally, we utilize cumulative distribution to verify the effectiveness of the proposed behaviour indicators.

Chapter 4. Graph-embedding based social spammer detection. We propose a "Send-Receive" Role Separable Graph-Embedding Model (*RS-GEM*) to meet the needs of extracting and fusing the latent factor hidden behind the heterogeneous relations in social networks. *RS-GEM* first extracts the number of interactions from the initiator to the receiver on each relation as the interaction vector. Then *RS-GEM* constructs the initiator feature matrix and the receiver feature matrix, and use the dot product to represent the user-user interaction vector. The difference between these two vectors is used to fit the probability matrix factorization model, and constraints are added to prevent overfitting in the optimization process. Finally, the hidden feature of each user is obtained through the multi-relation splicing method.

Chapter 5. Sequence-based social spammer detection. The spammer detection problem is investigated in this chapter in the context of multi-relational social network and also attempting to enhance the detection accuracy by fully exploit the sequences of heterogeneous relations. In specific, Multi-level Dependency Model(*MDM*) is presented. Long-term dependency can be exploited by *MDM* in users' relational sequences along with short-term dependency. Furthermore, *MDM* fully excavate short-term relational sequences from both individual-level and union-level.

Part II Rumour Analysis. This part shows our work on another kind of misbehaviours in social networks: rumour.

Chapter 6. In this chapter, we devise a rumour influence prediction model *RISM* (Rumour Impact on Social Media) based on a popular rumour intensity formula to predict the impact of a newborn rumour. Extensive numerical experiments are carried out on the real rumour data that are collected from *Toutiao.com*, which demonstrate the effectiveness of the proposed *RISM*.

Chapter 7. In this chapter, we first introduce the dataset that we collect from Twitter related to COVID-19. And then, we utilize the existing machine learning methods, i.e., LDA and the help with Valence Aware Dictionary and Sentiment Reasoner (VADER) to discover the public's sentiment trends towards COVID-19 across different topics on Twitter over time.

Part III Conclusion.

Chapter 8. In this chapter, we conclude this thesis, the contributions of our works are summarized and some future research directions are also listed.

CHAPTER 2

Literature Review

The earliest research on the detection of spammers appeared in the field of email. In recent years, with the rapid popularity and development of the Internet, spammers have also flowed into e-commerce platforms and social networks, accompanied by a rapid development trend. In this chapter, we shall first conduct a comparative analysis and summary of spammer detection methods in the two mainstream fields of e-commerce platforms and social networks in the literature.

Furthermore, for rumour analysis, four steps of rumour classification [98] will be illustrated, i.e., rumour detection, rumour tracking, stance classification and veracity classification, followed by detailed approaches to rumour detection in the literature, as the rumour detection is the first and non-ignorable process in all four steps.

2.1 Spammer Detection

Spammers specifically refer to the Internet spam opinion generators who are driven by commercial interests to achieve improper purposes such as influencing Internet public opinion and disrupting the Internet environment. Spammers usually spread false opinions and spam on the Internet by manipulating software robots or navy accounts. Early spammers mainly intentionally lead mail recipients to a cooperative commercial website by sending a large number of spam emails, or released a large number of spam emails through software robots to achieve the purpose of widely spreading spam information [76]. In order to detect such spammers, research focuses on the content generated by spammers. Because in the early network environment, the number of spammers is relatively small and their behaviours are not highly concealed. The spam generated usually had significant identifiable characteristics, e.g., emails containing significant commercial advertising information and spam website information [65]. Researches on spammer detection based on content features mostly uses natural language processing methods in machine learning to study content data. And this kind of content-based methods mainly detect spammers from the aspect of text classification [73], text sentiment analysis [93], and text tendency analysis [52]. Email content analysis includes methods such as Bayesian classification, keyword-based classification, genetic algorithm classification, and neural network classification et al.. The detection based on the characteristics of the spam content can find spammers with a high accuracy rate.

However, as the network environment has become more complicated, e-commerce platforms and social networks have risen rapidly. Compared with traditional e-mail, the application scenarios of e-commerce platforms and social networks are more abundant, and the data structure is more complex. Meanwhile, the manifestations of spammers are also more diverse. For example, in the e-commerce platform, spammers try to influence consumers' decision by commenting on commodities; in the social network, spammers disturb the order of the social network by forwarding a large number of meaningless or fraudulent content. Spammers may even fake user models to become neighbours of normal users, and take advantages of collaborative filtering to generate recommendations based on neighbour preferences, which makes them have a high probability of being recommended to normal users. This special attack method is called "Shilling Attack". Wu et al. [85] developed characteristic indicators that distinguish between shilling attackers and normal users based on the characteristics of shilling attacks, and then summarized three types of shilling attacks detection methods from the perspective of classification in machine learning.

Although spammers have different manifestations, the detection methods are interlinked. The most mainstream detection methods are based on the assumption that the behaviour of spammers is abnormal. And first define high-discrimination behaviour characteristics, then map users as vectors in the feature space, and further build a classifier to determine the category of unknown users. The following of this section will make a detailed summary and comparison of the researches on the detection of spammers in e-commerce platforms and social networks.

2.1.1 Spammer Detection in E-commerce Platforms

The literature [23, 88, 97] pointed out that the quality of reviews has a great influence on the sales of products. If a product has a large number of users' praise, users will show a greater tendency to buy. Besides, various commercial organizations and individuals will also use the content generated by users in e-commerce to assist decision-making and implement user-related recommendations. Therefore, spammers in e-commerce platforms are mainly in the form of false commenters. Such kind of spammers post false comments to influence the comment trend of a certain product, and further influence the user's purchase decision, so that to bring corresponding business benefits to their employers or themselves. In view of this, a large number of literatures use reviews as the research object to define the characteristics of spammers in e-commerce platforms. The basic attributes of a review generally include rating, review time, and review content [35], all of which can be used to construct review features. Research has found that the purpose of spammers is to comment on the target product in large quantities and in a very short period of time, so their comment content is very short, usually just a few words, such as "very good", "good product", "like it" etc.. Based on this characteristic, Lim et al. utilized the number of words in the comment and the similarity of the comment content to construct the features of spammers [48]. Li et al. also analyzed the content of the comments. They found that normal users usually express their subjective thoughts when filling in the comments, and the comment content is mostly about the feeling of using the product. On the contrary, spammers will use strong interjections in the reviews to attract the attention of others, and the content of the reviews is not directly related to the product itself. So Li et al. [45] came up with the following characteristic attributes: the frequency of the first person in the review, the frequency of the subjective/objective word in the review, the frequency of the interjection in the review etc.. Based on the review time, a lot of evidence shows that early reviews greatly affect consumers' attitudes toward products, so spammers will fill in early reviews first. Mukherjee et al. used this feature to define early review indicators [56]. Specifically, assuming that r_u represents a review produced by user u for product p, the early review feature (Early Time Frame, ETF) can be described as Eq. (2.1).

$$ETF(r_u, p) = \begin{cases} 0, & if\ L_{(a,p)} - A_{(p)} > \delta \\ 1 - \frac{L_{(a,p)} - A_{(p)}}{\delta}, & otherwise \end{cases} \tag{2.1}$$

where $L_{(a,p)}$ is the time when review r_u published, $A_{(p)}$ is the date that product p released, and δ represents a threshold value. In [56], δ is assigned a value of 7 months to describe the data of the Amazon website.

In addition to basic attributes such as evaluation level, comment time, and comment content, the behaviour of commenters can also be used to define the characteristic attributes of spammers in e-commerce platforms. Jindal et al. pointed out that spammers usually apply templates and comment on the same content under different products [41]. Fei et al. found that the fake reviewers generally are the water soldiers purchased by e-commerce to give them good reviews or place negative reviews on competitors' products. Meanwhile, the shopping records of these fake reviewers are usually very low or even zero. Therefore, the purchase confirmation rate can be a very good feature for this type of spammers [30]. Literature [37, 56] use the suspicious degree of the reviewer as a recessive variable to construct a Bayesian recognition model. They analyzed the various behaviours of fake commenters in the e-commerce platform and found that fake commenters have very different behaviour distributions from normal users. Based on this finding, it is pointed out that most fake commenters on e-commerce platforms have the characteristics of burstiness of review (BST). Specifically, the accounts of fake commenters are generally registered temporarily in a relatively short period of time and will not be used for a long time. Taking Amazon website data as an example, the time interval between the latest comment of a spammer and the earliest comment is mostly less than 28 days. The feature BST can be described as Eq. (2.2).

$$BST(u) = \begin{cases} 0, & if\ L_{(u)} - F_{(u)} > \tau \\ 1 - \frac{L_{(u)} - F_{(u)}}{\tau}, & otherwise \end{cases} \tag{2.2}$$

where $L_{(u)} - F_{(u)}$ indicates the number of days between the earliest and most recent comments posted by user u, τ is a threshold value. τ is assigned a value of 28 days to describe the data of the Amazon website.

Table 2.1 summarizes some representative features for reviews on e-commerce platforms. From the above researches in the literature, we can see that in the e-commerce platform, the feature construction of spammers is mainly from the perspective of comments, using information based on content data such as time nodes, comment content, and reviewer information. At the same time, through the analysis of the behaviour of reviewers, we can capture the difference between the behaviour patterns of spammers and normal users, and then construct behaviour-based features for spammer detection in e-commerce platforms. There are similarities between social networks and e-commerce platforms. For example, in social networks such as Weibo and Facebook, users can post blogs, and can comment or reply under blogs published by other users. Therefore, some

Table 2.1: Representative features for reviews on e-commerce platforms

Type	Name	Description	Source
Text	LEN	Review length in words.	Li et al. [45]
	PP1	Ratio of the first-person pronouns ('I', 'my', etc.).	Li et al. [45]
	OW	Ratio of objective words.	Li et al. [45]
	RCW	Ratio of ALL-capital words.	Li et al. [45]
	RC	Ratio of capital words.	Li et al. [45]
	SW	Ratio of subjective words.	Li et al. [45]
	RES	Ratio of exclamation sentences.	Li et al. [45]
	ARL	Average review/tweet length in number of words.	Mukherjee et al. [56]
	ACS	Average content similarity among one's reviews.	Lim et al. [48]
	MCS	Maximum cosine similarity among all review pairs.	Mukherjee et al. [56]
Behaviour	Rank	Rank order among all the reviews of product.	Jindal et al. [41]
	DEV	Deviation between rating of the review and the average rating.	Mukherjee et al. [56]
	EXT	Extremity of rating: 1 for ratings $\{4,5\}$, 0 otherwise.	Rayana et al. [69]
	ETF	Early time frame: early reviews can increase impact (refer to Eq. (2.1)).	Mukherjee et al. [56]
	ISR	If review is user's sole review, then 1; otherwise 0.	Rayana et al. [69]
	MNR	Max number of reviews/tweets one day.	Mukherjee et al. [56]
	PR/NR	Ratio of positive/negative reviews.	Mukherjee et al. [56]
	BST	Burstiness of review/tweet (refer to Eq. (2.2)).	Mukherjee et al. [56]
	BRR	Ratio of the reviews in a burst pattern to the total reviews.	Fei et al. [30]
	ARD	Average rating deviation of user's reviews.	Lim et al. [48]
	WRD	Weighed rating deviation.	Lim et al. [48]
	AVP	Ratio of Amazon verified purchase.	Fei et al. [30]

feature construction methods based on comments in e-commerce platforms can also be used in social networks.

2.1.2 Spammer Detection in Social Networks

The significance of the existence of social networks lies in the rapid and widespread dissemination of instant information, providing users with a platform for expressing emotions and making friends at anytime and anywhere. Its functions are far from limited to posting blogs and making comments. Users can also perform a variety of interactive behaviours such as "add friend", "give a gift", and "interactive games". In addition, both the quantity of information and the rate of generation of information in social networks far exceed e-commerce platforms. Moreover, social networks generally have restrictions on the number of words in text content, and the content-based characteristics that can be extracted from them are very limited. Furthermore, for the protection of user privacy, some data based on user identity information is difficult to obtain. Therefore, there are still differences between the feature construction of spammers in social networks and e-commerce platforms.

Benevenuto et al. [11] first published statistical dataset of spammer behaviour in online video sharing sites. They collected the behavioural data of the online navy from the famous Internet video site YouTube for statistical analysis. These data confirmed the existence of a large number of spammers in social networks. At the same time, Benevenuto et al. use manual labeling methods to build training datasets, and then analyze the behaviour of the identified spammers and define their characteristics. Three feature selection algorithms in Weka are used to evaluate the discrimination of each spammer behaviour characteristics, and traditional supervised classification methods are used to detect whether an unknown user is a spammer or not. This method is a representative method of spammer detection in the field of social networks, that is, identifying spammer based on user behaviour characteristics. Subsequent researches on spammer detection are mostly based on this method, adding features or optimizing detection methods to increase the accuracy of spammer detection in social networks.

Parameswaran et al. [60] developed a theoretical modeling method to model spammers' behaviour, and they found that spammers' behaviour strategies are constantly changing. Therefore, they proposed that they can monitor the behaviour of the spammer for a long time, and establish a blacklist to reduce the harm of the spammer. Lin et al. [49] utilized Weibo as the background to conduct research on the feature construction of spammers in social networks. From the perspective of user behaviour analysis, they summarized the behaviour of spammers in Weibo into three categories: a large number of advertisements, repeated uncontrolled reposts, and large-scale attention. And based on these three behaviours, they gave the following behaviour indicators to detect spammers in Weibo: follow-fan ratio, friend-fan ratio, the ratio of repeated Weibo reposts, the average number of reposts of a single Weibo, and the highest number of reposts of a single Weibo.

On top of user behaviour analysis, a user-centric social circle will be gradually formed by users in social media via social interaction, these relationships between each user often contains rich information. On the other hand, spammers from social network do not form normal social relationships with others and the relational network of a spammer is different from a typical user. Hence, users' relational network can be used to detect spammers.

Based on this, Murmann et al. [58] utilized Twitter as a background to conduct research. They use neighbour nodes with direct interactions to detect the trust relationship between users, and obtained a new relation feature set. And then, they use this feature set to perform a suspicious degree ranking, the higher the suspicious degree ranking, the higher the possibility of being a spammer. Gayo-Avello et al. [52] proposed an assumption that spammers in Twitter would spend a lot of time to follow target users or wait for the target users to follow back. Based on this assumption, they proposed a topic ranking model. Krestel et al. [43] took advantages of the characteristic that spammers' suspicion would spread in social networks, and utilized the propagation on the graph to detect spammers in the tag sharing site. This method realizes the modeling of the relation structure between spammers, tags, and network resources in the tag sharing site. Firstly, the suspiciousness of some seed nodes is given, and then the suspiciousness of all nodes in the entire graph model is calculated according to the suspiciousness of seed nodes' propagation, so as to find spammer' nodes. A certain degree of community can be formed by groups of spammers in social network just like normal users, according to a study by Bhat et al. [12]. Therefore, overlapping community graphs were found by them after they extracted user interaction graphs from user behaviour logs. A part of spammer nods were marked manually before community relations between each unidentified nodes and maked nodes were calculated by them, consequently unknown nodes were classified. By using complex network features such as K-Core [8], *Triangle Count* [70], Connected Components [62], PageRank [59] and other topological features to construct the features of spammers on social networks, Fakhraei et al. [27] tried to construct a topological structure graph for each relation on the social network. An assumption was made by them that spammers take a very prominent position in each network topological graph.

In addition, there are many works that use abnormal nodes in graph to detect spammers in social networks, and have made great progress as well. For example, graph-based isolated edge detection [15], abnormal nodes detection in bipartite graph [6], abnormal nodes detection in semantic graph [16], outlier detection in attribute graph [64], and so on.

Moreover, based on the analysis of social network relations, sequence-based features are proposed to reduce graph analytic methods limitation. Because the graph-based methods mentioned above are only effective when assuming that data is homogeneous, i.e., different types of relations are required to be modeled separately. Unfortunately, the interactions among different types of relations are ignored under this assumption. The graph analytic method limitations are somehow reduced by sequence-based methods, as all relations are modeled together. To be more specific, sequence-based features are extracted by converting different types of relations into a user-wise sequence, and each sequence length is dependent upon the user. Then the sequence of each user obtained previously then gets fed into a feature extraction function, this is to obtain a feature vector from sequence of user. For instance, a clickstream model proposed by Wang et al. [79] which determines the distance between each clickstream traces(i.e., users' sequences of click events). The assumption was made that there are different click transition patterns made by normal users and spammers, and they put their effort on different activities. Considering the activity sequence of users by measuring the frequency of each length k sub-sequence for every user, Fakhraei et al. proposed *Sequential k-gram Features* [27]. Nevertheless, due to the reason that the computing capacity cost is large, Fakhraei et al. only carried $k = 2$. Afterwards, to overcome the small k limitation in k-gram models, we can use *Mixture of Markov Models* by picking out a small subset of vital sequence from a long sequence chains which announced by Peng et al. [63]. However, only short-term information within users' relational sequences is considered in *Mixture of Markov Models*.

Social networks truly reflects the social circle of people in the real world. Both normal users and spammers form a certain social network structure in social networks. Therefore, using relational network characteristics to detect spammers in social networks has a good development prospect. However, through the analysis of the above researches, we find that the existing research methods for constructing the characteristics of spammers in social networks ignore the latent factors between relations, which gives experienced spammers an opportunity. Therefore, in this thesis, we focus on making up for the deficiencies in the existing research, and dig deeper into the hidden information behind the relation-relation, relation-user and user-user in multi-relational social networks, so as to improve the performance of spammer detection in social networks.

2.2 Rumour Classification Process

Rumour is another kind of misbehaviours in social network. In the literature, the approaches of rumor recognition is a little bit different from traditional spammer detection. Generally, the process of rumour classification can have slight variations depending on the specific scenarios. Zubiaga et al. [98] summarized a typical process for rumour classification, which takes most of the applications into consideration. As pointed out in the descriptions below, depending on requirements, some of these components can be omitted.

Identifying a none-confirmed piece of information usually is the first step on rumour classification process (i.e., rumour detection), and the last step is to calculate a veracity estimation score of that information(i.e., veracity classification). Fig. 2.1 shows the entire process from rumour detection to veracity classification going through the four steps.

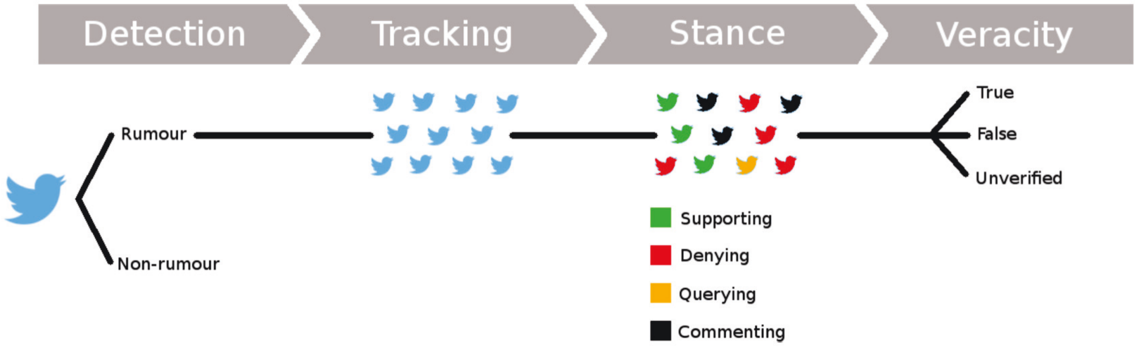


Figure 2.1: Processes of rumour classification

Rumour Detection: In the first stage of rumour classification, it starts with the identification of whether a piece of information makes up to a rumour. A stream of media posts are usually a typical input to a rumour detection component, then it goes through the binary classifier to determine if each post is considered a rumour or non-rumour. Another stream of posts output from this component with each post labeled as non-rumour or rummer. This component is great with identifying with emerging rumours, however, for dealing with rumours that are known a prior, this component is not necessary.

Rumour Tracking: When the rumour is identified, posts discussing the rumour then gets collected and filtered for either rumour that is detected by the detection component or known to prior. This component is looking for posts discussing the rumour while irrelevant posts are eliminated. The input are rumour posts which can be a set of keywords, a posts or sentence describing the rumour. A collection of posts discussing the rumour are output from this component.

Stance Classification: While posts related to a rumour are retrieved by the rumour tracking component, how each post is orienting to the rumour's veracity is determined by the stance classification component. Having a set of posts associated with the same rumour as input, it outputs a label for each of those posts, where the labels are chosen from a generally predefined set of types of stances. This component can be useful to facilitate the task of the subsequent component dealing with veracity classification. However, it can be omitted where the stance of the public is not considered useful, e.g., cases solely relying on input from experts or validation from authoritative sources.

Veracity Classification: The final, veracity classification component attempts to determine the actual truth value of the rumour. It can use as input the set of posts collected in the rumour tracking component, as well as the stance labels produced in the stance classification component. It can optionally try to collect additional data from other sources such as news media, or other websites and databases. The output of the component can be just the predicted truth value, but it can also include context such as URLs or other data sources that help the end user assess the reliability of the classifier by double checking with relevant sources.

2.3 Rumour Detection

Like the studies by Hamidian and Diab [33, 34], most rumour detection studies train classifiers from a pre-labelled set of rumours. For example, if a pre-labelled rumour consists of something like "sprouted potatoes are non-toxic and edible". It would be classified as a rumour if any news related to it, for example, "it is safe to remove the sprouts of sprouted potatoes before eating them," is classified as a rumour. Such rumours are categorised as long-standing rumours. These long-standing rumours are usually circulated on social media for a long time, and there are a number of relevant rumours known as prior [98].

Nevertheless, in the case of new sudden emergencies for which there is no a prior information, relying solely on the similarity of new rumours to a prior content is not sufficient. To address this problem, Zhao et al. [94] assumes that rumours trigger tweets from users who doubt or inquire about their authenticity. To put it differently, if a message has many related query tweets, it means that the tweet is a rumour. A manually curated list of five regular expressions was created to identify query tweets by Zhao et al. These query tweets were then clustered by similarity, and the tweets in each cluster were eventually considered as candidate rumours. Moreover, Zubiaga et al [99, 100] suggest an alternative method to learn the context of an entire groundbreaking news story in order to be able to estimate whether the tweet will become a rumour or not. The approach proposed by Zubiaga et al. is based on the assumption that without full knowledge of the context, we may not fully understand the truth of the underlying story underneath the tweet. McCreadie et al. investigated the feasibility of using a crowdsourcing platform to identify rumours and non-rumours on social media from a different perspective. This rumour identification gained a high level of consensus among annotators [55].

While research on rumour detection is all over the place, there is very little research on the impact of rumours. Some social media outlets even employ senior journalists who work 24 hours a day, 7 days a week to maintain the public website [82]. They regularly expose new rumours to limit the possible negative influences of rumours on their platforms (e.g. @*WeiboPiyao* in Sina Weibo). As far as we know, most of the existing work on the impact of rumours is based entirely on a prior knowledge or other kinds of assumptions, even human ones. Therefore, a targeted statistical description of the impact of rumours in social networks, and thus helping governments to control social rumours, is now an urgent priority.

2.4 Summary

This chapter first explains the traditional spammer detection methods, and then summarizes and compares the detection methods of spammers in e-commerce platforms and social networks. When detecting spammers in social networks, there is very little content-based data that can be used, so we need to consider from the perspective of relational networks. Therefore, in Chapter 4 and Chapter 5, we shall propose two detection frameworks which are content-independent but relation-dependent for spammer detection in social networks. Furthermore, in the second half of this chapter, we analyze the literatures on the propagation and detection of rumours, which trigger the work of the impact of rumours in the second part of this thesis.

Part I

Part I Spammer Detection

Chapter 3

Data Description and User Behaviour Analysis

Spammer detection on social networks is much more different than spammer detection in other platforms, i.e., e-mail, e-commerce sites. Because the text information is much less than other platforms and is relatively short. Furthermore, even compact textual information is difficult to collect for reasons of protecting user privacy. On the contrary, the types of relations in social networks are more complex than other platforms. Therefore, in the first part of our research work, we call for relation-dependent but content-independent methods, i.e., user behaviour analysis, graph-embedding method, sequence-based method, for spammer detection on social networks.

In the first part of our research work (spammer detection), all of our researches are based on a real-world dataset from *Tagged.com*. Fakhraei et al. released part of the dataset from *Tagged.com* when the paper [27] was published on the top international conference SIGKDD-2015. Because of user privacy, the release of real-word social network data is very limited. Datasets for multi-relational social networks are even rarer. Besides, the dataset released by Fakhraei et al. has already separated the data by relation. Therefore, it is really a good opportunity for us to use this dataset as our experimental data for multi-relational social networks spammer detection research. In this chapter, we first introduce the source and scale of the multi-relational *Tagged.com* dataset. Since each relation is given by a numerical name (i.e. 1, 2, 3, $\cdots$) instead of their corresponding semantic meanings, we then use data statistical analysis techniques to infer the actual semantic relation type names corresponding to each relation. At the last of this chapter, we utilize the actual semantic meaning of each relation to analyze the different behaviours between spammers and normal users. Afterwards, we give the definitions of a series of behavioural characteristic indicators to detect spammers in relational social networks.

3.1 Data Source and Scale

The dataset used in our spammer detection research work is from *Tagged.com*. *Tagged.com* is a social website for people to meet new friend and socialize with each other, and was established in 2004. In *Tagged.com*, users are allowed to find friends, play games, share personalized tags, give virtual gifts, and chat in real time. Fakhraei et al. released part of the dataset from *Tagged.com* when they published their paper [27] on the top international conference SIGKDD-2015[1]. This dataset contains $5,607,447$ users' implicit identity information and a total of $858,247,099$ interaction records in 7 relations between users within 10 days.

The dataset mainly contains two parts of information: (1) Users' implicit identity information, including user ID, gender, age group, and user's label given by domain experts that characterizes normal user or spammer. Detailed description is illustrated in Table 3.1. The example given in Table 3.1 is taken from a real sample. It shows that the user "00000092" is a male in his twenties and he has been marked as a spammer by Tagged.com. (2) Users' interaction records. Table 3.2 lists the detailed attribute information. The example given in Table 3.2 means that user "00000092" sent a message to user "03753402" at 02:00 on day "0" (Later, it will be introduced that relation "4" is presumed to be sending "Message").

3.2 Social Relations Speculation

Tagged.com is a multi-relational social network, including 7 types of relations, i.e. *Add Friend*, *Give a Gift*, *Message*, *Pet Game*, *Meet-Me Game*, *View Profile*, and *Report Abuse*. But, the released dataset only gives each relation a numerical name (i.e. 1, 2, 3, $\cdots$, 7) instead of their corresponding semantic meanings. Therefore, the existing research work [13, 27] that uses this dataset as the research object fails to take into account the differences in behaviour characteristics brought about by relation types. But in fact, the

[1] Data source: `https://linqs-data.soe.ucsc.edu/public/social_spammer/`

Table 3.1: Attribute description of *Tagged.com* user identity information dataset

Attribute	Attribute Description	Example
userId	the unique identity label for each user	“00000092”
sex	gender selected when the user registered	“M”
ageGroup	user’s true age group (“1”: 10-19; “2”: 20-29; ···)	“1”
label	user’s label given by domain experts (“0” represents normal user; “1” represents spammer)	“1”

Table 3.2: Attribute description of *Tagged.com* users’ interaction records dataset

Attribute	Attribute Description	Example
day	Date that record occurred (implicitly expressed as 0-9, 10 days in total)	“0”
time_ms	specific time that record occurred (milliseconds)	“7200000”
src	userId of the originator of the record	“00000092”
dest	userId of the recipient of the record	“03753402”
relation	interaction type between users (e.g. *Message*, *Give a gift*, etc.. implicitly expressed as 1-7, 7 relations in total)	“4”

difference between specific behaviour patterns of spammers and normal users is often directly related to the actual meaning of the relation. For example, spammers will focus more on the relation *Message* in order to spread rumours and fraudulent information, and rarely pay attention to the relations such as *Give a Gift* or *Add Friend*. On the contrary, when normal users use *Tagged.com*, a social network for making friends, for a wider range of friends with similar interests, relations such as *Add Friend* and *Give a Gift* will occur many times. Therefore, it is necessary to speculate the corresponding semantic relation type name to each relation through the analysis of the data. So that, we can further dig out the deep semantics in the data, and then define new behavioural indicators that characterize the spammer behaviour pattern according to the relation.

Table 3.3: Statistics of 7 relations within 1 day

Relations	#Nodes	#Edges	#Connected Components
relation 1	417,273	605,147	18,049
relation 2	424,078	969,233	37,345
relation 3	2,443,123	14,531,699	10,640
relation 4	1,925,333	21,039,127	16,140
relation 5	3,506,363	14,828,212	136
relation 6	3,492,863	33,481,799	85
relation 7	***15,184***	***12,518***	***4,973***

We convert the interaction records as shown in Table 3.2 into a directed graph with weight, where each directed edge in the graph represents a *relation* performed by *src* user towards *dest* user. While the weight of the edge represents the specific time that the *relation* been performed. According to the statistics of the number of nodes, the number of edges, and the number of weakly connected components constructed by the interaction records dataset, we find that the statistics of each relation are close to evenly distributed in 10 days. In other words, the attribute *day* has a very small influence on the statistics. In order to simply and clearly show our inference for each relation, the rest of this section takes the statistics of each relation within 1 day as an example to illustrate. Table 3.3 lists the number of nodes, the number of edges, and the number of weakly connected components of each relation on *day* 0.

Tagged.com officially released the following 8 types of relations:

- ***Message***: Users can send messages to any other user, including strangers.
- ***Add Friend***: Users can send adding friend request to any other user that they like.
- ***View Profile***: Users can view any other user's profile, and can view/edit their own profile.
- ***Give a Gift***: Users can give a gift to any other users that they like. However it will cost their virtual coins and there is a limit on the number of gifts given per day.
- ***Send Wink***: Users can send wink to any other users that they like in order to get their attention. Notice, wink can only be sent to someone once in a day.

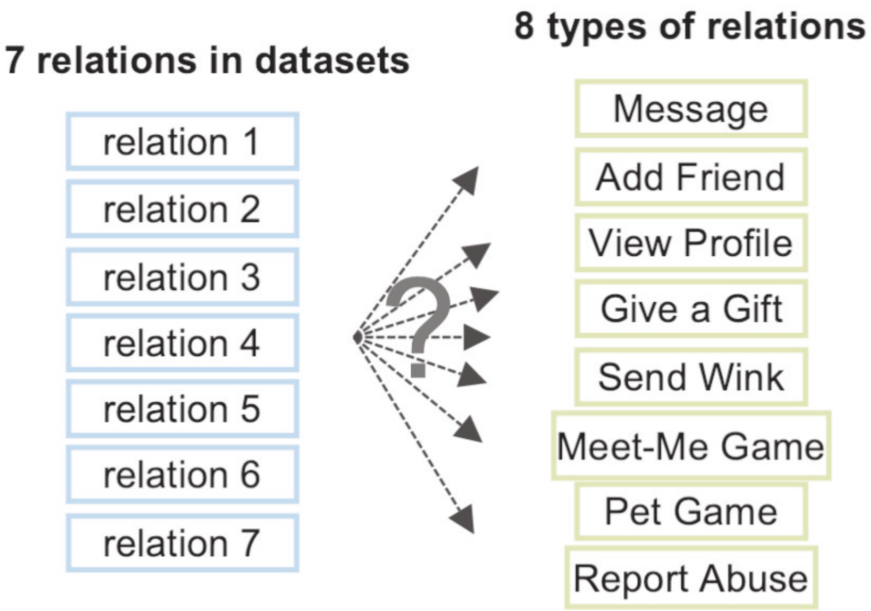


Figure 3.1: Relations speculation.

- ***Meet-Me Game***: Main project in *Tagged.com*. Users can browse other users profile photo one by one to see if they want to know more about them.
- ***Pet Game***: Each user have a virtual value. And users can use their virtual coins to buy any other user as their pet. Buying pets will have a chance to get cash rewards, and increase pet assets will have a chance to be on the star list.
- ***Report Abuse***: Users can report any other user for their abnormal behaviour (i.e., copyright infringement, threat of violence, identity theft or theft of personal information, etc.).

Nevertheless, the dataset released by Fakhraei et al. [27] only consists of 7 types of relations. Therefore, we first need to infer a type of relation that is not included in the dataset, and then use the elimination method to speculate the remaining 7 types of relations, as shown in Fig. 3.1. According to the hint given by *Tagged.com* website that "wink" can only be sent to one user once a day. Thus, we further counted the total number of edges, the number of duplicate edges and the number of self-loop edges within 1 day, as shown in Table 3.4. Obviously, relation 1-7 all have duplicate edges within 1 day, thus, we give the inference that the relation type "*Send Wink*" is not included in the 7 relations. In addition, it is not difficult to find from Table 3.4 that relation 3 is the only relation that has self-loop edges. Among the remaining 7 types of relations, it is

Table 3.4: Statistics of edges of 7 relations within 1 day

Relations	Total #Edges	#Duplicate Edges	#Self-loop Edges
relation 1	605,147	342	0
relation 2	969,233	249,958	0
relation 3	14,531,699	5,139,965	***1,102,132***
relation 4	21,039,127	13,424,835	0
relation 5	14,828,212	469,330	0
relation 6	33,481,799	1,250,771	0
relation 7	12,518	1,802	0

confirmed that only *View Profile* will have self-loop edge. In fact, on many social networks (e.g. WeChat), users tend to frequently browse or modify their own profile, so we give the inference: relation 3 is "*View Profile*". According to Table 3.3, the number of edges and nodes contained in relation 7 is much more smaller than the number of edges and nodes in other relations. However, the number of weakly connected components in relation 7 is relatively large, indicating that the probability of relation 7 happened between users is very low and the user groups in relation 7 are more scattered. Therefore, we give the inference: relation 7 is "*Report Abuse*". Moreover, in the original interaction dataset, we find that the records of relation 4 are mostly multiple exchanges between the same two users over a period of time, in another word, relation 4 is a frequent interaction between users. Utilizing the method of elimination, only sending *Message* among the remaining 5 relations meets this characteristic. Thus, we give the inference: relation 4 is sending "*Message*".

After the above analysis, we are left with relation 1, relation 2, relation 5 and relation 6. In *Tagged.com*, two games are their main projects, i.e., *Meet-Me Game* and *Pet Game*. So the number of occurrences, that is, the number of edges of these two relations will be significantly higher than others'. We infer that relation 5 and relation 6 are one of *Meet-Me Game* and *Pet Game*, while, relation 1 and relation 2 are one of *Give a Gift* and *Add Friend*.

For further analysis, we count the number of spammers and normal users within *src* users, as shown in Table 3.5, and the number of interaction records of them separately, as shown in Tabel 3.6.

As shown in the homepage of *Tagged.com*, *Give a Gift* will cost user's virtual coins and there is upper limit on the number of gifts given per day. So the number of edges, especially the number of duplicate edges of relation *Give a Gift* will not be many. Hence

Table 3.5: Statistics of the number of *src* user's identity (spammer / normal user)

Relations	#Spammers	#Normal Users	#Users	Ratio of Spammers
relation 1	8,623	125,300	133,923	6.44%
relation 2	10,046	240,576	250,622	***4.01%***
relation 3	62,032	1,202,934	1,264,966	4.90%
relation 4	89,741	1,177,602	1,267,343	7.08%
relation 5	47,265	878,957	926,222	***5.10%***
relation 6	45,629	971,937	1,017,566	4.48%
relation 7	587	6,023	6,610	8.88%

Table 3.6: Statistics of the number of *src* user's (spammer / normal user) interaction records

Relations	#Spammers'	#Normal Users'	#Users'	Ratio of Spammers'
relation 1	66,542	538,605	605,147	11.00%
relation 2	51,412	917,821	969,233	***5.30%***
relation 3	1,957,732	12,573,967	14,531,699	13.47%
relation 4	3,405,436	17,633,691	21,039,127	16.19%
relation 5	4,751,014	10,077,198	14,828,212	***32.04%***
relation 6	1,973,856	31,507,943	33,481,799	5.90%
relation 7	1,366	11,152	12,518	10.91%

we infer relation 1 might be *Give a Gift*, then relation 2 will be *Add Friend*. Refer to Table 3.5 and Table 3.6, the number of interaction records and the ratio of the number of spammers in relation 2 are the lowest, which is also in line with the "*Add Friend*" speculation. Because the main purpose of spammers is to spread rumour, fraudulent information on social networks using the interactions with normal users. While in *Tagged.com*, any other types of relations can be performed as usual even without *Add Friend*. Therefore, we give the inference that relation 1 is "*Give a Gift*" and relation 2 is "*Add Friend*". In addition, from Table 3.5 and Table 3.6, we can find that the ratio of spammers' interaction records on relation 5 is the highest (32.4%), while the ratio of spammers on relation 5 takes only 5.10%. That is to say, spammers are more likely to perform relation 5. We can also get the introduction from the homepage of *Tagged.com* that buying more pets will have a chance to get cash rewards and increase pet assets will have a chance to be on the star list, which will achieve more attention. Hence, we give the inference that relation 5 is "*Pet Game*" and relation 6 is "*Meet-Me Game*". We summarize the above speculation as shown in Fig. 3.2.

Summarizing the above analysis, Table 3.7 shows the correspondence between the

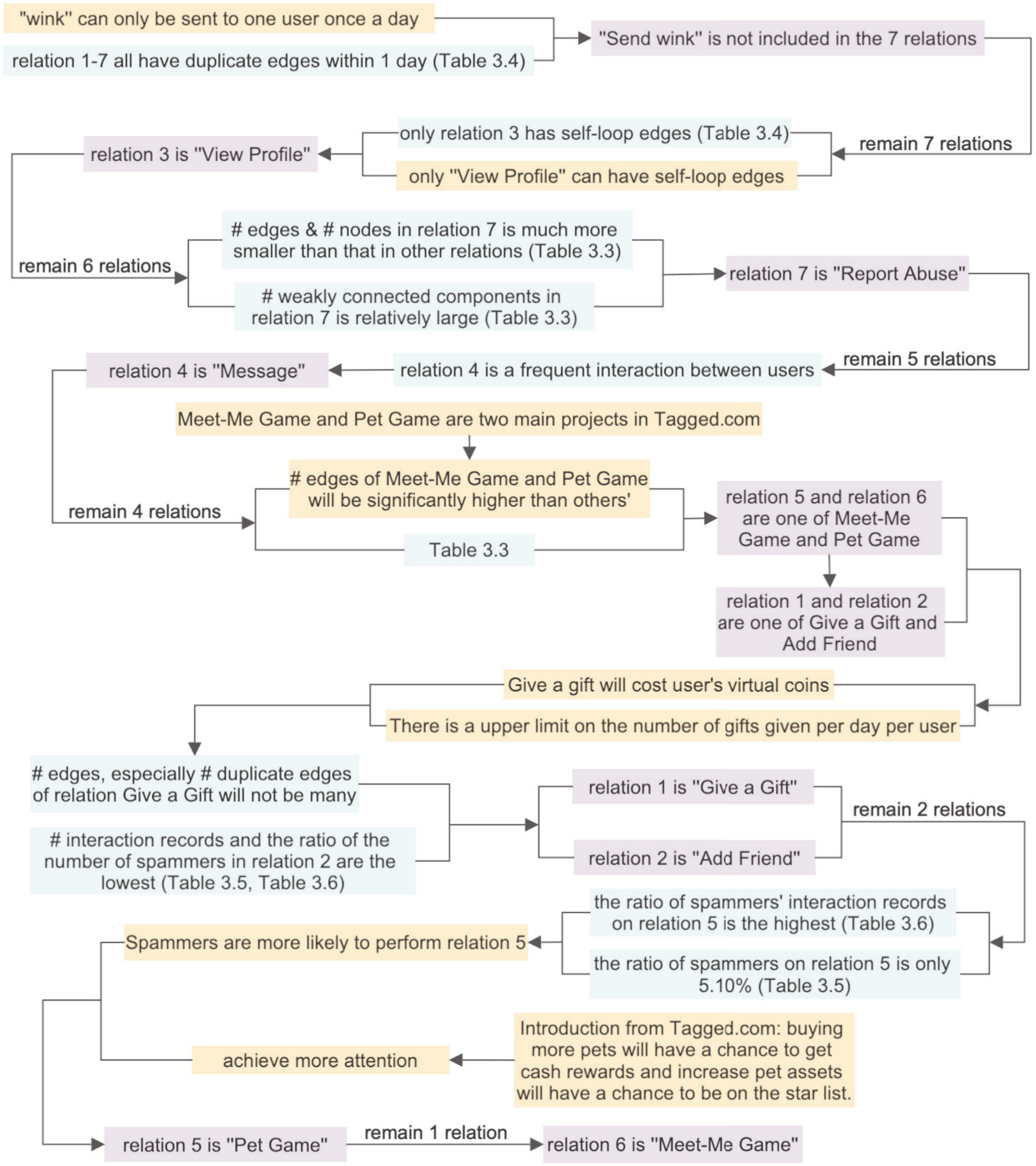

"wink" can only be sent to one user once a day
relation 1-7 all have duplicate edges within 1 day (Table 3.4)
"Send wink" is not included in the 7 relations
remain 7 relations
only relation 3 has self-loop edges (Table 3.4)
only "View Profile" can have self-loop edges
relation 3 is "View Profile"
remain 6 relations
# edges & # nodes in relation 7 is much more smaller than that in other relations (Table 3.3)
# weakly connected components in relation 7 is relatively large (Table 3.3)
relation 7 is "Report Abuse"
remain 5 relations
relation 4 is a frequent interaction between users
relation 4 is "Message"
Meet-Me Game and Pet Game are two main projects in Tagged.com
remain 4 relations
# edges of Meet-Me Game and Pet Game will be significantly higher than others'
Table 3.3
relation 5 and relation 6 are one of Meet-Me Game and Pet Game
relation 1 and relation 2 are one of Give a Gift and Add Friend
Give a gift will cost user's virtual coins
There is a upper limit on the number of gifts given per day per user
# edges, especially # duplicate edges of relation Give a Gift will not be many
# interaction records and the ratio of the number of spammers in relation 2 are the lowest (Table 3.5, Table 3.6)
relation 1 is "Give a Gift"
relation 2 is "Add Friend"
remain 2 relations
the ratio of spammers' interaction records on relation 5 is the highest (Table 3.6)
the ratio of spammers on relation 5 is only 5.10% (Table 3.5)
Spammers are more likely to perform relation 5
Introduction from Tagged.com: buying more pets will have a chance to get cash rewards and increase pet assets will have a chance to be on the star list.
achieve more attention
relation 5 is "Pet Game"
remain 1 relation
relation 6 is "Meet-Me Game"

Figure 3.2: Illustration of relations speculation using elimination.

seven relations and the actual relation names. Where relation 1 is “*Give a Gift*”, relation 2 is “*Add Friend*”, relation 3 is “*View Profile*”, relation 4 is “*Message*”, relation 5 is “*Pet Game*”, relation 6 is “*Meet-Me Game*” and relation 7 is “*Report Abuse*”.

Table 3.7: Relation ID and corresponding semantic relation type name

Relations	Relation Name
relation 1	Give a Gift
relation 2	Add Friend
relation 3	View Profile
relation 4	Message
relation 5	Pet Game
relation 6	Meet-Me Game
relation 7	Report Abuse

3.3 User Behaviour Analysis

The characteristics of users on social networks often depend on their unique behaviours. For example, a user who is active in a social network platform such as *Tagged.com* for the purpose of dating, then he/she may send friend requests to users with the opposite gender; a user who likes to dive to see photos, his/her friends might be very few, but the number of times he/she view other users' profiles would be quite considerable. It can be found that the user's purpose and tendency to use social networks determine his/her behaviour, and his/her behaviour will be displayed in the form of various behaviour characteristics. This phenomenon can be found in normal users, and it is even more obvious in spammers. Because comparing with normal users, spammers participate in social network activities with a stronger but single purpose. Spammers rarely do things that have nothing to do with their purpose, which results in their behaviour characteristics more obvious than normal users.

Along this line, in the following section, we first analyze the behaviour patterns of spammers in different relations in *Tagged.com* dataset, and derive assumptions about the differences between spammers and normal users in multi-relational social networks. Then, according on the hypothesis, we construct the spammers' behaviour features based on the multi-relational social network data. Finally, we use the *label* in *Tagged.com* user identity information dataset, which is given by domain experts to verify the effectiveness of our proposed features.

3.3.1 Misbehaviour Hypothesis and Feature Construction

Firstly, we keep an eye on relation sending *Message*, which is noted as relation 4 in *Tagged.com* dataset. Generally, when users perform relation *Message*, normal users will communicate with one or several users many times in a relatively concentrated time period. This kind of interaction has two characteristics: there are not too many people interacting, and the interaction is more frequent in a concentrated time period. On the contrary, some spammers only perform relation *Message* to distribute spam advertisements. This kind of spammers usually use machine-assisted methods to send messages to a large number of different users in a very short time, and will not contact the same user repeatedly in a short time period. Meanwhile, since most of the messages sent by spammers are spams, the proportion of replies received is much lower than that of normal users. Furthermore, we also find that the time period for spammers to send messages is different from normal users. Normal users usually browse social networks in free time such as commuting, lunch break, and evening, while spammers will send large amounts of messages during abnormal time periods such as midnight to six in the morning. Based on the analysis of these behaviours in relation *Message*, we have hypothesis 1:

Hypothesis 1, H1 Spammers usually use machine-assisted methods to send spams to a large number of users in a very short period of time, and they will not contact the same user repeatedly in a short period of time. On the contrary, the interactions between normal users will be conducted in a more concentrated time period, so the time difference for a spammer to send message to a specific user is longer than the time difference for a normal user to send message to a specific user (**H1a**). Meanwhile, the proportion of replies received after the messages being sent is lower than that of normal users as the spammers mostly send spams (**H1b**). In addition, compared with normal users, spammers have a higher probability of sending messages during abnormal time periods (from midnight to six in the morning) (**H1c**).

Based on hypothesis 1, we proposed the behavioural features as shown in Table 3.8. Specifically, for feature FOTD, let $\{T_a^0, T_a^1, \cdots, T_a^{L_a}\}$ be the time sequence sending from user u_a to another user. The c^{th} first-order difference value of user u_a can be represented as Eq. (3.1):

$$\Delta T_c(a) = T_a^c - T_a^{c-1} \; (1 \leq c \leq L_a) \tag{3.1}$$

where L_a is the total number of messages that user u_a sent within 1 day. Based on the

Table 3.8: Behavioural features based on Hypothesis 1

Feature Name	Feature Description	Source
FOTD	First-order time difference	H1a
RSR	Ratio of sending and receiving	H1b
RU	Ratio of user	H1b
RSST	Ratio of sending within spamming time	H1c

first-order time difference of user u_a, we further select the sequence with the first-order time difference value less than 1 hour. We think the user is offline when his/her first-order time difference is larger than 1 hour. We then ascendingly sort the new first-order time difference series of user u_a, noted as $\{\Delta T_a^0, \Delta T_a^1, \cdots, \Delta T_a^{L_a'}\}$. Eq. (3.2) gives the definition of cumulative distribution probability of user u_a:

$$f_{FOTD}(a) = \sum_{c=1}^{L_a'} \frac{|\Delta T_a^{c'}|}{L_a'} \frac{\Delta T_a^{c'}}{\Delta T_a^{L_a'}} \tag{3.2}$$

where the larger the value of $f_{FOTD}(a)$ indicates the larger the time difference between sending messages when user u_a contacts a certain user. According to hypothesis H1a, user u_a is more likely to be a spammer.

Eq. (3.3) gives the definition of feature RSR:

$$r_{RSR}(a) = \frac{L_a}{H_a + L_a} \tag{3.3}$$

where H_a represents the total number of messages that user u_a received within 1 day, while L_a is the total number of messages that user u_a sent within 1 day. It means that the number of messages sent by user u_a far exceeds the number of messages received when the value of $r_{RSR}(a)$ is more close to 1. And according to hypothesis H1b, user u_a is more likely to be a spammer.

Considering some users are used to disassembling a sentence into several pieces of messages to send, and some users just prefer to listening than speaking, so for hypothesis H1b, we proposed the feature RU as well:

$$r_{RU}(a) = \frac{U_{L_a}}{U_{H_a} + U_{L_a}} \tag{3.4}$$

where U_{L_a} is the total number of users that user u_a sent messages to within 1 day, and U_{H_a} is the number of users that replied user u_a's messages. The closer the value of $r_{RU}(a)$ is to 1 means that the user u_a is more actively sending messages than receiving

Table 3.9: Behavioural features based on Hypothesis 2

Feature Name	Feature Description	Source
EXT	User's extreme activeness over all relations	H2
RA	Ratio of accepting friend request	H2

messages from other users. According to hypothesis H1b, user u_a is more likely to be a spammer.

We use Eq. (3.5) to formulate feature RSST:

$$f(a) = \frac{|m \in S_a : m\ is\ sent\ from\ midnight\ to\ six\ in\ the\ morning|}{|L_a|} \tag{3.5}$$

where L_a is the total number of messages that user u_a sent within 1 day. The larger the value of $f(a)$ means that the more likely user u_a tends to send messages during abnormal periods of time. According to hypothesis H1c, user u_a is more likely to be a spammer.

In addition to relation *Message*, relation *Pet Game* is a major project of *Tagged.com* social network. We then have a deep analysis on the relation *Pet Game*, which is noted as relation 5 in the dataset. We find that spammers take advantage of this convenient condition of *Tagged.com* website to excessively participate in pet games. They purchase pets in large quantities using virtual coins and then sell them, and continue to accumulate assets to gain the chance to be on the star list, thereby improving their exposure and credibility. The purpose of these kind of spammers trying their best to get on the star list is to attract more ignorant new users to actively contact them. And then achieve their purpose of fraud and information deception. So they are usually the *dest* user of the relation "Add Friend ". Based on the analysis of these behaviours in relation *Pet Game*, we have hypothesis 2:

Hypothesis 2, H2 The distribution of records of normal users on multi-relational social networks is relatively even. On the contrary, spammers will be extremely active on relation *Pet Game*. Meanwhile, such kind of spammers are usually the *dest* user of the relation "Add Friend ".

Based on hypothesis 2, we proposed the behavioural features as shown in Table 3.9. To be more specific, let $\mathscr{U} = \{u_1, \cdots, u_n\}$ be the set of n users who are connected by m kinds of relations denoted as $\mathscr{R} = \{r_1, \cdots, r_m\}$. For $\forall r_i \in \mathscr{R}$, we use $\{C_{u_1}^{r_i}, C_{u_2}^{r_i}, \cdots, C_{u_n}^{r_i},\}$ to represent the number of each user being the *src* user on relation r_i. Then, user u_a's activeness on relation r_i can be denoted as Eq. (3.6):

$$R^{r_i}_{EXT}(a) = \frac{|C^{r_i}_{u_a}|}{|\overline{C^{r_i}}|} \tag{3.6}$$

where $\overline{C^{r_i}}$ is the average value of all the items in $\{C^{r_i}_{u_1}, C^{r_i}_{u_2}, \cdots, C^{r_i}_{u_n},\}$. The larger the value of $R^{r_i}_{EXT}(a)$ indicates that the more actively user u_a performed on relation r_i. Then, we can get a sequence of user u_a's activeness on every relations, denoted as $\{R^{r_1}_{EXT}(a), R^{r_2}_{EXT}(a), \cdots, R^{r_m}_{EXT}(a)\}$. We use the variance to represent the feature EXT, as shown in Eq. (3.7):

$$\sigma^2_{EXT}(a) = \frac{\sum_{1\le i\le m}(R^{r_i}_{EXT}(a) - \overline{R})^2}{m} \tag{3.7}$$

where $\overline{R}$ is the average value of all the items in $\{R^{r_1}_{EXT}(a), R^{r_2}_{EXT}(a), \cdots, R^{r_m}_{EXT}(a)\}$. The larger the value of $\sigma^2_{EXT}(a)$ indicates the more unstable of user u_a performed on every relations. According to hypothesis 2, user u_a is more likely to be a spammer.

As for feature RA, we formulated as Eq. (3.8):

$$r_{RA}(a) = \frac{X_a}{X_a + Y_a} \tag{3.8}$$

where X_a is the number of user u_a being the *src* user on relation *Add Friend*, Y_a is the number of user u_a being the *dest* user on relation *Add Friend*. The closer the value of $r_{RA}(a)$ is to 1 means that the user u_a is more likely to be the *dest* user on relation *Add Friend*. According to hypothesis 2, user u_a is more likely to be a spammer. It should be noted that the feature RA is not applicable to all social networks. Because *Tagged.com* social network has a major characteristic: users can also perform any other relations without performing relation *Add Friend*, including *Message* and *Report Abuse*. So we find that spammers in *Tagged.com* will not spend time on relation *Add Friend*, while normal users with the purpose of making friends will leave more records on the relation *Add Friend*.

3.3.2 Effectiveness Analysis

In order to verify the effectiveness of the features proposed in Table 3.8 and Table 3.9, we take the data of "*day* 0" from *Tagged.com* as a sample. Then, we calculate the value of each feature, and draw the cumulative distribution function curve (CDF curve) [22] of each feature for spammers and normal users separately. CDF curve can quantitatively display the distribution of the data. Each CDF curve represents the data distribution of the sample on a certain feature. Specifically, each point on the CDF curve corresponds to a feature value and the percentage of the number of samples whose feature value is less than this point to the total number of samples. Using the CDF curve, we can easily find the difference in the data distribution of a feature regarding spammers and normal users. And this kind of difference is exactly the discrimination of the feature we are looking for.

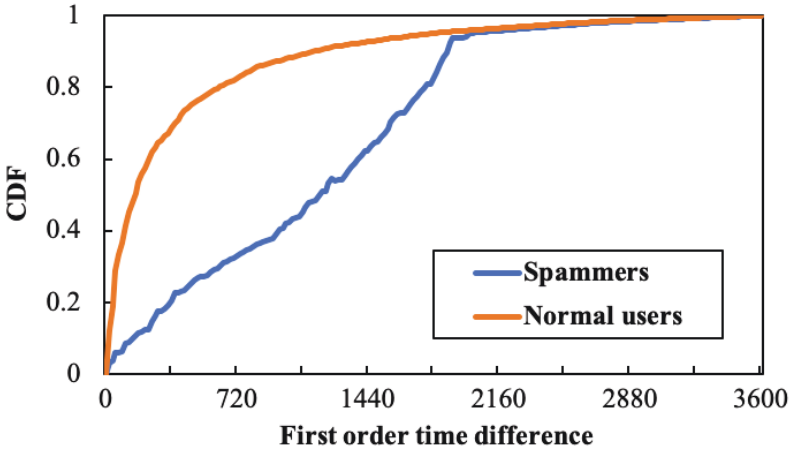
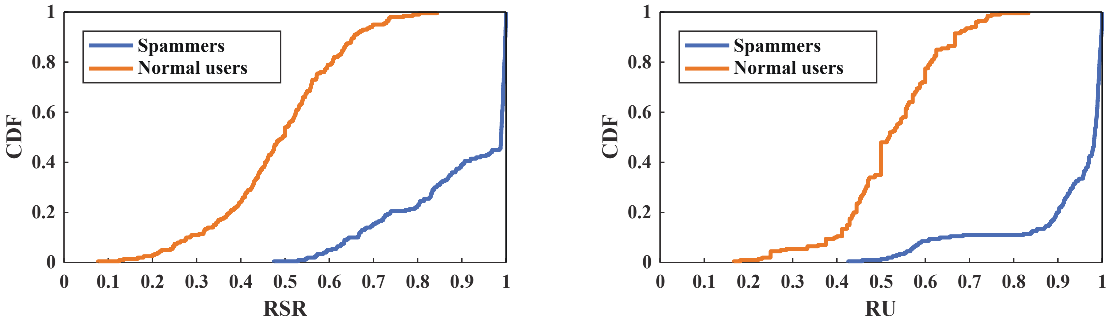


Figure 3.3: CDF curve of feature FOTD.

Fig. 3.3 shows the CDF curve of the proposed feature FOTD, which is based on hypothesis H1a. The abscissa is the first-order time difference value (in seconds) of relation *Message*. It can be clearly seen from the figure that nearly 80% of normal users will send the second message within ten minutes after sending a message to a user, which means that normal users will be relatively concentrated in a period of time when chatting with other users. However, the probability of a spammer sending the second message to the same user within half an hour is relatively low. However, there are still some spammers who will send the second message in a short time. The purpose of these kind of spammers might be achieving fraud or other purposes through chatting instead of just sending spam advertisements.

Fig. 3.4 shows the CDF curve of the proposed feature RSR and RU. The abscissas are the values of feature RSR and RU, which are calculated based on Eq. (3.3) and Eq. (3.4) separately. It can be observed from Fig. 3.4(a) that the RSR value of most normal users is close to 0.5, which means that the amount of messages sent by normal users is similar to the amount of messages they received. Meanwhile, there are some normal users who send far less messages than received, but this rarely happened among spammers. Similarly, in Fig. 3.4(b), the RU value of most normal users is close to 0.5, indicating



(a) CDF curve of feature RSR

(b) CDF curve of feature RU

Figure 3.4: CDF curves of features RSR & RU.

that the number of normal users in sending is basically the same as the number of normal users in receiving. Conversely, spammers will send messages to a large number of different users, while receive very few replies.

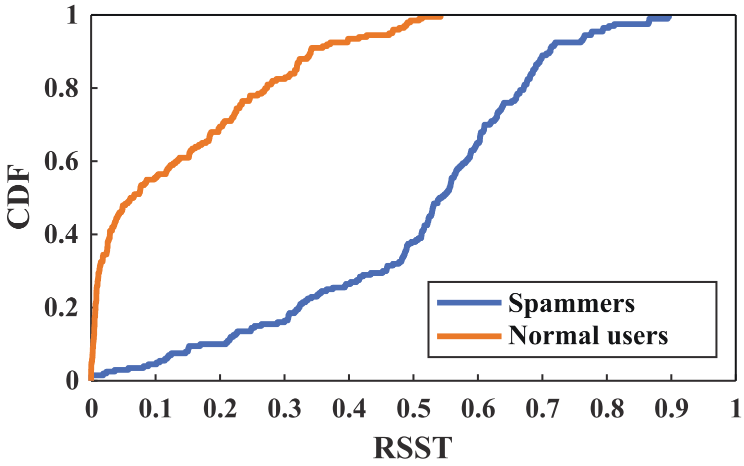
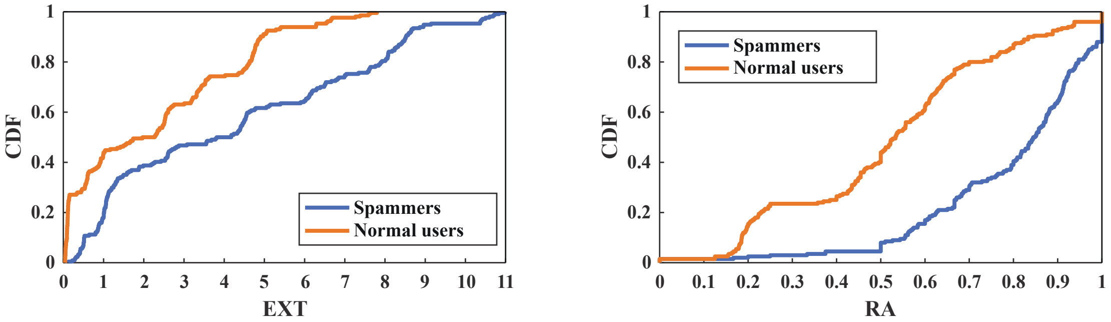


Figure 3.5: CDF curve of feature RSST.

Fig. 3.5 is the CDF curve of proposed feature RSST. The difference between spammers and normal users can be clearly seen from the figure. Specifically, most spammers have nearly 60% of their messages sent during abnormal time periods, while most normal users have less than 1% of their messages sent during abnormal time periods. In addition, it can also be observed from Fig. 3.5 that a small number of normal users will send more messages between midnight and six in the morning. This behaviour is also understandable as this very small number of users might be night workers.

Fig. 3.6 is the CDF curve of proposed features EXT and RA. In general, the cu-



(a) CDF curve of feature EXT

(b) CDF curve of feature RA

Figure 3.6: CDF curves of features EXT & RA.

mulative distribution of spammers and normal users in Fig. 3.6(a) and Fig. 3.6(b) are significantly different, which illustrates the effectiveness of the features EXT and RA in characterizing the behaviour differences between spammers and normal users. To be more specifical, from Fig. 3.6(a), we can find that the variance of activeness between spammers and normal users in each relation is relatively large, indicating that normal users in *Tagged.com* social network will also have a preference towards different relations, but they are not as purposeful as spammers. In Fig. 3.6(b), we can tell that spammers and normal users are truly different on relation *Add friend*. Normal users will choose to perform relation *Add Friend* in order to gain insights into users with good feelings, and the number of times they receive friend requests depends on whether their personal charm will attract the attention of other users. Compared with normal users, most spammers will not actively perform relation *Add Friend*, which is also caused by the characteristics of the *Tagged.com* social network (users can also perform any other relations without performing relation *Add Friend*, including *Message* and *Report Abuse*). It can also be seen in Fig. 3.6(b) that there are still a small number of spammers who will actively perform relation *Add Friend*. It may be because spammers chose more sophisticated method to pretend to be normal users. After defrauding the trust of normal users, they achieved extreme malicious behaviours such as defraudation.

In general, Fig. 3.3 to Fig. 3.6 clearly show the difference between the features described above in characterizing the behaviour of spammers and normal users, which fully proves the effectiveness of the features based on non-content data proposed in this chapter.

3.4 Summary

In this chapter, we first introduces the *Tagged.com* dataset that drives the research of our spammer detection work, and uses statistical analysis to reasonably infer the actual relation name corresponding to each relation. Then, based on the analysis of the behaviour of spammers in different relations, we propose the hypothesises to discriminate the behavioural difference of spammers and normal users on multi-relational social networks. At last, a series of behaviour features based on relational data are given and the validity of the features is verified.

CHAPTER 4

GRAPH-EMBEDDING BASED SOCIAL SPAMMER DETECTION

The analysis based on user behaviours can, to a certain extent, take into account the differences in the performance of users in different relations and dig out the deep semantic information in the heterogeneous relational network without relying on the previous conditions of content data. Nevertheless, this kind of behaviour analysis is still limited, far from being able to meet the integration needs between different relations on multi-relational social networks. Therefore, in this chapter, we shall develop a new multi-relational graph embedding model based on probability matrix decomposition to extract and fuse the hidden information behind the multiple relations on social networks. We name it "Send-Receive" Role Separable Graph-Embedding Model (*RS-GEM*) [91]. First, we build a graph in a shared embedding space, where nodes represent for users and edges represent for relations between users. Second, the number of interactions between the sending (*src*) and receiving (*dst*) users is extracted as interaction vectors. Third, the sending (*src*) user feature matrix and receiving (*dst*) user feature matrix are constructed, and the user-user interaction vector is represented by dot product. The difference between these two vectors is used to fit the probability matrix decomposition model, and the constraint conditions are added to prevent the overfitting problem in the optimization process. Finally, the embedding features of each user in multi-relational social networks are obtained through the joint of multiple relations. In order to further verify the effectiveness of our proposed *RS-GEM*, in the later of this chapter, we use the outstanding topological graph-based features and time sequence-based features in existing research as baselines, and compare them on the real-world *Tagged.com* dataset. Finally, the parameters (dimensions) of the embedding features are discussed.

4.1 "Send-Receive" Role Separable Graph-Embedding Model (*RS-GEM*)

In this section, we begin by introducing the motivation of proposing the "Send-Receive" Role Separable Graph-Embedding Model (*RS-GEM*). Then, we introduce the overall framework of RS-GEM, followed by the model formulation.

4.1.1 Motivation

Different from platforms such as e-mail and e-commerce, social network platforms pay much more attention to diversified development. In social networks, users can achieve the purpose of communication through various forms (*relations*). This diverse interaction relations also gives spammers in social networks an opportunity. Fig. 4.1 uses a toy example of a social network with three interaction relations ("*Message*", "*Report Abuse*" and "*Give a gift*") to clearly illustrate the importance of potential connections between relations.

To be more specific, in relation *Message*, users who send messages to too many users look like spammers. Nevertheless, in relation *Give a gift*, a user who has received gift from other users is most likely a normal user. In relation *Report Abuse*, a user is frequently reported by other users (shown as "block" in Fig. 4.1), then the credibility of this user will be reduced. In Fig. 4.1, the suspicious user in the middle has sent messages to too many users, which makes him look like a spammer. In other word, in relation *Message*, he will be detected as a spammer. However, in relation "Give a Gift", the user in the middle have received gifts from user u_i and user u_j, which means that this user will be more likely to be detected as a normal user in this relation. Furthermore, u_i and user u_j who have sent gifts to the user in the middle have been reported by many other users, that is, u_i and user u_j are untrustworthy users with extremely low credit. Hence, this series of behaviours is most likely the spammer in the middle who used the loopholes in the relations to confuse the detection system and clear his own suspicions.

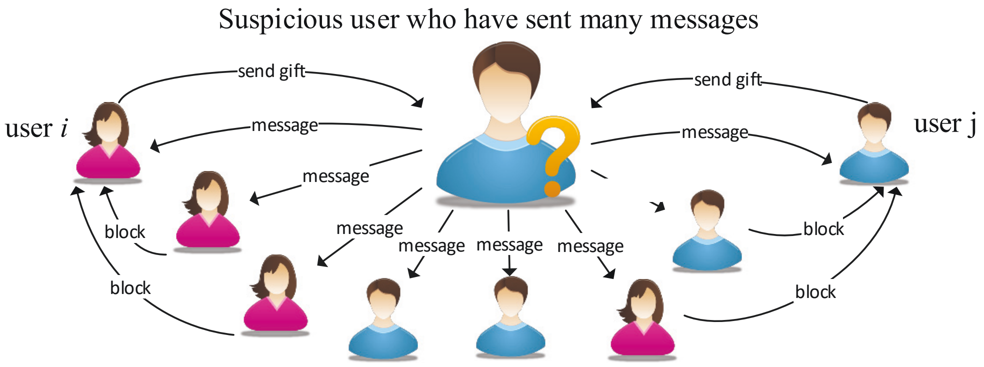


Figure 4.1: Motivation of RS-GEM. The middle user sends messages to too many users, which makes him suspicious, like a spammer. However, he received gifts from users i and j, which is a powerful indicator that he is a good user. But we found that the users who gave the gift are actually low-credit users who were blocked by others, so the fact may be that the spammer is trying to deceive the detection system.

In a social network that contains only three interactive relations, spammers can find the potential characteristics between the relations to cover their identity. Then, the potential connections between relations in social networks with richer types of interaction relations will only become more complicated. And spammers who exploit the potential connections between relations will hide deeper. Therefore, mining the hidden information between relations is of great significance for the detection of spammers in multi-relational social networks.

So far, the research on detecting spammers based on the relation characteristics in the multi-relational social network is still very limited, and most of these studies analyze multiple relations in the social network separately. For example, Fakhraei et al. [27] tried to generate each relation in the social network a network topological graph. They assumed that spammers are the important nodes in the graph, which usually with more links from other nodes. Xing et al. [86] also tried to use frequent relation sequences in social networks to construct spammer features. Nevertheless, these studies did not consider the deep hidden information of relations in multi-relational social networks.

In chapter 3, we analyse the difference of behaviours of spammers and normal users in several relation types separately. Afterwards, we give the definitions of a series of behavioural characteristic indicators to detect spammers in relational social networks. These behaviour indicators take the potential connections between relations and the differences in the roles of users in different relations into account to a certain extent. For instance, feature EXT based on hypothesis 2 comprehensively considers the user’s activeness in each relation, rather than just observing the user's behaviour pattern in a specific relation. However, the analysis of users' behaviour is still limited, and some deep-seated hidden information are difficult to dig through behaviour patterns. Therefore, in this chapter, we propose the "Send-Receive" Role Separable Graph- Embedding Model (*RS-GEM*) to extract and fuse the hidden information of heterogeneous relations for spammer detection in multi-relational social networks.

4.1.2 Framework Overview

In multi-relational social networks, let $\mathscr{U} = \{u_1, \cdots, u_n\}$ be the set of n users who are connected by m kinds of relations denoted as $\mathscr{R} = \{r_1, \cdots, r_m\}$, where *relations* refer to the interactions between users in *Tagged.com* (e.g., *Give a Gift*, *View Profile*, *Add Friend*, etc.). We use $C_{ij}^{(r)}$ to denote that user u_i performs relation r towards user u_j. It should be noted that the relation is directional, therefore $C_{ij}^{(r)}$ and $C_{ji}^{(r)}$ are different. $C_{ij}^{(r)}$ is the number of times that relation r been performed by user u_i towards user u_j and $C_{ji}^{(r)}$ is the same relation but from user u_j to user u_i. Usually $C_{ij}^{(r)}$ and $C_{ji}^{(r)}$ are not equal.

Firstly, we model all users and all types of relations in a shared embedding space, where the dimension of the embedding space is defined by the detector. Let's note the set of all interaction records as $\mathbf{C}$. Then a graph $\mathscr{G}$ can be built with the vertices representing users and the edges representing relations. Considering the two vertices connected by an edge in the directed graph $\mathscr{G}$: the initiator node (*src*) and the receiver node (*dest*) are different. For example, the main purpose of spammers is to spread spam to huge number of users, so he/she usually appears as an initiator (*src*) node. Therefore, we perform role separation based on the two different roles of users: "Send" and "Receive". Fig. 4.2 illustrates the proposed *RS-GEM* in the network with two relations, which considering both *src* node and *dest* node. It can be seen that we have a initiator user vector $\boldsymbol{u}_i^{src}$ and a receiving user vector $\boldsymbol{u}_j^{dest}$, which are mapped to the shared embedding space of two types of relations relation k and relation l. On the basis of minimizing the prediction error of the red part as much as possible, we fuse the feature vectors $\boldsymbol{u}_i^{src}$ and $\boldsymbol{u}_i^{dest}$ of user u_i in each relation to obtain the embedding feature $\boldsymbol{s}_i$ of user u_i in the multi-relational social networks.

To be more specific, for each user $u_i \in \mathscr{U}$, we define two vectors $\boldsymbol{u}_i^{src} \in \mathbb{R}^z$ and $\boldsymbol{u}_i^{dest} \in \mathbb{R}^z$ to represent. Then we build *src* user matrix denoted as $\mathbf{U} = \{\boldsymbol{u}_1^{src}, \cdots, \boldsymbol{u}_i^{src}, \cdots, \boldsymbol{u}_n^{src}\}^\top$, and *dest* user matrix denoted as $\mathbf{V} = \{\boldsymbol{u}_1^{dest}, \cdots, \boldsymbol{u}_i^{dest}, \cdots, \boldsymbol{u}_n^{dest}\}^\top$ for each user, where $\boldsymbol{u}_i$ is the graph embedding vector for *src* user u_i^{src}, and $\boldsymbol{v}_i$ is the graph embedding vector for *dest* user u_i^{dest}. The graph embedding vectors of each user as the *src* user and *dest* user

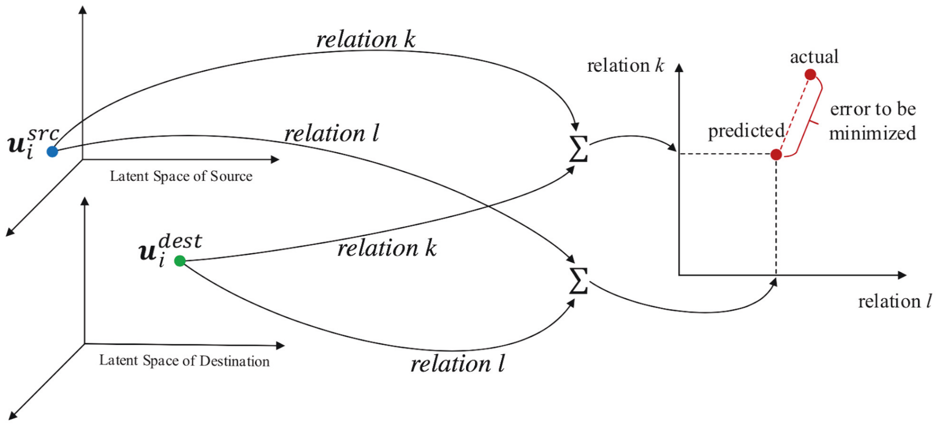


Figure 4.2: Illustration of the RS-GEM on two relations.

on relation r are denoted as $\boldsymbol{u}_i^{(r)}$ and $\boldsymbol{v}_i^{(r)}$ separately. We represent the interaction value of user u_i^{src} and user u_j^{dest} in relation r in the form of dot product. Furthermore, we define Eq. (4.1) to fit the difference between the actual number of interactions between the *src* user u_i^{src} and the *dest* user u_j^{dest} on relation r and the dot product of the vector between the two users:

$$C_{ij}^{(r)} - \boldsymbol{u}_i^{(r)} \boldsymbol{v}_j^{(r)} \tag{4.1}$$

Therefore, we can optimize the graph embedding vector for each user in relation r based on the difference between the actual number of interactions and the vector dot product, and then optimize the implementation of the proposed *RS-GEM*.

4.1.3 Model Formulation

We explain the overall framework of *RS-GEM* and illustrate the basic vector representation in section 4.1.2. Based on section 4.1.2, we implement the probability matrix factorization in this section to realize the matrixed graph embedding model.

Let $\mathbf{C}^{(r)} \in \mathbb{R}^{n*n}$ denotes the interaction matrix, where $C_{ij}^{(r)}$ represents for the number of interactions that user u_i^{src} performs relation r towards usre u_j^{dest}. If user u_i^{src} didn't perform relation r towards usre u_j^{dest}, then $C_{ij}^{(r)} = 0$, otherwise $C_{ij}^{(r)}$ is an integer greater than 0. $\mathbf{U}^{(r)} \in \mathbb{R}^{n*k}$ and $\mathbf{V}^{(r)} \in \mathbb{R}^{n*k}$ are the *src* users matrix and *dest* users matrix on relation r separately, where $\mathbf{U}^{(r)} = \{\boldsymbol{u}_1^{(r)}, \cdots, \boldsymbol{u}_i^{(r)}, \cdots, \boldsymbol{u}_n^{(r)}\}$ $(1 \le i \le n)$, $\mathbf{V}^{(r)} = \{\boldsymbol{v}_1^{(r)}, \cdots, \boldsymbol{v}_j^{(r)}, \cdots, \boldsymbol{v}_n^{(r)}\}$ $(1 \le j \le n)$. Each column vector $\boldsymbol{u}_i^{(k)}$ and $\boldsymbol{v}_j^{(k)}$ are the k-dimensional vectors for *src* user u_i^{src} and *dest* user u_j^{dest} separately. We define the following probability matrix factorization model:

$$\mathbf{C}^{(r)} = \mathbf{U}^{(r)}(\mathbf{V}^{(r)})^{\top} + E_1^{(r)} \tag{4.2}$$

where $E_1^{(r)}$ is an error matrix, and each element is usually modeled as a Gaussian observation error. We use $\mathcal{N}(0, \sigma^2_{\mathbf{C}^{(r)}})$ to represent the disturbance error. Obviously, $\mathbf{C}^{(r)}$ is a sparse matrix. Hence, the matrix $\mathbf{H}^{(r)}$ is constructed to constrain the non-zero elements in $\mathbf{C}^{(r)}$ and reduce the optimization process. We use $\odot$ to combine the matrix $\mathbf{H}^{(r)}$ and the interaction number matrix $\mathbf{C}^{(r)}$, where $H_{ij}^{(r)} = 1$ when $C_{ij}^{(r)} \neq 0$, otherwise $H_{ij}^{(r)} = 0$. Eq. (4.3) defines the conditional probability of the observation matrix:

$$P(\mathbf{C}^{(r)}|\mathbf{U}^{(r)}, \mathbf{V}^{(r)}, \sigma^2_{\mathbf{C}^{(r)}}) = \prod_{i=1}^{n}\prod_{j=1}^{n}[\mathcal{N}(C_{ij}^{(r)}|\boldsymbol{u}_i^{(r)}\boldsymbol{v}_j^{(r)}, \sigma^2_{\mathbf{C}^{(r)}})]^{H_{ij}^{(r)}} \tag{4.3}$$

Furthermore, Eq. (4.4) and Eq. (4.5) define the conditional probability model of the *src* user matrix $\mathbf{U}^{(r)}$ and the *dest* user matrix $\mathbf{V}^{(r)}$:

$$P(\mathbf{U}^{(r)}|\sigma^2_{\mathbf{C}^{(r)}}) = \prod_{i=1}^{n}\mathcal{N}(\boldsymbol{u}_i^{(r)}|0, \sigma^2_{\mathbf{U}^{(r)}}I) \tag{4.4}$$

$$P(\mathbf{V}^{(r)}|\sigma^2_{\mathbf{C}^{(r)}}) = \prod_{i=1}^{n}\mathcal{N}(\boldsymbol{v}_i^{(r)}|0, \sigma^2_{\mathbf{V}^{(r)}}I) \tag{4.5}$$

Then, according to Bayesian theory, Eq. (4.6) can be obtained:

$$P(\mathbf{U}^{(r)}, \mathbf{V}^{(r)}|\mathbf{C}^{(r)}, \sigma^2_{\mathbf{C}^{(r)}}, \sigma^2_{\mathbf{U}^{(r)}}, \sigma^2_{\mathbf{V}^{(r)}}) \propto P(\mathbf{C}^{(r)}|\mathbf{U}^{(r)}, \mathbf{V}^{(r)}, \sigma^2_{\mathbf{C}^{(r)}})P(\mathbf{U}^{(r)}|\sigma^2_{\mathbf{C}^{(r)}})P(\mathbf{V}^{(r)}|\sigma^2_{\mathbf{C}^{(r)}}) \tag{4.6}$$

Incorporating Eq. (4.3), Eq. (4.4) and Eq. (4.5) into Eq. (4.6), we can obtain the representation of the *src* users' graph embedding matrix $\mathbf{U}^{(r)}$ and the *dest* users' graph embedding matrix $\mathbf{V}^{(r)}$, as shown in Eq. (4.7):

$$\begin{aligned} &\log P(\mathbf{U}^{(r)}, \mathbf{V}^{(r)}|\mathbf{C}^{(r)}, \sigma^2_{\mathbf{C}^{(r)}}, \sigma^2_{\mathbf{U}^{(r)}}, \sigma^2_{\mathbf{V}^{(r)}}) \propto -\frac{1}{2\sigma^2_{\mathbf{C}^{(r)}}}\sum_{i=1}^{n}\sum_{j=1}^{n}H_{ij}^{(r)}(C_{ij}^{(r)} - \boldsymbol{u}_i^{(r)}(\boldsymbol{v}_j^{(r)})^{\top}) \\ &-\frac{1}{2\sigma^2_{\mathbf{U}^{(r)}}}\sum_{i=1}^{n}\boldsymbol{u}_i^{(r)}(\boldsymbol{u}_i^{(r)})^{\top} - \frac{1}{2\sigma^2_{\mathbf{V}^{(r)}}}\sum_{i=1}^{n}\boldsymbol{v}_i^{(r)}(\boldsymbol{v}_i^{(r)})^{\top} \end{aligned} \tag{4.7}$$

By calculating the Maximum A Probability (MAP) of the *src* users' graph embedding matrix $\mathbf{U}^{(r)}$ and the *dest* users' graph embedding matrix $\mathbf{V}^{(r)}$ based on Eq. (4.7), we can define the objective function of *RS-GEM*, as shown in Eq. (4.8):

$$\mathscr{J} = \sum_{r\in\mathscr{R}}\frac{1}{\sigma^2_{\mathbf{C}^{(r)}}}\cdot\left\|\mathbf{H}^{(r)}\odot(\mathbf{C}^{(r)} - \mathbf{U}^{(r)}(\mathbf{V}^{(r)})^{\top})\right\|_F^2 + \frac{1}{\sigma^2_{\mathbf{U}^{(r)}}}\left\|\mathbf{U}^{(r)}\right\|_F^2 + \frac{1}{\sigma^2_{\mathbf{V}^{(r)}}}\left\|\mathbf{V}^{(r)}\right\|_F^2 \tag{4.8}$$

where $\|\cdot\|$ is the Frobenius norm. $\frac{1}{\sigma^2_{\mathbf{U}^{(r)}}}\left\|\mathbf{U}^{(r)}\right\|_F^2 + \frac{1}{\sigma^2_{\mathbf{V}^{(r)}}}\left\|\mathbf{V}^{(r)}\right\|_F^2$ is the constraint condition added to prevent overfitting in the optimization process. Afterwards, we use gradient descent to optimize the objective function (Eq. (4.8)), as shown in Eq. (4.9):

$$\begin{aligned}\frac{\partial \mathscr{J}}{\partial \mathbf{U}^{(r)}} &= \frac{1}{\sigma^2_{\mathbf{C}^{(r)}}} \cdot \mathbf{V}^{(r)\top}(\mathbf{H}^{(r)} \odot (\mathbf{U}^{(r)}(\mathbf{V}^{(r)})^\top - \mathbf{C}^{(r)})) \\ \frac{\partial \mathscr{J}}{\partial \mathbf{V}^{(r)}} &= \frac{1}{\sigma^2_{\mathbf{C}^{(r)}}} \cdot \mathbf{U}^{(r)\top}(\mathbf{H}^{(r)} \odot (\mathbf{U}^{(r)}(\mathbf{V}^{(r)})^\top - \mathbf{C}^{(r)}))\end{aligned} \tag{4.9}$$

In each iteration, we update the *src* users' graph embedding matrix $\mathbf{U}^{(r)}$ and the *dest* users' graph embedding matrix $\mathbf{V}^{(r)}$ so that:

$$\begin{aligned}\mathbf{U}^{(r)} &= \mathbf{U}^{(r)} - \xi \frac{\partial \mathscr{J}}{\partial \mathbf{U}^{(r)}} \\ \mathbf{V}^{(r)} &= \mathbf{V}^{(r)} - \xi \frac{\partial \mathscr{J}}{\partial \mathbf{V}^{(r)}}\end{aligned} \tag{4.10}$$

where ξ is the step size, and we set the step size to 0.001 in the experiment.

Finally, the feature vectors of the two roles of the user in different relations are spliced to obtain the final graph embedding feature that combines multiple relations and multiple roles, as shown in Eq. (4.11)

$$\boldsymbol{s}_i = \bigoplus_{r \in \mathscr{R}} (\boldsymbol{u}_i^{(r)}, \boldsymbol{v}_i^{(r)}) \tag{4.11}$$

where $\bigoplus$ is the splicing symbol.

4.2 Experiment and Analysis

4.2.1 Experiment Setup

In section 3.2, we find that the statistics of each relation are close to evenly distributed in 10 days. That is to say, the attribute *day* has a very small influence on the detection. Therefore, in order to reduce the hardware requirements of the experiment, we take tens of millions of interaction records in a single day (data from "*day 0*") as analysis samples for experiments. Statistics of the dataset is shown in Table 4.1.

According to the analysis on section 3.2, relation 7 in *Tagged.com* dataset has speculated to be "Report Abuse". In the report abuse mechanism of *Tagged.com*, the user u_i^{src} reports user u_j^{dest} for breaching of terms and conditions. Nevertheless, the reported user is not necessarily spammer. The detection framework proposed by Fakhraei et al. [27]

Table 4.1: Statistics of experiment dataset

Dataset	#User	#Spammer	#Normal User	#Interaction Record
Tagged.com	4,111,179	182,939	3,928,240	85,470,637

uses the probabilistic soft logic (PSL) rule to take advantages of the relation *Report Abuse*, and then combines it with the classification results to improve the performance. There are two important PSL rules proposed as below.

$$\begin{aligned} &\text{Normal user}(u_i^{\text{src}}) \wedge \text{Report}(u_i^{\text{src}}, u_j^{\text{dest}}) \rightarrow \text{Spammer}(u_j^{\text{src}}),\\ &\text{Spammer}(u_j^{\text{dest}}) \wedge \text{Report}(u_i^{\text{src}}, u_j^{\text{dest}}) \rightarrow \text{Normal user}(u_i^{\text{src}}). \end{aligned} \tag{4.12}$$

The PSL rules restrict evaluation to users who appear in the reporting mechanism. In order to be consistent with related research, we adopt the same test plan and extract the users who appear in the report relation as our test data.

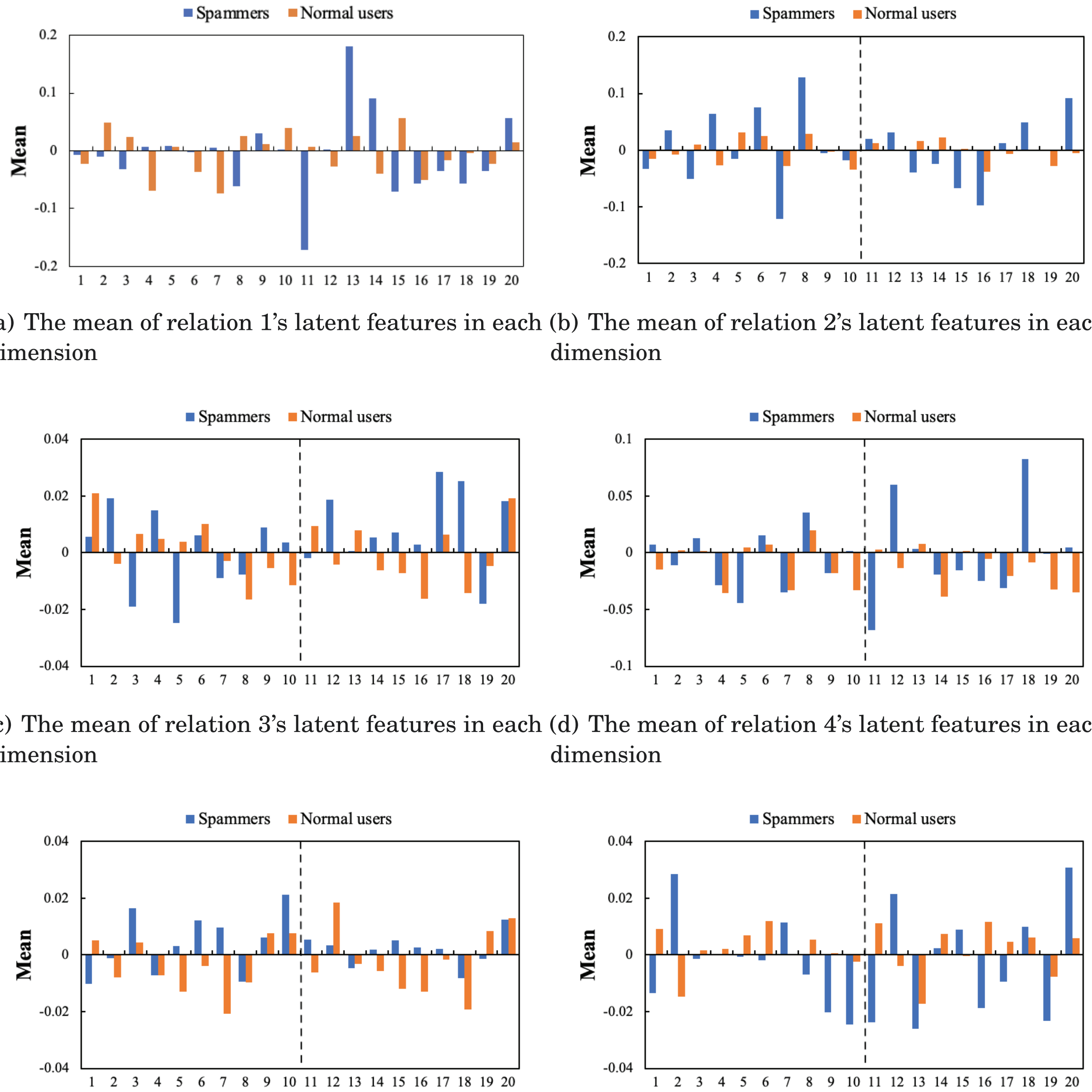
Since the ground-truth label of each user is provided by the data set, we use standard indicators (P-R-F), including precision (P), recall (R), and F-measure (F) to evaluate the effectiveness of the proposed RS-GEM model, as shown in Eq. (4.13). In addition, all indicators are calculated based on the class of spammers.

$$P = \frac{TP}{TP+FP}, \; R = \frac{TP}{TP+FN}, \; F = \frac{2PR}{P+R}, \tag{4.13}$$

where precision P is the number of true positive results divided by the number of all positive results, including those not identified correctly, recall R is the number of true positive results divided by the number of all samples that should have been identified as positive, F-measure F is a measure of a test's accuracy and is calculated from the precision and recall of the test, TP represents for true positive, which means the number of spammers that have been detected correctly, on contrast, FP represents for false positive, which means the number of spammers that have been detected incorrectly, and FN is false negative, which means the number of spammers that have been missed by the model. According to the application scenario, these indicators can be weighed.

4.2.2 Effectiveness Analysis

For the reason that the test dataset has already taken the relation *Report Abuse* into consideration according to the PSL rules (refer to Eq. (4.12)), we removed relation 7 (*Report Abuse*) when extracting the embedding features of each relation. Fig. 4.3 and Fig. 4.4 takes the feature dimension of 20 as an example to show that the embedding features of the proposed *RS-GEM* can clearly distinguish the difference between normal users and spammers.



(a) The mean of relation 1's latent features in each dimension

(b) The mean of relation 2's latent features in each dimension

(c) The mean of relation 3's latent features in each dimension

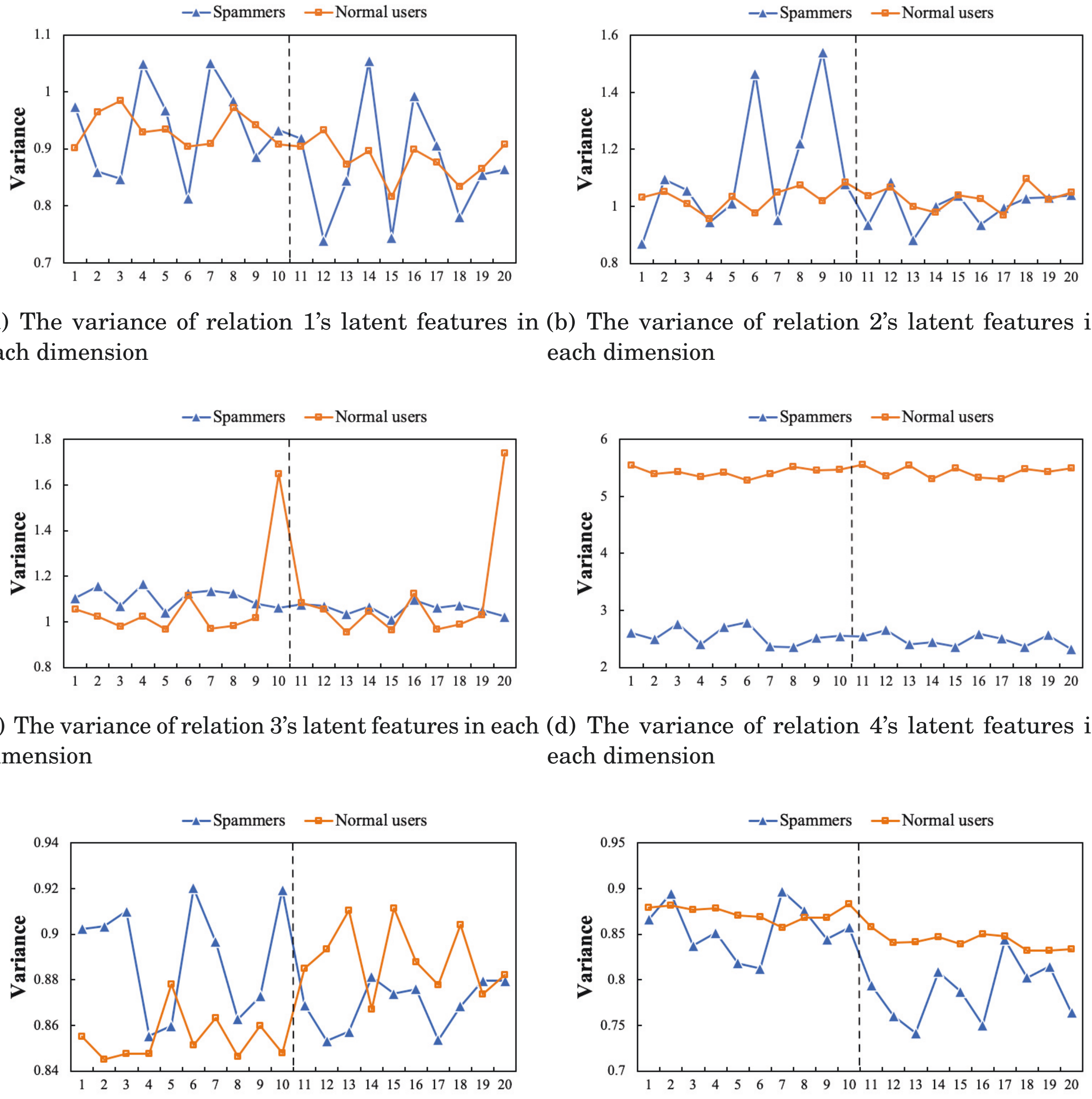
(d) The mean of relation 4's latent features in each dimension

(e) The mean of relation 5's latent features in each dimension

(f) The mean of relation 6's latent features in each dimension

Figure 4.3: Effectiveness analysis of latent features.

Fig. 4.3 are the average values of the embedding feature vectors of normal users and spammers in each dimension from relation 1 to relation 6. While Fig. 4.4 are the variances of the embedding feature vectors of normal users and spammers in each dimension from relation 1 to relation 6. Each figure in Fig. 4.3 and Fig. 4.4 is divided by a



(a) The variance of relation 1's latent features in each dimension

(b) The variance of relation 2's latent features in each dimension

(c) The variance of relation 3's latent features in each dimension

(d) The variance of relation 4's latent features in each dimension

(e) The variance of relation 5's latent features in each dimension

(f) The variance of relation 6's latent features in each dimension

Figure 4.4: Effectiveness analysis of latent features (continued).

dotted line in the middle. The left side of the dotted line is the embedding feature vector of the user as the *src* user, and the right side of the dotted line is the embedding feature vector of the user as the *dest* user. It can be clearly observed from the figure that, in each relation, the difference in embedding features between spammers and normal users. For example, as shown in Fig. 4.3(a), in relation 1, the average value of the embedding features of spammers and normal users in each dimension is very large. The average values of embedding features of normal users are relatively close in each dimension, while spammers will experience strong fluctuations in some dimensions. And as shown in Fig. 4.4(d), in relation 4, the variance of spammers in each dimension is always smaller than that of normal users, which means that in relation 4, the fluctuations in the values of the embedding features of spammers in all dimensions are smaller than that of normal users. In addition, comparing the left and right sides of the dotted line in one figure, we can also find that the different roles played by users in some relations are also reflected in embedding features. For example, in Fig. 4.3(a) and Fig. 4.3(d), when spammers are the *dest* users, the embedding features taken out by spammers will occur jumping wave in certain dimensions.

Through the above analysis, it is found that the proposed “Send-Receive” Role Separable Graph-Embedding Model (*RS-GEM*) can effectively characterize the differences between spammers and normal users and the differences when users play different roles.

4.2.3 Cross Validation and Comparison

According to the investigation of the features of spammer detection in existing researches, as illustrated in chapter 2, the graph-based features and time sequence-based features comprehensively consider the relational characteristics in the multi-relational social networks to a certain extent. In order to further verify the effectiveness of our proposed embedding features, in this section, we first compare the embedding features extracted from our proposed *RS-GEM* with the graph-based features and the time sequence-based features separately. Graph-based features and sequence-based features are then combined to demonstrate the effectiveness of our proposed approach.

Specifically, Graph-based features are computed using *Graphlab Create* [1], including *Page Rank* [59], *Graph Coloring* [39], *Weakly Connected Components* [62], *k-core* [8], *Triangle Count* [70], and *Degree* (including total degree, in-degree, and out-degree of each node) [27] on each type of relation. Thus, we get a total of 56 graph-based features,

[1] `https://turi.com/`

where 8 for each type of relation. It should be noted that topological structure graphs with directions are constructed for each relation. And graph-based features are computed based on these constructed topological structure graphs. For sequence-based features, we compute them using *Sequential k-gram Features* [27], where $k = 2$. There are 7 relations in the data set, and we finally get 49 sequence-based features. In our multi-relational embedding features, we set the dimension of features as 30 for overall comparison. We shall discuss other dimensions of the features later in this section.

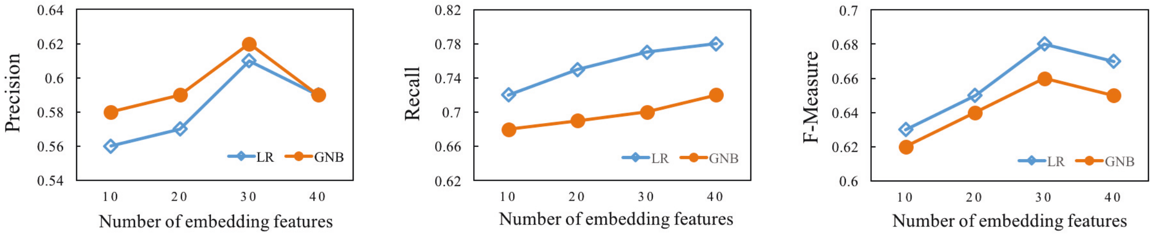
After obtaining the baseline features, we use 10 different random seeds to split the training and test data sets for evaluation. Since the key concerns is to evaluate the quality of features extracted from multi-relational data rather than new classification algorithms, we choose two simple classic supervised models: Logistic Regression (*LR*) and Gaussian Naive Bayes (*GNB*) as they are the most commonly used models for binary classification problems and perform better in our circumstance.

Table 4.2: Comparison of two classifiers with different kinds of features

	Logistic Regression			Gaussian Naive Bayes		
Features	Precision	Recall	F-measure	Precision	Recall	F-measure
Graph	0.4537	0.6390	0.5308	0.5978	0.3840	0.4675
Sequential	0.4907	0.8620	0.6253	0.4168	0.9320	0.5759
Graph+Sequential	0.5316	0.8600	0.6570	0.4571	0.9260	0.6120
RS-GEM (dimension=30)	**0.6138**	0.7730	**0.6844**	**0.6165**	0.7020	**0.6566**

From the table 4.2, we can see that the embedding features extracted by *RS-GEM* get much higher score than other features on the F-measure indicator in both *LR* and *GNB*. This means that we can catch spammers more accurately with minimal harm to normal users. Excitingly, the precisions of embedding features has always been the highest, which proves that the proposed features can reveal most spammers, while the recall rate is slightly reduced. In terms of recall, although the sequential features rank the highest, They perform the worst in terms of precision. This means that they treat more users as spammers, which has a great impact on normal users.

In order to thoroughly examine the performance of the proposed *RS-GEM*, we shall analyze its performance by changing the size of the embedding space from 10 to 40. Fig. 4.5 shows the performance of embedding features with different dimensions on three metrics separately. It can be seen from Fig. 4.5(b) that the recall increases when the dimension raises, indicating that increasing the dimension will expose more spammers. Meanwhile, the precision and F-measure peak when the dimension is 30, and then decrease. That is, if the dimension continues to grow after reaching 30, *RS-GEM* will lose its precision by listing more users as spammers. In general, we can get the most effective performance when adjust the dimension to 30 on the Tagged.com data set. But this is not to say that dimension 30 is necessarily the best. It still needs to be judged based on the data set. Our speculation is that the number of embedding features should increase with the number of relation types, because more types of relations mean more complex interactions.



(a) Precision

(b) Recall

(c) F-Measure

Figure 4.5: Impact of the number of dimensions in our *RS-GEM* model.

4.3 Summary

In response to the need for the integration of different interaction relations in multi-relational social networks, in this chapter, we propose a new "Send-Receive" Role Separable Graph-Embedding Model (*RS-GEM*) to extract and fuse the hidden information of heterogeneous relations in multi-relational social networks. The *RS-GEM* first extract the number of interactions between the sending (*src*) and receiving (*dst*) users as interaction vectors. Then, the sending (*src*) user feature matrix and receiving (*dst*) user feature matrix are constructed, and the user-user interaction vector is represented by dot product. The difference between these two vectors is used to fit the probability matrix decomposition model, and the constraint conditions are added to prevent the overfitting problem in the optimization process. Finally, the embedding features of each user in multi-relational social networks are obtained through the joint of multiple relations. In the second half of this chapter, a semantic analysis of the embedding features extracted from the *RS-GEM* is performed. In addition, the graph-based features and the time sequence-based features which are outstanding in existing researches are computed on the *Tagged.com* dataset. A comparative verification is carried out, which fully demonstrated the effectiveness of the proposed *RS-GEM*. At last, we discuss the influence of dimensions when taking embedding features.

CHAPTER 5

Sequence-based Social Spammer Detection

In Chapter 4 we use graph embedding method to deeply explore the latent information hidden behind relations in a multi-relational social network to identify weather the user is a spammer or not. However, we overlooked one point: the order in which the relations occur. Hence, in this chapter, we shall have a deep research on how to use the chronological sequence of relations (i.e., [viewing profile→ thumbs up→ forward posts→ sending messages]) to detect spammers in multi-relational social networks. In another word, we shall use user's relational sequence as input to detect the identity of the user.

In the literature, some scholars have made some effort on the research of chronological sequence for spammer detection. For instance, Fakhraei et al. [27] define a short sequence of segments consisting of k consecutive actions, called a k-gram. They use the number of occurrences of the k-gram sequence to partially reveal the differences between spammers and normal users. Though the proposing k-gram features can somehow capture the sequence in the aspects of short-term, they may omit the long-term dependency of the sequence. Further, to capture significant information from a longer chain of sequences and to investigate the predictive power of that information, Peng et al. used a mixture of Markov models to overcome the limitations of small k [63]. In particular, Peng et al. use the ratio of the posterior probability and its logarithm as a small feature set. They train their classifier by treating this small feature set as a small set of significant sequences from a long sequence chain.

Generally, existing sequence-based methods make use of either long-term or short-term dependencies, which may be more likely to ignore any potential correlations between them. In addition, most existing sequence-based methods only trained on a limited training data sets, which usually suffers from overfitting [46, 63]. What if undetected spammers deliberately or intentionally do not follow the known patterns of behaviour they normally possess? In order to deal with this issue, we aim to reveal the deeper information hidden behind the sequences and thus more accurately identify abnormal user behaviour. Encouraged by deep sequential networks developed in recommendation system [75, 77, 81, 92, 96] , we take advantage of both *long-term* and *short-term* dependencies to fully explore the deeper complementary information behind the multi-relational sequences. To be more specific, long-term dependencies model the overall behaviour of users on a multi-relational social network, and we use user's relational sequence throughout a day to represent their overall behaviour. For short-term dependencies, we represent partial behavioural information using the last n ($1 \le n < 10$) relational sequences. Furthermore, we consider short-term dependencies both on the individual-level and on the union-level. On the individual-level, we only consider a relation that the user has recently executed, as this relation may trigger his/her next action. At the union level, we capture the collective impact between the union of relations performed by the user.

In this chapter, we propose a novel *Multi-level Dependency Model* (*MDM*). The *MDM* comprehensively exploits the behavioural characteristics of users from both long-term and short-term. Especially in the short term, we also consider the individual-level and the union-level [89]. The individual-level dependency considers only a single recent behaviour that may trigger subsequent behaviours. In contrast, the union-level dependency considers the collective influence among a union of relations that are involved in the user's short-term behaviour sequence. *MDM* is able to improve the performance of spammer detection in multi-relational social networks by exploiting deeper information hidden behind the sequence of user relations.

5.1 Multi-level Dependency Model

In this section, an explanation of the individual-level and union-level dependencies of user interaction sequences that motivated our work are firstly given. Then, we formulate the social spammer detection problem. Finally, we present the overall framework of our proposed multi-level dependency model(*MDM*).

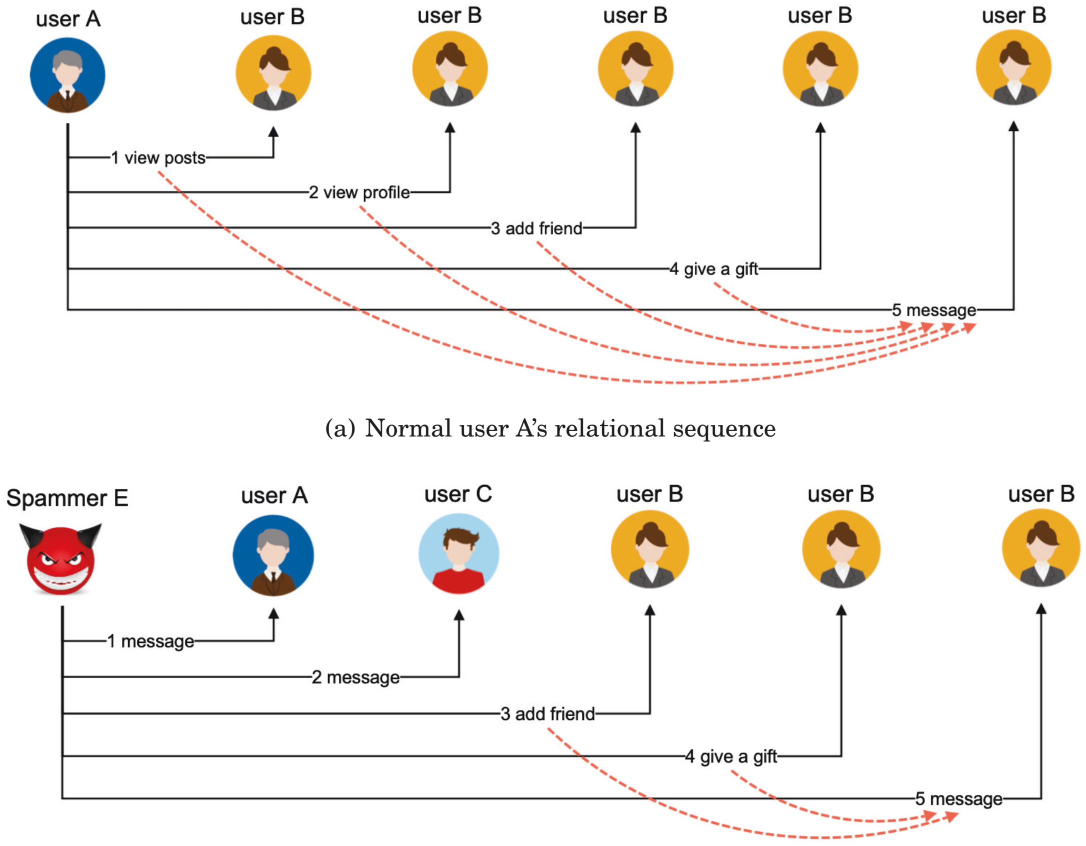
5.1.1 Motivation

Encouraged by the deep sequential method recommendation system [54, 75, 92, 96], user's most recent n purchase plays an essential role in predicting the next item the user will want to buy. In the environment of multi-relational social networks, we assume that the ultimate goal of the spammer is to *send messages* to as many users as possible with the intention of spreading false information. Therefore, the most recent n relations before *send messages* of users' can be used to represent the short-term dependency of their relational sequence, rather than a full-day relation sequence (long-term). Moreover, we investigate the users' short-term relational sequence from both individual-level and union-level dependency. We give an toy example in Fig. 5.1 and Fig. 5.2 to illustrating in detail.

Fig. 5.1(a) gives an example of a normal user A's relational sequence. In this behavioural sequence, user A first views the posts from user B, followed by a profile check of user B. Then, user A sends add friend request to user B as user B raised his/her interest. Afterwards, user A starts to send messages to user B and have further interactions e.g., give a gift. In the view of individual-level, relation *message* might be triggered by any relation user A performed before as the dotted lines with arrows indicating in Fig. 5.1(a). Nevertheless, this kind of simple and straightforward behavioural sequence is easy for a spammer to imitate. As shown in Fig. 5.1(b), spammer E tries to hide its identity by performing *add friend* and *give a gift* before *message*.

While spammers are able to imitate the individual-level patterns of normal users, spammers may not be able to imitate the union-level patterns of relational sequences. Fig. 5.2(a) gives our illustration by another example. The relation *message* is performed between User A and User B after different combinations of relations' being performed. Spammer E in Fig. 5.2(b) can imitate the simplest union-level to some extent, but, it is difficult to imitate complex union-level. Therefore, we will exploit the short-term dependency of user relational sequences to improve the performance of social spammer detection, both at the individual-level and at the union-level.

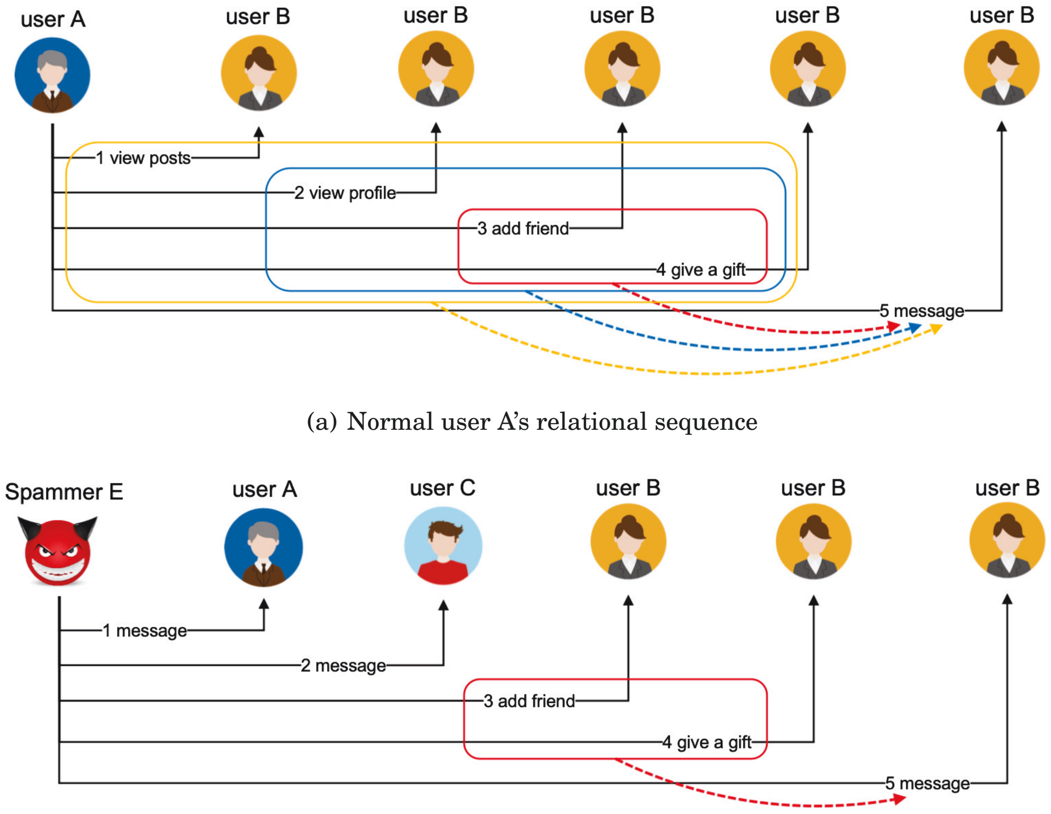
In addition to modelling short-term dependencies, we also exploit long-term dependencies between users' relational sequences. This is mainly because it is biased to consider only the short-term (only a few relations performed by the user), as the long-term (a sequence of relations throughout a day or even a whole week) may reveal the general behaviour and intentions of the user. Overall, our work exploits users' relational sequences according to their long-term dependencies as well as short-term dependencies from the individual and union levels.



(a) Normal user A's relational sequence

(b) Spammer E's relational sequence

Figure 5.1: Examples of individual-level dependency among the relational sequence. Normal users may be involved in one of the sequences of relations given in (a), (e.g., *add friend* first, and then send *message*). On the other hand, spammers may only imitate one or two relational behaviours of normal users (e.g., *add friend* first , then send *message*). However, for the reason that spammers always have their own malicious purposes, thus, they cannot completely imitate all the behavioural sequences of a normal user in (a).



(a) Normal user A's relational sequence

(b) Spammer E's relational sequence

Figure 5.2: Examples of union-level dependency among the relational sequence. Union-level dependencies can capture the collective impact between relational unions performed by users to some extent. For instance, the normal user is more likely to view profile, add friend, give a gift and sending messages together than view profile, add friend, give a gift or send messages individually. This combination of behavioural sequences makes it much more difficult for spammers to imitate the normal users.

5.1.2 Framework Overview

We note $\mathscr{U}$ as the set of N users, $u \in \mathscr{U}$ be a user. Note that we will also use $u_i, u_j \in \mathscr{U}$ to denote different users. Suppose there are M types of relations among users, denoted as $\mathscr{R} = \{r_1, \cdots, r_M\}$. Specifically, $M = 7$ in this work indicates seven relations including "add friend", "message", "give a gift", "view profile", "pet game", "meet-me game", "report abuse". We represent each user as a relational sequence $u = \langle s_1^u, \cdots, s_t^u, \cdots, s_T^u \rangle$, where $s_t^u \in \mathscr{R}$, $1 \leq t \leq T$ and the index t denotes the order in which one type of relation is used by u. The target of spammer detection is to estimate the likelihood that every user belongs to the spammer class, denoted as $P(y_u = \text{spammer}|u)$, where y_u is the label of u within the domain $\{\text{normal user}, \text{spammer}\}$. For simplicity, we let $\phi_u = P(y_u = \text{spammer}|u)$ and it is defined as:

$$\phi_u = \mathbf{F}(u,n) \cdot \sum_{r_m \in u} \boldsymbol{m}_{r_m}^{\top}, \tag{5.1}$$

where $\boldsymbol{m}_{r_m} \in \mathbb{R}^d$ is the embedding of relation, $r_m \in \mathscr{R}$, n is the selected most recent n relations for short-term modeling. $\mathbf{F}(u,n)$ is the output of proposed *MDM*.

The overall architecture of proposed *MDM* is made up of *User-relation Representation*, *Long-term Dependency Modeling* and *Short-term Dependency Modeling* as shown in Fig. 5.3. *MDM* first uses skip-gram with Recurrent Neural Network for representing user-relation into vector embedding. As shown in the bottom layer of Fig. 5.3, the input of this layer is one user's relational sequence $u = \langle s_1^u, \cdots, s_t^u, \cdots, s_T^u \rangle$. While the output is the d-dimensional latent vector of the input relational sequence.

Afterwards, *MDM* models long-term order constraint over the whole user-relation vector embeddings with a *Long-term Dependency Modeling* layer. The *Long-term Dependency Modeling* layer maps the whole user-relation vectors into a sequence of hidden vectors. With the output of *User-relation Representation* layer, *Long-term Dependency Modeling* generates the most recent n relations latent vectors as matrix $\mathbf{H}^u$. More importantly, we design one further step to input $\mathbf{H}^u$ to an attention layer, from which the short-term dependency is learned by *Short-term Dependency Modeling*, as shown in Fig. 5.3. This is similar to the most recent n items containing the potential intentions and preferences of the user, which can predict users' next behaviour. Finally, both long-term and short-term hidden information are extracted as embedding features and fed into a classification model for the spammer detection task. All the notations are listed in Table 5.1.

5.1.2.1 An Illustrative Example

We shall give a simple example in this section to facilitate our readers to better understand the overall process of our proposed *MDM*.

Let's input the user's relational sequence (e.g., $u = \langle 5,5,5,4,4,3,5,4,4 \rangle$) collected from Tagged.com. When inputting u to our *MDM* method, *User-relation Representation* layer outputs two components: One is a d-dimensional latent vector $\boldsymbol{e}_t^u$ $(1 \leq t \leq 9)$ for each item in the input sequence u, resulting in a $9 \times d$ matrix; Another one is the d-dimensional embeddings of 7 relations, $[\boldsymbol{m}_{r_1}, \boldsymbol{m}_{r_2}, \cdots, \boldsymbol{m}_{r_7}]^{\top}$. The first component $9 \times d$ matrix is then input to *Long-term Dependency Modeling* layer that outputs a sequence of hidden vectors, i.e., $[\boldsymbol{z}_1^u, \boldsymbol{z}_2^u, \cdots, \boldsymbol{z}_t^u, \cdots, \boldsymbol{z}_9^u]^{\top}$. Then, we define the most recent n relations latent vectors (i.e., $[\boldsymbol{z}_7^u, \boldsymbol{z}_8^u, \boldsymbol{z}_9^u]^{\top}$) as the matrix $\mathbf{H}^u \in \mathbb{R}^{n \times d}$. $\mathbf{H}^u$ is then input into the *Short-term Dependency Modeling* (individual-level) layer. We can obtain the final contextual embedding $\boldsymbol{v} \in \mathbb{R}^d$

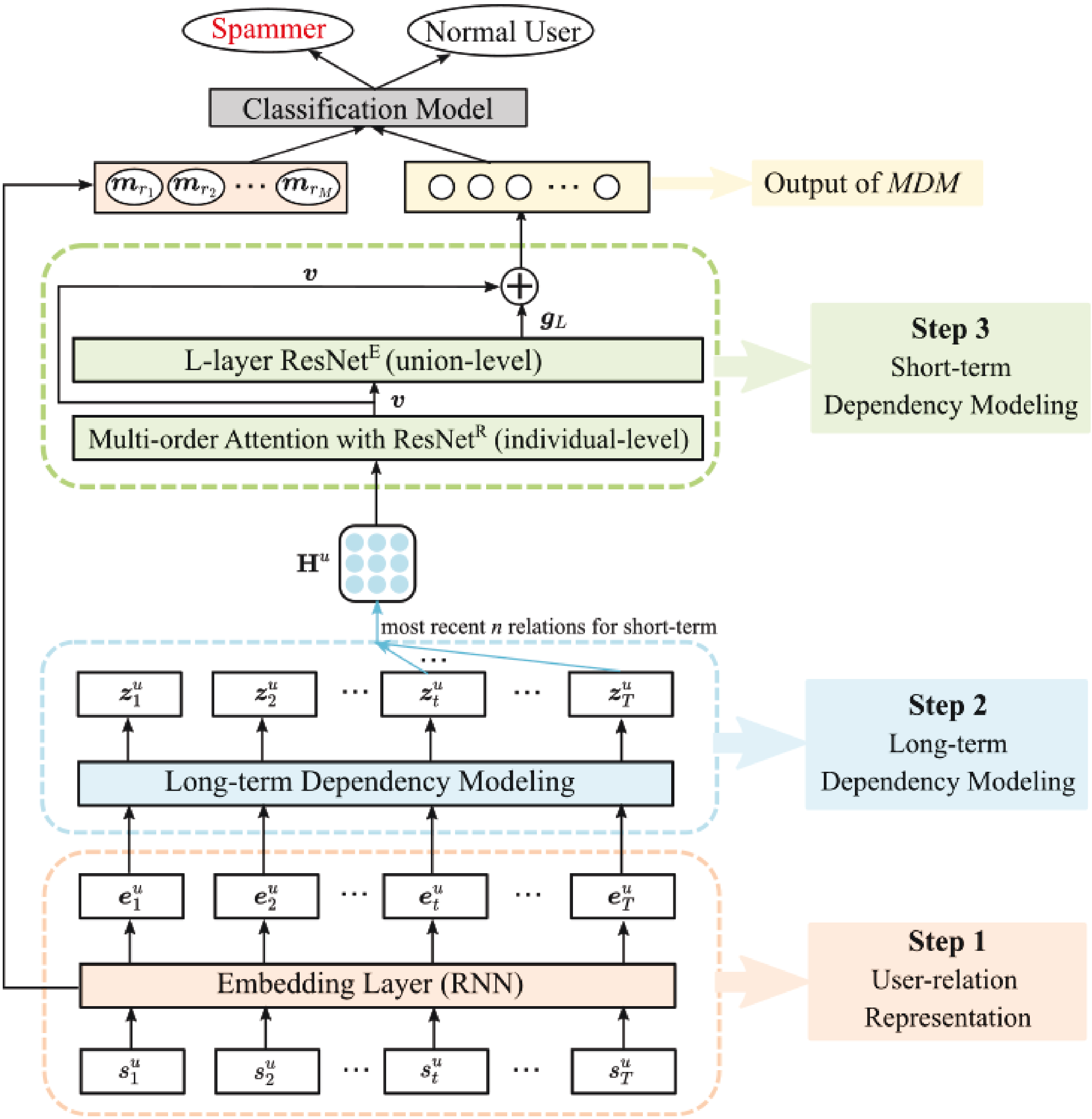

Spammer
Normal User
Classification Model
m_{r_1} m_{r_2} ⋯ m_{r_M}
Output of *MDM*
v
g_L
L-layer ResNet^E (union-level)
v
Multi-order Attention with ResNet^R (individual-level)
Step 3
Short-term
Dependency Modeling
$\mathbf{H}^u$
most recent n relations for short-term
z_1^u z_2^u ⋯ z_t^u ⋯ z_T^u
Long-term Dependency Modeling
Step 2
Long-term
Dependency Modeling
e_1^u e_2^u ⋯ e_t^u ⋯ e_T^u
Embedding Layer (RNN)
s_1^u s_2^u ⋯ s_t^u ⋯ s_T^u
Step 1
User-relation
Representation

Figure 5.3: Framework of Multi-level Dependency Model (*MDM*).

Table 5.1: Summary of notations

Notation	Description
$\mathscr{L},\mathscr{S},\mathscr{U}$	Set of normal users, spammers and all users respectively, $\mathscr{L}\cup\mathscr{S}=\mathscr{U}$
u	User's relational sequence, $u=\langle s_1^u,\cdots,s_t^u,\cdots,s_T^u\rangle$, $u\in\mathscr{U}$
$\mathscr{R}$	Set of relations, $\mathscr{R}=\{r_1,\cdots,r_m,\cdots r_M\}$
$\mathbf{F}(u,n)$	Output of *MDM*
$\boldsymbol{m}_{r_m}$	The embedding of relation, $\boldsymbol{m}_{r_m}\in\mathbb{R}^d$
$\boldsymbol{e}_t^u$	The user-relation representation for position t in u, $\boldsymbol{e}_t^u\in\mathbb{R}^d$
$\boldsymbol{z}_t^u$	Output of *Long-term Dependency Modeling*, $\boldsymbol{z}_t^u\in\mathbb{R}^d$
u_n	Set of most recent happened n relations, $u_n=\langle s_{T-1}^u,s_{T-2}^u,\cdots,s_{T-n}^u\rangle$
$\mathbf{H}^u$	The most recent n relations' outputs from *Long-term Dependency Modeling* layer
k, L	Numbers of layers for $ResNet^R$ and $ResNet^E$ respectively
$\mathbf{H}_1^u,\mathbf{H}_2^u,\cdots,\mathbf{H}_k^u$	Hidden status of $ResNet^R$
$\boldsymbol{v}_l$	High-order features for each layer of $ResNet^R$, $\boldsymbol{v}_l\in\mathbb{R}^d$ $(0\le l\le k)$
$\boldsymbol{h}_{i:}^k$	Corresponding i-th row of matrix $\mathbf{H}_k^u$
α_i^k	Weight scale for $\boldsymbol{v}_k$
$[\boldsymbol{v}_0,\boldsymbol{v}_1,\cdots,\boldsymbol{v}_k]^\top$	Set of aggregated high-order features
$\boldsymbol{v}$	Output of *Multi-order Attention with* $ResNet^R$ *(indiviual-level)* layer
β	Attention weight vector, $\beta\in\mathbb{R}^{k+1}$
$\boldsymbol{g}_L$	Output of *L-layer* $ResNet^E$ *(union-level)* layer
Θ	Parameters for optimizing, including: W_{LSTM} for long-term modeling; $\mathbf{W}_k,\boldsymbol{b}_k$ for $ResNet^R$; $\omega_1,\omega_2,c_1,c_2,\varphi_1,\varphi_2,b_1,b_2$ for attention model and $\mathbf{W}_L,\boldsymbol{b}_L$ for $ResNet^E$.

at individual-level and $\boldsymbol{g}_L\in\mathbb{R}^d$ at union-level. Finally, our *MDM* method outputs the concentration of $\boldsymbol{v}$, $\boldsymbol{g}_L$ and $[\boldsymbol{m}_{r_1},\boldsymbol{m}_{r_2},\cdots,\boldsymbol{m}_{r_7}]^\top$ as the learned embedding features for user u. This embedding can be further input into traditional classification methods to detect whether user u is spammer or not.

5.1.3 Model Formulation

In this section, we give the detailed discussion of the three components of *Multi-level Dependency Model* (*MDM*).

5.1.3.1 User-relation Representation

Before exploiting the hidden information behind the sequence of user relations, we need to model it first. To uncover the relational sequence features implicit in the relational sequence, an efficient representation needs to be found that learns high-quality user-relational vectors directly from the user's relational sequence. We utilize the user's relational sequence and apply skip-gram [51] with recurrent neural networks to generate the user-relation representation.

To be more specific, given a relation r_m $(1 \leq m \leq M)$ and a user's relational sequence $u = \langle s_1^u, \cdots, s_t^u, \cdots, s_T^u \rangle$, we denote the likelihood of $s_t^u = r_m$ as

$$P(r_m | \, t, u) = \frac{\exp(\varepsilon(r_m, s_t^u))}{\sum_{m'=1}^{M} \exp(\varepsilon(r_{m'}, s_t^u))}, \tag{5.2}$$

where $\varepsilon(r_m, s_t^u) = \boldsymbol{m}_{r_m} \cdot \boldsymbol{e}_t^{u^\top}$, $\boldsymbol{m}_{r_m} \in \mathbb{R}^d$ $(1 \leq m \leq M)$ is the latent vector for each relation in $\mathscr{R}$, and $\boldsymbol{e}_t^u \in \mathbb{R}^d$ is the user-relation representation for position t $(1 \leq t \leq T)$ in u. To obtain the embedding $\boldsymbol{m}_{r_m}$ and $\boldsymbol{e}_t^u$, the *Embedding layer* implemented by RNN optimize the objective function as follows:

$$\max_{\boldsymbol{m}_{r_m}, \boldsymbol{e}_t^u} \sum_{m=1}^{M} \sum_{t=1}^{T} \log P(r_m | \, t, u) = \max_{\boldsymbol{m}_{r_m}, \boldsymbol{e}_t^u} \sum_{m=1}^{M} \sum_{t=1}^{T} \log \frac{\exp[\boldsymbol{m}_{r_m} \cdot \boldsymbol{e}_t^{u^\top}]}{\sum_{m'=1}^{M} \exp[\boldsymbol{m}_{r_{m'}} \cdot \boldsymbol{e}_t^{u^\top}]}, \tag{5.3}$$

where T is the length of relational sequence u, and M is the number of relations.

5.1.3.2 Long-term Dependency Modeling

We shall take advantage of the standard LSTM [36] over the whole relational sequence as shown in Fig. 5.3 to model the long-term dependence of users' relational sequences in multi-relational social networks. For each $u \in \mathscr{U}$ we can get a user-relation representation from Eq. (5.3), denoted as $\{\boldsymbol{e}_1^u, \cdots, \boldsymbol{e}_t^u, \cdots, \boldsymbol{e}_T^u\}$, where $\boldsymbol{e}_t^u$ denotes the d-dimensional latent vector of position t. Given the user-relation representation for user u from the last *User-relation Representation* layer, we can obtain a sequence of hidden vectors $\{\boldsymbol{z}_1^u, \cdots, \boldsymbol{z}_t^u, \cdots, \boldsymbol{z}_T^u\}$ by recurrently inputting $\boldsymbol{e}_t^u$ $(1 \leq t \leq T)$ into LSTM, i.e.,

$$\boldsymbol{z}_t^u = \text{LSTM}(\boldsymbol{e}_t^u, \boldsymbol{z}_{t-1}^u, W_{LSTM}), \tag{5.4}$$

where LSTM is the output function of Long Short-Term Memory, W_{LSTM} contains the weight parameters and we set $\boldsymbol{z}_0^u = \boldsymbol{0}$.

Through this stage, the *Long-term Dependency Modeling* in Fig. 5.3 outputs a sequence $\{\boldsymbol{z}_1^u, \cdots, \boldsymbol{z}_t^u, \cdots, \boldsymbol{z}_T^u\}$ for the next multi-order attentive relation modeling stage. Since only capturing long-term dependency is not sufficient, as it neglects the importance of adjacent relation within the sequence. In next section, we will illustrate how to augment long-term dependency with short-term dependency in terms of individual-level and union-level.

5.1.3.3 Short-term Dependency Modeling

This section will give the discussion on how to extend general user's embedding with short-term dependency over a small set of the most recent n happened relations, which can be denoted as $u_n = \langle s^u_{T-1}, s^u_{T-2}, \cdots, s^u_{T-n} \rangle$.

As shown in Fig. 5.3, the *MDM* applies $ResNet$ to learn high-order non-linear interactions among the short-term dependency of u_n. *MDM* instantiates two residual networks with a fully connected multi-layer perceptron, i.e., k-layer $ResNet^R$ for individual-level and L-layer $ResNet^E$ for union-level, respectively.

Individual-level It can be seen from Fig. 5.1 that the individual-level dependency of users' relational sequences might leads to the different subsequent relations for spammers and normal users respectively. This means that the exploiting of individual-level dependencies certainly facilitates the detection of spammers in multi-relational social networks. Hence, we adopt the *Long-term Dependency Modeling* layer's most recent n relations' outputs, denoted as $\mathbf{H}^u \in \mathbb{R}^{n\times d}$:

$$\mathbf{H}^u = \begin{bmatrix} \boldsymbol{z}^u_{T-1} \\ \boldsymbol{z}^u_{T-2} \\ \vdots \\ \boldsymbol{z}^u_{T-n} \end{bmatrix}, \tag{5.5}$$

where $\boldsymbol{z}^u_{T-1}$ is produced by Eq. (5.4) with $t = T-1$. And we use $\mathbf{H}^u$ as the input of *Multi-order Attention with* $ResNet^R$ *(individual-level)* layer shown in Fig. 5.3. Then, we propose an attention mechanism to aggregate high-order features of individual level dependency as show in Fig. 5.4.

With the input embedding $\mathbf{H}^u$, the *Multi-order Attention with* $ResNet^R$ *(individual-level)* layer is instantiated with a k-layer residual network $ResNet^R$, i.e., $ResNet(k, \mathbf{H}^u)$. Then, we can obtain the output of a sequence of hidden status as

$$\begin{aligned} \mathbf{H}^u_1 &= \mathrm{ReLU}(\mathbf{H}^u\mathbf{W}_1 + \mathbf{b}_1 + \mathbf{H}^u) \\ \mathbf{H}^u_2 &= \mathrm{ReLU}(\mathbf{H}^u_1\mathbf{W}_2 + \mathbf{b}_2 + \mathbf{H}^u_1) \\ &\cdots\cdots \\ \mathbf{H}^u_k &= \mathrm{ReLU}(\mathbf{H}^u_{k-1}\mathbf{W}_k + \mathbf{b}_k + \mathbf{H}^u_{k-1}), \end{aligned} \tag{5.6}$$

where ReLU is the activation function for *rectifier linear unit*, $\mathbf{H}^u_k \in \mathbb{R}^{n\times d}$ is the high-order features generated at k-th layer of $ResNet^R$, k denotes the maximum number of residual layers. $\mathbf{W}_k \in \mathbb{R}^{d\times d}$ and $\mathbf{b}_k \in \mathbb{R}^d$ denote weight matrix and bias vector, respectively.

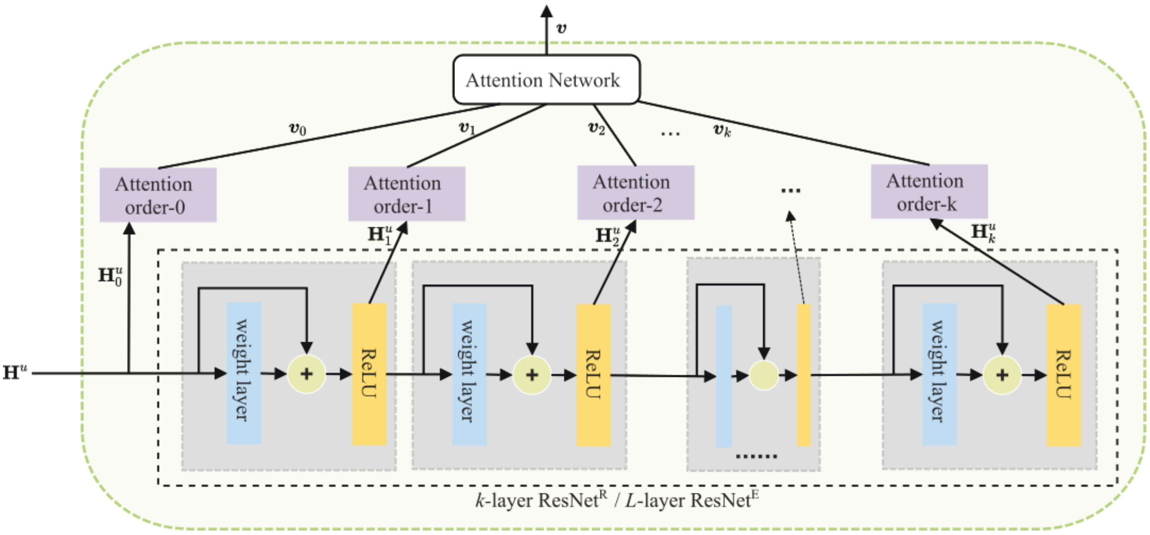


Figure 5.4: Multi-order Attention Network.

The sequence of hidden status from $ResNet^R$, i.e., $\{\mathbf{H}^u_1, \mathbf{H}^u_2, \cdots, \mathbf{H}^u_k\}$, can capture potential high-order interactions between partial fields in relation embedding, which helps us to discriminate the significance of relations in different levels with respect to a given relational sequence. Besides the high-order features, we also keep the raw embedding $\mathbf{H}^u$, resulting a set of extended encoded features $\{\mathbf{H}^u_0, \mathbf{H}^u_1, \cdots, \mathbf{H}^u_k\}$, where $\mathbf{H}^u_0 = \mathbf{H}^u$.

To aggregate $\{\mathbf{H}^u_0, \mathbf{H}^u_1, \cdots, \mathbf{H}^u_k\}$, we use a soft attention model. We denote $\boldsymbol{v}_l \in \mathbb{R}^d$ $(0 \le l \le k)$ as the contextual embedding for each layer, which can be generated by the soft attention model [92] as follows.

$$\begin{aligned} \boldsymbol{v}_0 &= \sum_{i=1}^{n} \alpha_i^0 \cdot \boldsymbol{h}^0_{i:} \\ \boldsymbol{v}_1 &= \sum_{i=1}^{n} \alpha_i^1 \cdot \boldsymbol{h}^1_{i:} \\ &\cdots\cdots \\ \boldsymbol{v}_k &= \sum_{i=1}^{n} \alpha_i^k \cdot \boldsymbol{h}^k_{i:}, \end{aligned} \tag{5.7}$$

where $\boldsymbol{h}^k_{i:}$ $(1 \le i \le n)$ is the corresponding i-th row of matrix $\mathbf{H}^u_k$. And weight scale α^k_i is normalized by a softmax layer on the attention scores, $\sum_{i=1}^{n} \alpha^k_i = 1$. We utilize a network with two-layers to calculate the attention scores with Eq. (5.8).

$$\begin{aligned} \alpha_i^k &= \omega_1 \tanh(\omega_2 \boldsymbol{h}_{i:}^k + c_1) + c_2 \\ \alpha_i^k &= \frac{\exp(\alpha_i^k)}{\sum_{i'}^n}, \end{aligned} \tag{5.8}$$

where ω_1, ω_2 are the shared weight matrices for attention layer. Then, the final contextual embedding of short-term dependency with the most recent n relations is

$$\begin{aligned} \boldsymbol{v} &= \beta[\boldsymbol{v}_0, \boldsymbol{v}_1, \cdots, \boldsymbol{v}_k]^\top \\ \beta &= \mathrm{softmax}(\varphi_1 \tanh(\varphi_2 [\boldsymbol{v}_0, \boldsymbol{v}_1, \cdots, \boldsymbol{v}_k]^\top + b_1) + b_2), \end{aligned} \tag{5.9}$$

where φ_1, φ_2 are the weight matrices for attention layer and $\beta \in \mathbb{R}^{k+1}$ is the attention weight vector. According to Eq. (5.7) and Eq. (5.9), we can obtain the final contextual embedding $\boldsymbol{v}$ on individual-level as Eq. (5.10)

$$\boldsymbol{v} = \beta \cdot \sum_{l=0}^{k} \sum_{i=1}^{n} \alpha_i^l \cdot \boldsymbol{h}_{i:}^l. \tag{5.10}$$

Union-level Additionally, we also exploit the union-level of short-term dependencies between sequences of user relations. As shown in Fig. 5.2, while spammers may be able to imitate some individual-level patterns from normal users' relational sequences, it is difficult for them to imitate complex combinations of normal relations, i.e. union-level dependencies. Hence, we believe that individual-level and union-level dependencies can complement each other to address users' short-term relational sequences.

Union-level dependencies can be understood conceptually by estimating the probability of association rules $X \to Y$, where X is one user's most recent n relations and Y is the subsequent relation to be performed. Particularly, we combine an attention network and a residual network to represent the set of relations X. To be more specific, We use embedded features from the individual-level as input to a multilayer perceptron with a residual structure, instantiated as $ResNet^E$. With the input $\boldsymbol{v}$ given by Eq. (5.10), $ResNet^E$ outputs the representation of union-level dependencies as belows.

$$\begin{aligned} \boldsymbol{g}_1 &= \mathrm{ReLU}(\boldsymbol{v}\mathbf{W}_1 + \mathbf{b}_1 + \boldsymbol{v}) \\ \boldsymbol{g}_2 &= \mathrm{ReLU}(\boldsymbol{g}_1\mathbf{W}_2 + \mathbf{b}_2 + \boldsymbol{g}_1) \\ &\cdots\cdots \\ \boldsymbol{g}_L &= \mathrm{ReLU}(\boldsymbol{g}_{L-1}\mathbf{W}_L + \mathbf{b}_L + \boldsymbol{g}_{L-1}), \end{aligned} \tag{5.11}$$

where $\mathbf{W}_L \in \mathbb{R}^{d \times d}$ and $\mathbf{b}_L \in \mathbb{R}^d$ denote weight matrix and bias vector, respectively.

5.1.3.4 Objective Function

In order to concatenate individual-level features with union-level features, we formulate $\mathbf{F}(u,n)$ in Eq. (5.1) for user with relational sequence $u \in \mathscr{U}$ as:

$$\mathbf{F}(u,n) = \boldsymbol{v} + \boldsymbol{g}_L, \tag{5.12}$$

where $\boldsymbol{v}$ is given by Eq. (5.10) and $\boldsymbol{g}_L$ is given by Eq. (5.11). $\mathbf{F}(u,n)$ is then the embedding of context information integrated at both individual-level and union-level. Afterwards, the predictive model Eq. (5.1) can be extended as:

$$\phi_u = \mathbf{F}(u,n) \cdot \sum_{r_m \in u} \boldsymbol{m}_{r_m}^{\top} = (\boldsymbol{v} + \boldsymbol{g}_L) \cdot \sum_{r_m \in u} \boldsymbol{m}_{r_m}^{\top}. \tag{5.13}$$

The spammers will have relative larger values of Eq. (5.13) than normal users, i.e., $\phi_{u_i} > \phi_{u_j}$.

The predictive model ϕ_u can be fitted by optimizing the underlying parameters Θ that is from W_{LSTM} in Eq. (5.4), $ResNet^R$ in Eq. (5.6), $ResNet^E$ in Eq. (5.11) and soft attention model in Eq. (5.10). Let $\mathscr{S}$ represents the set of spammers' relational sequences and $\mathscr{L}$ denotes the set of normal users' relational sequences, i.e., $\mathscr{U} = \mathscr{S} \cup \mathscr{L}$. With the inputs of user-relation sequences $u = \langle s_1^u, \cdots, s_t^u, \cdots, s_T^u \rangle$, Θ can be obtained by optimizing the following objective function:

$$\arg\min_{\Theta} \sum_{u_i \in \mathscr{S}} \sum_{u_j \in \mathscr{L}} -I(\phi_{u_i}, \phi_{u_j}) + \frac{\lambda}{2} ||\Theta||_F^2, \tag{5.14}$$

where $I(\cdot,\cdot)$ is an indicator function that equals 1 for $\phi_{u_i} > \phi_{u_j}$, otherwise equals 0, $||\cdot||_F^2$ represents *Frobenius* norm weighted with a hyper-parameter λ. We use Adam optimizer [42] to optimize the objective function (5.14) and produce the optimal Θ. The pseudocode of leveraging *MDM* for social spammer detection is presented in Alg. 1[1], where line 4 refers to step 1 "User-relation Representation" in Fig. 5.3, line 5 refers to step 2 "Long-term Dependency Modeling" in Fig. 5.3, line 6-9 refer to step 3 "Multi-order Attention with $ResNet^R$ (individual-level)" in Fig. 5.3, line 10 refers to step 3 "L-layer $ResNet^E$ (union-level)" in Fig. 5.3, line 11 is a concatenate within step 3 "Short-term Dependency Modeling" in Fig. 5.3.

5.2 Experiment and Analysis

To evaluate the effectiveness of the proposed *Multi-level Dependency Model* (*MDM*), experiments were conducted on a large real-world dataset from *Tagged.com*, which is

[1] The details of step 1, 2 and 3 can be found in Fig. 5.3

Algorithm 1 Algorithm of Leveraging *MDM* for Social Spammer Detection

```
Input: Labeled set $\mathscr{U} = \mathscr{S} \cup \mathscr{L}$ includes all users' relational sequences, and each user's
    relational sequence is $u = \langle s^u_1, \cdots, s^u_t, \cdots, s^u_T \rangle$; Number of relations for short-term $n$;
    Embedding size $d$; Number of layers $k$.
Output: Users' label: {Spammer, Normal user}.
procedure MDM($\mathscr{U}, n, d, k$)
    repeat
    for each $u \in \mathscr{U}$ do
        Compute $\boldsymbol{e}^u_t$ and $\boldsymbol{m}_{r_m}$ via Eq. (5.3);                         ▷ Step 1
        Compute the long term embedding $\boldsymbol{z}^u_1, \cdots, \boldsymbol{z}^u_T$ by Eq. (5.4);            ▷ Step 2
        Compute $\mathbf{H}^u$ via Eq. (5.5);                          ▷ Step 3, individual-level
        Compute $\{\mathbf{H}^u_0, \mathbf{H}^u_1, \cdots, \mathbf{H}^u_k\}$ via Eq. (5.6);                  ▷ $\mathbf{H}^u_0 = \mathbf{H}^u$
        Compute $[\boldsymbol{v}_0, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k]^\top$ by $\{\mathbf{H}^u_0, \mathbf{H}^u_1, \cdots, \mathbf{H}^u_k\}$ via Eq. (5.7);
        Compute the embedding of individual level $\boldsymbol{v}$ via Eq. (5.9);
        Compute the embedding of union level $\boldsymbol{g}_L$ by Eq. (5.11);        ▷ union-level
        Compute $\mathbf{F}(u, n)$ via Eq. (5.12);                        ▷ Concatenate
        Update the parameter set $\Theta$ in (5.14) by Adam algorithm;
    end for
    until converge.
end procedure
procedure PREDICTION($MDM(\cdot)$, $\mathscr{U}$)
    Take the output of MDM, $\mathbf{F}(u, n)$, as the feature for each user $u$;
    Use classification model to classifier spammers and normal users;
end procedure
```

described in Chapter 3. In our experiments we compare with several state-of-the-art methods for spammer detection in multi-relational social networks, including graph-based and sequence-based methods. Our algorithms were implemented in TensorFlow, and the experiments were conducted on a computer with 28 CPU cores and 256 GB of memory.

5.2.1 Experiment Setup

5.2.1.1 Dataset

To keep consistence, we take the same dataset as introduced in section 4.2.1. We extracted all interactions from "*day 0*" to obtain a dataset containing 85M interactions between 4M users, i.e. the average length of a user's relational sequence is 21. 182K of the 4M users are labelled as spammers (4.45%). Statistics of the dataset is shown in Table 5.2.

Table 5.2: Statistics of *Tagged.com* dataset

Dataset	*Tagged.com*
#user	4,111,179
#spammer	182,939
#normal user	3,928,240
#interactions	85,470,637
AVG length of relational sequence	21

5.2.1.2 Evaluation Metrics

As the ground-truth label of each user is provided by the data set, we use standard indicators (P-R-F), including precision (P), recall (R), and F-measure (F) to evaluate the effectiveness of the proposed *MDM* model, as shown in Eq. (5.15), which also keep the consistence with Chapter 4.

$$R = \frac{TP}{TP+FN},\ P = \frac{TP}{TP+FP},\ F = \frac{2P\cdot R}{P+R}, \tag{5.15}$$

Since the key concerns is to evaluating the quality of features extracted from multi-relational data rather than new classification algorithms, we choose two simple classic supervised models: Logistic Regression (*LR*) [5] and XGBoost (*XGB*) [18]. We choose XGB here instead of Gaussian Naive Bayes (GNB) in section 4.2.3 due to the consideration of feature structure. XGB achieves better results than GNB for the features extracted by MDM in this section. In order to avoid overfitting problems, we employ 10-fold cross-validation to select the most appropriate parameters for logistic regression and XGBoost. For logistic regression, we assign an l_2 parametric penalty and a default strength of $C = 1$. We also set the tolerance of the stopping criterion to 0.0001 and the maximum number of iterations to 50. We use XGBoost to implement a tree-based component for all methods, where the number of trees is 200 and the maximum depth of the tree is 5.

5.2.1.3 Baselines

We choose several state-of-the-art graph-based and sequence-based methods as the baselines.

More concretely, graph-based features are extracted by converting relations into a directed graph $\mathscr{G}$, where the vertices $\mathscr{V}$ indicate the user and the edges $\mathscr{E}$ indicate the relations that performed by the user. There are 7 types of relations in *Tagged.com* data set, total 7 graphs are generated: $\{\mathscr{G}_1,\ldots,\mathscr{G}_7\}$. Then, for each graph we use *Graphlab*

Create[2] to extract graph-based features, including *Page Rank* [59], *Graph Coloring* [39], *Weakly Connected Components* [62], *k-core* [8], *Degree* [27], and *Triangle Count* [70]. This converts the directed graph into a numerical or categorical feature matrix for each relation. Therefore a total of 7 × 8 graph-based features are generated, which can be thought of as a 56-dimensional vector.

As the source code of generating sequential k-grams features was provided by Fakhraei et al. [27], we choose sequential k-grams as a baseline to compare with our proposed model. To be more specific, sequential k-grams are designed to construct sequences from short sequential segments of k consecutive actions. The sequence can be represented as a k-gram of frequency vectors. To keep the feature space computationally tractable, the baseline method [27] sets $k = 2$, e.g., sequences 1-1 or 2-1. That is, there are 49 sequences of 7 relations, denoting sequences of dimension 49. For a particular user with actions 1-1-2-3, the corresponding 49-dimensional sequence vector is $[1,1,0,0,0,0,0,0,0,0,0,1,0,\cdots,0]$.

5.2.2 Experimental Results

We use the most recent n relations for short-term dependency modeling, where n is chosen from $\{2,4,6,8\}$. The embedding size d in our *MDM* is chosen from $\{8,16,32\}$. We also try different number of hidden layers of $ResNet^R$ and $ResNet^E$ from $\{2,4\}$, as we find that 4 layers are enough to ensure competitive results for both $ResNet^R$ and $ResNet^E$.

After getting all the features from baseline methods and *MDM*, we split train and test dataset with 10 different random seeds for evaluation on *LR* and *XGB* classifiers. First, we compare our *MDM* with them separately. Then, we combine the baseline methods together to show the effectiveness of our proposed model.

5.2.2.1 Overall Comparison with Baselines

Table 5.3 gives the experimental results of the comparison between *MDM* and baselines. Not surprisingly, higher recall rates occur with lower precision. It implies that normal users may be incorrectly identified as spammers to ensure that more spammers are detected. Under such circumstances, recall and precision are not sufficient to validate the effectiveness of our method. We further introduce F-measure to evaluate our performance by calculating a summed average of precision and recall. From the experimental compar-

[2] https://turi.com/

ison, it can be seen that *MDM* shows a significant performance advantage over both *LR* and *XGB* classifiers in the F-measure metric. This means that we can catch spammers more accurately with minimal harm to the normal user. More excitingly, the accuracy of *MDM* is consistently the highest accuracy with the best performance parameters ($d = 32, n = 6, k = 4$), proving that the proposed features can reveal most spammers at the cost of a slightly lower recall. In terms of recall, although the continuous k-gram features occupy the highest position, they perform the worst in terms of precision as price, which means that they treat more users as spammers and have a high impact on the normal users.

Table 5.3: Performance comparison with baselines (the best result of each metric is bold)

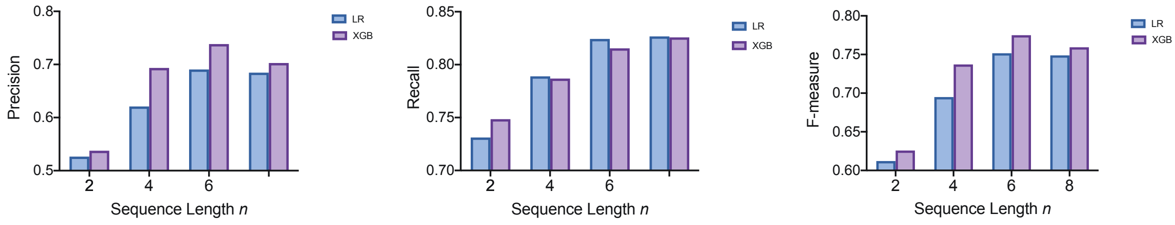
	Logistic Regression (*LR*)			XGBoost (*XGB*)		
Methods	Precision	Recall	F-measure	Precision	Recall	F-measure
Graph-based [13]	0.5576	0.6937	0.6182	0.6378	0.6712	0.6541
Sequential k-gram [27]	0.5217	**0.8620**	0.6500	0.5268	**0.9221**	0.6705
Graph-based+Sequential k-gram	0.6116	0.8600	0.7148	0.6253	0.9127	0.7421
MDM	**0.6909**	0.8243	**0.7516**	**0.7385**	0.8154	**0.7750**

5.2.2.2 Effect of Parameters within MDM

We carried out a further evaluation of the performance of *MDM* in terms of parameter settings. We first varied the sequence length in short-term information modelling. We set the sequence length n from $\{2,4,6,8,\}$. Fig. 5.5 shows the results of the comparison for the different settings. It shows that $n = 6$ promotes the best performance when the other parameter settings are equal. We speculate that this is because the behaviour of the 6 steps better summarises the user intent in *Tagged.com*.

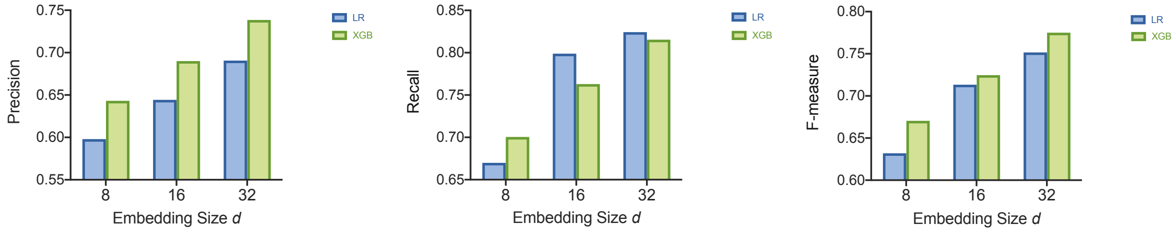
We then vary the embedding size d to analyse the performance of *MDM*, and we set d from $\{8,16,32\}$. Fig. 5.6 shows the performance of the embedding features for each size in terms of precision, recall and F-measure, respectively. Clearly, the ratio of these three metrics increases with increasing dimension, suggesting that more spammers will be found when the embedding dimension of our *MDM* is increased, and that it will be more accurate.

Overall, the experimental results show that MDM achieves the most efficient performance when the embedding feature dimension is 32. Due to computational space constraints, we only increased the embedding size to 32. However, the number of embedding sizes depends on the dataset. We propose a reasonable conjecture that the number of embedded features should be increased together with the number of relation types, as more types of relations imply more sophisticated interactions.



(a) Precision (b) Recall (c) F-measure

Figure 5.5: Performances of MDM under different sequence lengths n.



(a) Precision (b) Recall (c) F-measure

Figure 5.6: Performances of MDM under different embedding sizes d.

5.2.2.3 Components Influence of MDM

MDM is made up of three components as shown in Fig. 5.3, i.e. *User-relation Representation*, *Long-term Dependency Modeling*, and *Short-term Dependency modeling*, where the last component consists of individual-level and union-level. We set up different combinations of components for evaluation in order to analyse the impact of different components on the overall detection performance. The comparison results are shown in Table 5.4.

It is evident from the table that although *User-relation Representation* achieves the highest recall on both *LR* and *XGB*, it has the lowest precision and F-measure scores. The addition of the *Long-term Dependency Modeling* layer increased precision, decreased recall slightly, and increased the overall F-measure score for both *LR* and *XGB*. In other words, the *Long-term Dependency Modeling* layer can help improve our estimated social spammer detection performance.

When *Short-term Dependency Modeling (individual-level)* is added, we can see from the table that there is a significant improvement in both precision and F-measure. This indicates that the short-term dependencies within the relational sequence largely enable the user-hidden sequential information modelling to be improved. Subsequently, *MDM* consists of all three layers and obtains the best performance in terms of precision and F-measure with the best performance parameters ($d = 32, n = 6, k = 4$). In other words, with *MDM* we can correctly detect more spammers without hurting the normal user.

Table 5.4: Performance comparison on different components in MDM (+ represents adding a layer to the last row, and the best result of each metric is bold)

	Logistic RegressionD (*LR*)			XGBoost (*XGB*)		
Components	Precision	Recall	F-measure	Precision	Recall	F-measure
User-relation Representation	0.5399	**0.8496**	0.6602	0.5778	**0.8722**	0.6951
+ Long-term	0.5687	0.8467	0.6804	0.5937	0.8718	0.7064
+ Individual-level	0.6314	0.8477	0.7237	0.6659	0.8523	0.7477
MDM	**0.6909**	0.8243	**0.7516**	**0.7385**	0.8154	**0.7750**

5.2.3 Discussion

We studied behavioural sequences in a multi-relational social network (i.e. *Tagged.com*) to detect unknown spammers. Some interesting spamming behaviours have been identified from our experiments. Our outcomes show that users with sequences $\langle 5,5,5,5,5,5,5,5\rangle$, $\langle 5,5,5,5,5,4\rangle$, $\langle 4,4,3,5,4,4\rangle$ are easily detected as spammers in our proposed *MDM*. From Section 3.2 we get the speculation that relation 5 ought to be "*Pet Game*", which means that the sequence $\langle 5,5,5,5,5,5,5\rangle$ indicates that the user has been playing this "*Pet Game*". A sequence of such behaviour by a user is identified as a spammer. This is because *Tagged.com* has a reward mechanism for playing "*Pet Game*". The spammers always get more rewards by playing "*Pet Game*" in order to be seen/contacted by more users, and they will appear in the star list. Other than this particular sequence, we observe that users who repeated individual relations and occasionally switched between one or two relations to hide their behaviour were more probably spammers.

5.3 Summary

A novel *Multi-level Dependency Model (MDM)* is proposed in this work to leverage the deeper complementary information behind users' relational sequences. *MDM* utilises the user's behaviour in terms of *long-term* and *short-term* dependencies. Notably, *short-term* dependencies can be well utilised at both the individual and union levels. Consequently, *MDM* is able to expose deeper information behind relational sequences, thereby improving the accuracy of identifying anomalous behaviour. We have conducted extensive experiments on a real-world dataset from *Tagged.com* and verified that *MDM* outperforms other baselines significantly.

Part II

Part II   Rumour Analysis

Chapter 6

Rumour Impact Analysis on Social Media

6.1 Introduction

Social media and the rumours circulating on it have grown exponentially in the last decade. Not only does the widespread dissemination of rumours undermine truthful information, it can also mislead public opinion. The detection and monitoring of rumours has therefore become crucial to maintaining a healthy social media environment.

Even so, the concept of rumour is not new and has been around for a long time, since the invention of the printing press in 1439. Whilst the concept of rumour has a long history, there is no agreed definition of the term “Rumour”. There are two schools of thought on the definition of rumour, as suggested by recent publications in the research literature. For the first school, some recent work has incorrectly defined a rumour as a type of information that is considered false [14, 47], which is conflated with fake news. For the second school of thought, which makes up the bulk of the literature, they define a rumour as “an unverified and instrumentally relevant statement of information in circulation” [7, 24, 26]. In this work, we have adopted a definition of rumour consistent with the second school, which is also consistent with the definitions given by the major dictionaries. The Oxford English Dictionary defines a rumour as “a currently circulating story or report of uncertain or doubtful truth”[1]; the Merriam Webster Dictionary defines it as “a statement or report current without known authority for its truth”[2].

[1] https://en.oxforddictionaries.com/definition/rumour.

[2] http://www.merriam-webster.com/dictionary/rumor.

Definition 6.1. Rumour is an item of information that are unverified at the time of posting, and may turn out to be true, or partly or entirely false; alternatively, it may also remain unresolved.

Based on the unsubstantiated nature of the rumours, verifying their authenticity is vital. Nevertheless, social media platforms are increasingly being used for information and news gathering and their uncontrolled nature has led to a massive rumour emergency. As a result, authenticating every rumour on social media is not practicable. In addition, from a refutation viewpoint, it is not cost effective to pay more attention to rumours that receive little or no attention.

Faced with this problem, the first step we need to take is to screen rumours before proceeding to the next step. In the case of rumours that have a high impact on social media, we need to be more careful to verify their authenticity and to work to refute them accordingly. Conversely, there is probably no need for us to specifically check their authenticity. A further important research question therefore arises. How should we screen rumours when they appear? It is common for rumours with a widespread tendency to spread to have their own narrative style to grab the attention of the audience, e.g. scientific narrative style, celebrity effect style, etc. In some rumours, for instance, the data and images are factually based and each figure is labelled with a reference or a sentence such as “based on relevant research ......”. And this is how data is used to pull the wool over the eyes of the public. If you dig deeper, what you will find is that these so-called relevant studies are baseless. So, do we need to take the time to research in depth? A different kind of rumour uses popular keywords to draw the focus of the public for the purpose of getting the word out on social media, such as the celebrity effect. For example, in 2017 a piece of news posted on *Toutiao.com*, a Chinese news platform, said that Lu Han and Guan Xiaotong, two superstars with huge followings in China, would be performing together at the 2018 CCTV Spring Festival Gala, accompanied by photos of the rehearsals. The news was confirmed to be false, however it received a large number of shares and retweets on social media. This is a common use of the celebrity effect in disseminating rumours through the public. The good thing is that the news did not cause any serious repercussions but only let down some fans. But if rumours were spread using the celebrity effect, which could cause damage to public property or even risk to life, then there could be a very serious impact on society. Our proposed *RISM* model will provide a higher impact score for such rumours in order to gain people’s awareness. When faced with a rumour with a high score, we would need to be cautious.

The majority of researchers are in the field of rumour detection. On the basis of their work, two further challenges we aim to address are as follows.

1. How to give the definition of the impact of rumours on social media?

2. How can we predict the potential impact of a rumour early on in its life?

Therefore, we propose a novel prediction model *RISM* to learn the impact of rumours on social media [90]. Our main contributions are summarized as follows.

- While related literature is limited, we provide a novel measurement on the impact of rumours.

- A content-based model *RISM* is proposed, which can detect the impactful rumours before being spread.

- We conduct extensive experiments on real-world datasets to demonstrate the effectiveness of *RISM* model.

6.2 Preliminaries

6.2.1 Problem Statement

Let D be a rumour dataset, consisting of N rumour news $\{d_i\}_{i=1}^{N}$, while each news $d_i = \left\{w_1^i, ..., w_{P_i}^i\right\}$ contains P_i words. Let $H_i = \{h_j\}_{j=1}^{k}$ be a set of k comments related to the rumour news d_i, where each comment $h_j = \left\{w_1^j, ..., w_{Q_j}^j\right\}$ contains Q_j words. We aim to learn a rank list RI based on all sentences in $\{d_i\}_{i=1}^{N}$. Rumour's impact score represents the degree of negative effects caused by rumours. In other words, if a rumour news d_i is predicted to be a higher impact rumour, then, the government or some official institutions need to take measures to refute this rumour officially.

6.2.2 Measuring Rumour Impact

The rumour intensity formula was first proposed by American sociologists G.W. Allport and L. Postman in 1947 [7]:

$$R = I * A \tag{6.1}$$

where R represents the impact of the rumour, I represents the importance of the information mentioned in the rumour and A represents the ambiguity of the rumour.

Dutch scholar Chorus believes that the intensity of rumour is not only related to events, but also includes human factors. Thus, he introduced the concept of audience judgment ability in 1953 [20]. Chorus believes that audience judgment should include personally relevant knowledge, observation and moral cultivation, which are negatively correlated with rumour circulation. In other words, the richer the individual's knowledge is, the stronger the observation is, and the higher the moral cultivation is, the more resistant the spread of rumours is. Therefore, he developed the rumour intensity formula as below.

$$R = I * A/C \tag{6.2}$$

where C reflects the public's attitude towards the rumour.

The rumour intensity formula has been further developed by some researchers recently. Through the analysis of public emergencies, Wang [78] proposed his rumour intensity equation as below.

$$R = I * A * J * E \tag{6.3a}$$

$$E = c * s * \frac{1}{o}(s > 1, 0 < o < 1, c > 1) \tag{6.3b}$$

where I and A have the same representations as in Eq. (6.1) and Eq. (6.2), J represents the public critical ability and E refers to the environmental index. The new variable E includes the communication environment index c (communication) and the political environment index. Political environment index is composed of the political stimulus index s (stimulate) and political transparency o (open-politics). From the practical application point of view, the "Political Environment Index" and the "Communication Environment Index" have no specific measurement standards to give them corresponding values, which weakens the operability of the formula to some extent. Furthermore, Hou [67] improved the rumour intensity of Eq. (6.4) based on the dataset from Weibo (i.e., a popular social media in China). Hou claimed that rumour intensity has some relationship with the identity of the publisher.

$$R = I * A * (V + f) * \frac{1}{c + w} \tag{6.4}$$

where V denotes the identity of users, f refers to the number of fans, c represents the publics' critical ability and w represents the publics' willingness. However, Hou didn't propose standard measurement for the identity V and public willingness w, which makes Eq. (6.4) impractical.

Some researchers [10, 25, 28] focused on the evolution of the rumour intensity equation in Eq. (6.2) and took advantage of the investigation of rumours' spread within social media. However, we noticed that existing rumour impact models are derived from specific scenarios, which limits their applications.

6.3 Methodology

In this work, we give a formula for defining the impact of rumours that applies to most social media platforms, based on the rumour intensity formula proposed by Chorus (Eq. (6.2)). Three important definitions are given in the following section, namely importance, ambiguity and public critical ability. The framework of our proposed *RISM* model is shown as Fig. 6.1. First, we exact content information from the rumour dataset. Then, based on Eq. (6.7), the impact score is calculated as label for each rumour. Afterwards, we use content-based features TF-IDF and Word2vec [21] to predict the impact score.

6.3.1 Importance

If an event (or a person) is likely to cause a rumour, then the event (or person) is of some importance (a so-called "focus event" or "top person"), the "focus event " or "top person" can be a stage for a rumour producer to catch public eyes. That is, if a rumour attracts a lot of attention (i.e. *thumbs up*, *sharing*), many people are inclined to spend time discussing the topic and even spreading it on social media. Under the circumstances, this rumour is of greater importance. The actions of *thumbs up*, *sharing* and *thumbs down* indicate that the rumour has caught the attention of the public, despite the fact that the public does not like it.

Thus, we use the following objective functions to define the importance of a rumour:

$$I_i = Z_{score}(\sum CN_i) \tag{6.5}$$

where I_i means the importance for rumour d_i, CN_i is the sum of *concern*, *thumbs up*, *thumbs down* and *sharing*, Z_{score} is used to normalise values in order to avoid large value spans.

6.3.2 Ambiguity

As we have stated in Def. 6.1, a rumour is an unverified piece of information that is unverified at the time of publication and may be wholly or partly true, or completely false

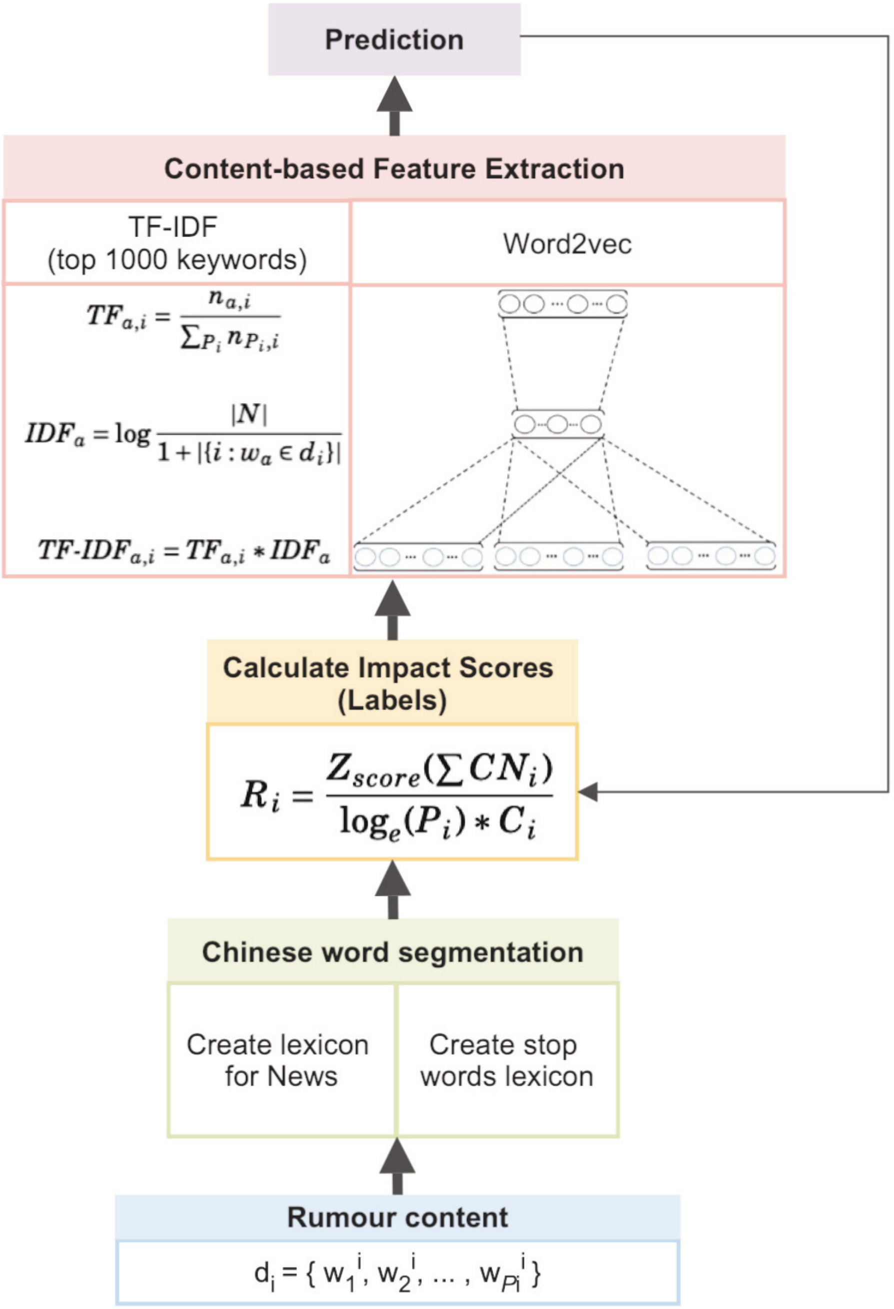

Prediction
Content-based Feature Extraction
TF-IDF
(top 1000 keywords)
Word2vec
$TF_{a,i} = \frac{n_{a,i}}{\sum_{P_i} n_{P_i,i}}$
$IDF_a = \log \frac{|N|}{1+|\{i : w_a \in d_i\}|}$
$TF\text{-}IDF_{a,i} = TF_{a,i} * IDF_a$
Calculate Impact Scores
(Labels)
$R_i = \frac{Z_{score}(\sum CN_i)}{\log_e(P_i) * C_i}$
Chinese word segmentation
Create lexicon for News
Create stop words lexicon
Rumour content
$d_i = \{ w_1^i, w_2^i, \dots, w_{P_i}^i \}$

Figure 6.1: Framework of RISM.

or still unresolved. Several rumours may be subject to this uncertainty, and to the extent that the more they are refuted, the more uncertain the truth becomes. Ambiguity is a major factor influencing this uncertainty. The originator of ambiguity is the deliberate concealment or even distortion of the truth by the rumour-monger. Examples include rumour-mongers who describe only one part of the truth and conceal another part of the news, giving rise to public speculation and suspicion about the original news. Different people, on the other hand, may interpret the news differently, which ultimately leads to the spread of rumours. To accommodate most social media, we assume that the lower the number of words and the higher the ambiguity, the greater the possible impact of a rumour, as illustrated in the following equation.

$$A_i = \frac{1}{\log_e(P_i)} \tag{6.6}$$

where A_i is the ambiguity of rumour d_i, while P_i is the number of words in the rumour, For rumour news $d_i(1 \leqslant i \leqslant N)$, we have $P_i > 1$.

6.3.3 Public Critical Ability

The public cannot make an impartial judgement in the absence of open and transparent information. Moreover, when such information is pertinent to the public's personal interests without timely feedback. Some people may then fall prey to rumours of "reasonable gimmicks" due to a lack of calmness. Back then, the public was more likely to believe it and then to have it confirmed. Especially when the rumour is about issues such as official integrity, the public is used as a "secondary distributor" and a "loudspeaker" once the story is maliciously fabricated. Thus, the critical capacity of the public does have an influence on the impact of rumours. In particular, the more dispassionate and critical the public is, the less impact the rumour will have, and vice versa.

In addition, on social media in general, reviews are a direct way of expressing public opinion on rumours. For example, reviews such as "It must be false!" or "Only a fool would believe it" indicate that the users who have read the rumour and left such comments are highly critical. Alternatively, reviews such as "Is it true?" or "I don't want it to be true" reflect the weak critical skills of those who read the rumour and left such comments.

In this work, we therefore measure the critical capacity of the public based on their attitude to the comments. We used *HowNet*[3] to score the comments for each rumour. A higher score implies that the reader has a high critical ability, whereas the opposite is true for a lower score.

Analogous to *WhatNet* in the English-speaking world, *HowNet* is a large linguistic knowledge base of Chinese (including English) vocabulary and concepts. *HowNet* holds to the reductionist idea that word meanings can be described by a smaller semantic unit.

[3] http://www.keenage.com

This semantic unit is called a "Sememe". As its name suggests, it is the basic, smallest semantic unit, which should not be subdivided. In the course of continuous annotation, *HowNet* has gradually built up a fine-grained system of sememes (about 2000 sememes). *HowNet* has accumulated semantic information on hundreds of thousands of word senses based on the semantic system. Within this work, we employ *HowNet* to analyse the sentiment words in the comments and finally give the rumour $d_i(1 \leqslant i \leqslant N)$ a comment sentiment score $C_i(1 \leqslant i \leqslant N)$).

Although there is a number of rumour intensity equations proposed, which are mainly based on the evolution of Eq. (6.2), we noticed that existing rumour impact models are derived from specific scenarios, which limits their applications. Therefore, we shall choose Eq. (6.2) as our fundamental rumour intensity equation. For rumour $d_i(1 \leqslant i \leqslant N)$ in D, we can compute the value of *importance* (I_i), *ambiguity* (A_i) and *public critical ability* (C_i). Based on rumour intensity of Eq. (6.2) proposed by Chorus (Eq. (6.2)),impact score R_i of rumour d_i is

$$R_i = \frac{Z_{score}(\sum CN_i)}{\log_e(P_i) * C_i} \tag{6.7}$$

6.3.4 Content-based Feature Extraction

Due to the fact that our goal is to predict the impact of rumours at an early stage, it is common for only the content of the rumour, with no other relevant attributes, to be present at the time of the rumour. Hence, in this work, the features are extracted based on the content of the rumour only. In the research of text mining, *TF-IDF* is the most commonly used method, which reflects how important a word is to a document in a collection or corpus. However, *TF-IDF* computes each term within the text separately, which means the possible connections between terms are ignored. So we then choose *Word2vec* to make up for this lack. To sum up, there are two parts to our extracted features, *TF-IDF*, which is widely used in text mining, and *Word2vec*, which represents the semantic information hidden in the text.

6.3.4.1 TF-IDF

The full name of *TF-IDF* is term frequency-inverse document frequency, a numerical statistical method to reflect the importance of a word in a document [68]. The term frequency is often used as a weighting factor in searches for information retrieval, text mining and user modelling. *TF-IDF* is proportionally increased with the number of times the word occurs in the document and is offset by how many documents are in the corpus containing the word, helping to adjust with the fact that certain words occur more frequently in documents. *TF-IDF* is one of the most popular term weighting schemes today, and it is reported that around 83% of the text-based recommendation systems in digital libraries[4] using *TF-IDF* as a feature.

Similar to most of previous work, we take advantages of *TF-IDF* as of our features:

$$TF_{a,i} = \frac{n_{a,i}}{\sum_{P_i} n_{P_i,i}} \tag{6.8a}$$

$$IDF_a = \log \frac{|N|}{1 + |\{i : w_a \in d_i\}|} \tag{6.8b}$$

$$TF\text{-}IDF_{a,i} = TF_{a,i} * IDF_a \tag{6.8c}$$

where $n_{a,i}$ is the number of occurrences for word w_a in rumour d_i, and the denominator of Eq. (6.8a) is the number of words in d_i, in Eq. (6.8b), $|N|$ denotes the total number of rumours in D, $|\{i : w_a \in d_i\}|$ is the number of rumours where the word w_a appears. To prevent the denominator from being zero, we modify the denominator to $1 + |\{i : w_a \in d_i\}|$. And then, we take the product of *TF* and *IDF* as the value of *TF-IDF*.

In order to extract more valuable words as features, we then calculate the information gain of the *TF-IDF* value for each term in each rumour, and filter out the top 1000 keywords, utilising their *TF-IDF* value as *TF-IDF* features.

6.3.4.2 Word2vec

Each term is independent when computing *TF-IDF*, with the possible connections between terms being unknown. Hence, we use *Word2vec* to represent the potential connections across terms. To be more specific, we use Continuous Bag-Of-Words (CBOW) as our training manner.

For the experiments, the dataset used is from *Toutiao.com* and will be presented in section 6.4.1. The Chinese corpus [53], which includes *Wikipedia 2019*, *News corpus* and *Baidu Baike*, was utilised in order to learn vector representations from words. Specifically, *Wikipedia 2019* carries 1 million well-constructed Chinese words. *News Corpus* includes 2.5 million news items. *Baidu Baike* comprises 1.5 million answers and questions. We then used the trained model to generate word vectors for the top 1000 keywords of each rumour. The statistics of the Chinese natural language processing corpus are listed in Table 6.1.

[4] https://en.wikipedia.org/wiki/Tf-idf

Table 6.1: Statistics of Corpus

Chinese Corpus	Description
Wikipedia 2019	$1,043,224$ well-structured Chinese words from Wikipedia
News Corpus	2.43 million news, collected from 2014 to 2016, covering $6,300$ media
Baidu Baike	1.425 million pre-filtered, high quality questions and answers from Baidu Baike

Table 6.2: Statistics of Impact Scores

Field	Mean	Variance	Median
Impact Scores	9.25	8.09	6.95

6.4 Experiment and Analysis

Throughout this section, we conduct experiments to evaluate the validity of the proposed *RISM* model. To begin with, our impact score is calculated as a label for each rumour using the formula Eq. (6.7) defined in Section 6.3, the statistics of the impact scores is shown in Table 6.2. Then we predict the impact scores using the content-based features *TF-IDF* and *Word2vec* described in part 6.3.4. All of the experiments were conducted on a computer with 28 CPU cores and 256 GB of memory.

6.4.1 Dataset

A real-world dataset was collected from *Toutiao.com*, a Chinese news content platform. This dataset consists of news content and social contextual information. The news content includes the meta-attributes of the rumour (e.g. body text) and the social context includes the social engagement of users associated with the rumour item (e.g. number of user comments, shares, likes, etc.). For each news piece, experts in the relevant field labeled it as a rumour or non-rumour. The labeled news items with rumours were collected as our rumour dataset D. Statistics for dataset D are given in Table 6.3.

Table 6.3: Statistics of Dataset

Dataset	Toutiao.com
#rumours	10,548
AVG #words in a rumour	485
AVG #comments	34
AVG #concern	12
AVG #sharing	136
AVG #thumbs up	278
AVG #thumbs down	139

6.4.2 Experiment Metrics

The aim of *RISM* is to predict the rumour impact of the rumour news. To evaluate the effectiveness of the proposed *RISM* model, we adopt a standard metric coefficient of determination, i.e., R-squared.

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{6.9a}$$

$$\textit{R-squared} = 1 - \frac{\sum_i (y_i - \widehat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{6.9b}$$

where the dataset D has n values y_1,...,y_n, y_i for *TF-IDF* features or as a vector $y_i = [y_1, ..., y_n]^\top$ for *Word2vec* features. $\bar{y}$ is the mean of the dataset D. Each value y_i associates with a predicted value $\widehat{y_i}$.

6.4.3 Comparison

First of all, we calculate the impact scores for each rumour news d_i in D based on the Eq. (6.7) illustrated in Section 6.3, and use the calculated impact scores as the labels.

TF-IDF values *TF-IDF*$_{a,i}$ are then calculated for each word w_a^i in d_i ($1 \leqslant a \leqslant P_i$, $1 \leqslant i \leqslant N$) based on the equations illustrated in Section 6.3.4.1. Meanwhile, in order to extract the most valuable words as features, we calculate the information gain IG_a^i of the *TF-IDF*$_{a,i}$. And then we filter out the top 1000 keywords with higher IG and utilize their *TF-IDF*$_{a,i}$ values as *TF-IDF* features. *Word2vec* features are then computed using *Gensim*[5] on the top 1000 keywords. We tried to set the size of each vector from {5, 10,

[5] https://pypi.org/project/gensim/

20} while doing the experiments. When the vector size is set to 20, the computation time is so long that we have to give it up. When the vector size is set to 5, the accuracy is not as expected. Thus, we set the size of each vector as 10, which is a trade-off between accuracy and computational time.

After defining the features and impact score for each rumour news d_i in D, we split train and test datasets with 10 different random seeds for evaluation on Linear Regression (LR) [71], Bayesian Ridge [61], Support Vector Regression (SVR) [9] and Gradient Boosting Regressor (GBR) [31] models. First, we compare *TF-IDF* features with *Word2vec* features separately. Then, we combine *TF-IDF* features and *Word2vec* features together to evaluate the effectiveness of the *RISM* model.

Table 6.4 shows the performance of comparison methods on different kinds of features. Overall, the combination of *TF-IDF* features and *Word2vec* features, i.e., *RISM*, outperforms the *TF-IDF* feature or *Word2vec* feature. *RISM* enables the methods to achieve R-squared value with average 0.7. Besides, it is interesting to find that if we look at *Word2vec* and *TF-IDF* separately, *Word2vec* focuses on the relationship between words and words, while *TF-IDF* focuses on the proportion of a single word in the text. The impact of these two methods on the results is different on different classifiers, indicating that the two are indispensable.

Table 6.4: Comparison of Four Classifiers with Different Kinds of Features

	Linear Regression	Bayesian Ridge	SVR	GBR
Features	R-squared	R-squared	R-squared	R-squared
TF-IDF	0.722	0.631	0.687	0.719
Word2vec	0.752	0.601	0.602	0.734
RISM	**0.811**	**0.690**	**0.689**	**0.804**

6.5 Summary

There are more and more recently proposed ways to detect social media rumours [40, 72]. Meanwhile, the number of rumours on social media is increasing wildly. The mere detection of rumours does not essentially solve the impact of rumours on the public, and some official suppression of rumours is needed. However, there are so many rumours on social media that it is not necessary to refute every single one of them. In this work, therefore, we ask two research questions about rumours on social media. The first is "How can we define the impact of rumours on social networks?" . The second is "How can we predict the likely impact of rumours on social media at an early stage?". To address these challenges, we propose the *RISM* model. The model consists of two components: a calculation of the rumour impact score and a prediction of the impact. Experiments conducted on a real dataset collected from *Toutiao.com* demonstrate the effectiveness of our proposed model.

CHAPTER 7

Sentiment Analysis towards COVID-19 on Twitter

7.1 Introduction

Since the first report of a series of acute respiratory infections in Wuhan, Hubei Province, China in December 2019, the 2019 Coronavirus Disease (COVID-19) pandemic and our understanding of the virus have grown exponentially [1]. As of 8 January, 2020, the cause of these cases was determined to be the new β-coronavirus and then named 2019-nCoV, and 41 cases have been reported [2]. Three months later, more than 1.3 million cases were reported worldwide and $75,000$ people lost their lives [3]. The humanistic and social hazards of this pandemic have inspired several major public health "lessons learned", and the topic of effective and responsible scientific communication is the main theme among them [66]. The spread of the pandemic requires a rapid response from public health authorities. Basic epidemiological and scientific evidence has been obtained at an alarming rate to support these decisions. In the past several months, the COVID-19 science has developed rapidly and in large numbers, and timely scientific exchanges have been conducted through traditional biomedical journals. However, the challenges are still great, as the public are still in panic.

Twitter's global user network is estimated to have 330 million users every month [4], including scientists and epidemiologists who widely use the media for scientific communication [50]. We believe that Twitter has played a fundamental but often erratic role in allowing real-time global communication between scientists during the COVID-19 epidemic on an unprecedented scale. Moreover, preventive measures implemented by national, state, and local governments now affect the daily lives of millions of people around the world. “Social distancing” is the most widely used of these measures, aiming to reduce new infections by reducing physical contact between people. Social distancing measures have resulted in the cancellation of sporting events and conferences, closure of schools and universities, and forced many companies to require their employees to work from home. As more and more social interactions take place online and the conversations surrounding COVID-19 continue to expand, more and more people turn to social networks for up-to-date information. Platforms like Twitter have become the core of technology and social infrastructure, allowing us to stay connected even during the pandemic. In addition to daily information, the main issues regarding the spread of the virus, immunity, medication after recovery, economy etc. are also the focus of people’s attention [87].

Unfortunately, information on social networks may not always align with science. Rumours and misinformation is inevitable and can even spread more quickly than real news. For example, conspiracy theories that COVID-19 originated in China have increased xenophobia towards Asian Americans [80]. The impact and the spreading speed of the COVID-19 pandemic around the world determine the importance of understanding public perceptions and analyzing public behaviour. Failure to do so may cause more serious social panic.

In view of this, Emily et al. [17] began actively collecting posts on Twitter (referred to as “tweets”) from 28 January, 2020, leveraging Twitter’s streaming API and Tweepy to follow specific keywords and accounts that were trending at the time. This kind of actively collecting dataset enables the study of online conversation in the context of a planetary-scale pandemic outbreak of unprecedented proportions and implications, and can also help track scientific coronavirus misinformation and unverified rumours, or enable the understanding of fear and panic.

In this chapter, we shall analyze the public’s sentiments and reactions toward COVID-19 based on the dataset that Emily et al. collected from Twitter. The research questions that we aim to address can be summarized as below.

1. How has the public’s focus on COVID-19-related topics changed over time?

2. What is the difference between the public’s sentiment trends toward COVID-19 related topics?

3. Has the trend of public sentiment on COVID-19-related topics changed over time?

Accordingly, we will discuss in detail in the next few sections. Our main contributions are summarized as follows.

- We summarized five main topics of COVID-19 related tweets using topic modeling method, i.e., Family Situation, Economic Situation, Social Situation, Healthcare Environment and Mental Health.
- We analyzed the sentiment of each tweet on five main topics.
- We gave the analysis of the changes in the distribution of users' attention on five topics and the distribution of sentiment over time.

7.2 Preliminaries

7.2.1 Data Source and Scale

Twitter is a social network platform where users can post and interact with texts, referred to as "tweets". Twitter is a valuable data source for social network discussion related to global events, as it has 166 million daily users [83]. During the outbreak of COVID-19 pandemic, to help with the researches such as tracking scientific coronavirus misinformation or unverified rumours, Emily et al. [17] started the actively collecting of tweets from some official Tweeter account or with some specific keywords. The keywords that Emily et al. actively tracking in their Twitter collection are shown in Table 7.1. Table 7.2 gives the account names that Emily et al. are actively tracking since 22 January, 2020.

In compliance with Twitter's Terms & Conditions, Emily et al. [17] only released the tweet IDs of the tweets related to COVID-19 which have been collected since 22 January, 2020[1]. In order to analyze the changes in public sentiment over time, in our research, we select the tweet IDs released by Emily et al. from February 1, 2020 to May 31, 2020 for a total of 4 months' analysis. Fig. 7.1 displays the number of tweets related to COVID-19 from February 1, 2020 to May 31, 2020.

From Fig. 7.1, we can obviously find that there is a rapid increase in the number of tweets posted around March 12, 2020. As we all known, the World Health Organization (WHO) on March 11 declared COVID-19 a pandemic, pointing to the over 118,000 cases

[1] https://github.com/echen102/COVID-19-TweetIDs.

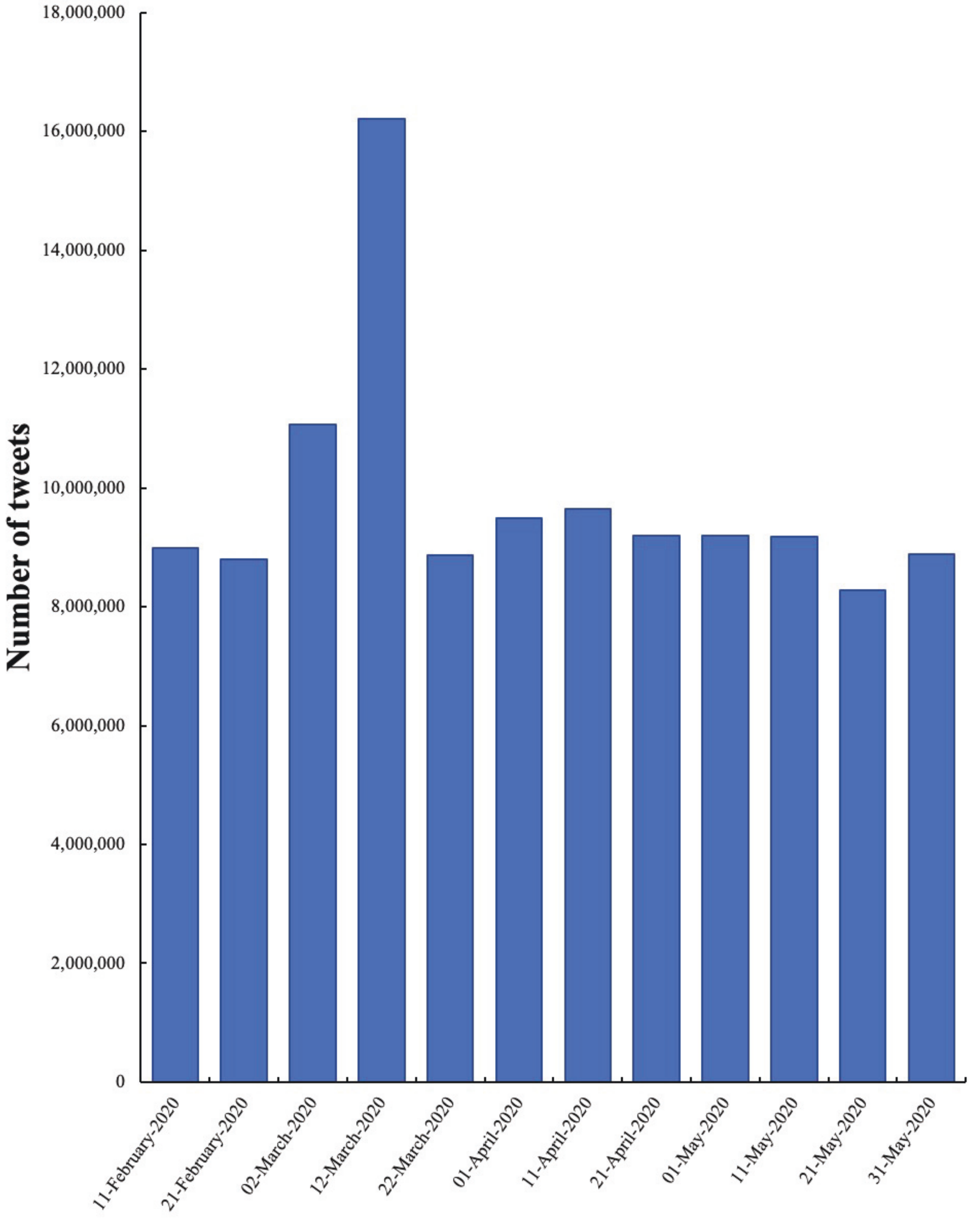


Figure 7.1: Number of COVID-19 related tweets from February 1, 2020 to May 31, 2020.

Table 7.1: Actively tracking keywords in Twitter according to COVID-19

Keywords	Tracked Since
Coronavirus	22/01/2020
Koronavirus	22/01/2020
Corona	22/01/2020
CDC	22/01/2020
Wuhancoronavirus	22/01/2020
Wuhanlockdown	22/01/2020
Ncov	22/01/2020
Wuhan	22/01/2020
N95	22/01/2020
Kungflu	22/01/2020
Epidemic	22/01/2020
Outbreak	22/01/2020
Sinophobia	22/01/2020
China	22/01/2020
Covid-19	16/02/2020
Corona virus	02/03/2020
Covid	06/03/2020
Covid19	06/03/2020
Sars-cov-2	06/03/2020
COVID-19	08/03/2020
COVD	12/03/2020
Pandemic	12/03/2020

Table 7.2: Actively tracking accounts in Twitter according to COVID-19

Account Names	Tracked Since
PneumoniaWuhan	22/01/2020
CoronaVirusInfo	22/01/2020
V2019N	22/01/2020
CDCemergency	22/01/2020
CDCgov	22/01/2020
WHO	22/01/2020
HHSGov	22/01/2020
NIAIDNews	22/01/2020

Table 7.3: Attribute description from *Hydrator* for each tweet

Attribute	Attribute Description
id	tweet ID
created_at	the time when user post the tweet
hashtags	hashtags that user used in the tweet
urls	urls containing in the tweet
lang	language of the tweet (including the most 10 popular languages)
favourite_count	total number of likes
retweet_count	total number of retweets
text	the content of the tweet
tweet_url	url link to the tweet
user_location	the location of the user when registered
user_screen_name	user's nick name when registered
user_time_zone	user's time zone according to the location
user_verified	whether the user is authenticated (True/False)

of the coronavirus illness in over 110 countries and territories around the world and the sustained risk of further global spread. Obviously, the public has a great response to this news. And Tweeter, as a worldwide social network platform, has become an important platform for them to express their opinions and communicate with others. In addition, we can also see the timeliness of the news, because after this sudden increase, the total number of public discussions on Tweeter each month has gradually decreased until the end of May.

For further in-depth analysis, we utilize the *Hydrator*[2] to retrieve the full tweet objects using the tweet IDs released by Emily et al. from February 1, 2020 to May 31, 2020. The attributes we get from the *Hydrator* for each tweet are listed in Table 7.3. In order to better analyze the tweet content, we clean the data according to the following rules.

- *tweet_url* is not null. We can use the URL to find the original tweet which ensures the authenticity of the data.
- *user_verified* is True. Ensure the authenticity of the data.
- *lang* is English. We choose all content in English for analysis.

[2] https://github.com/DocNow/hydrator

Table 7.4: Number of COVID-19 related tweets in each month after screening

Month	#tweets
February	326,833
March	406,903
April	332,908
May	353,902
Total	1,420,546

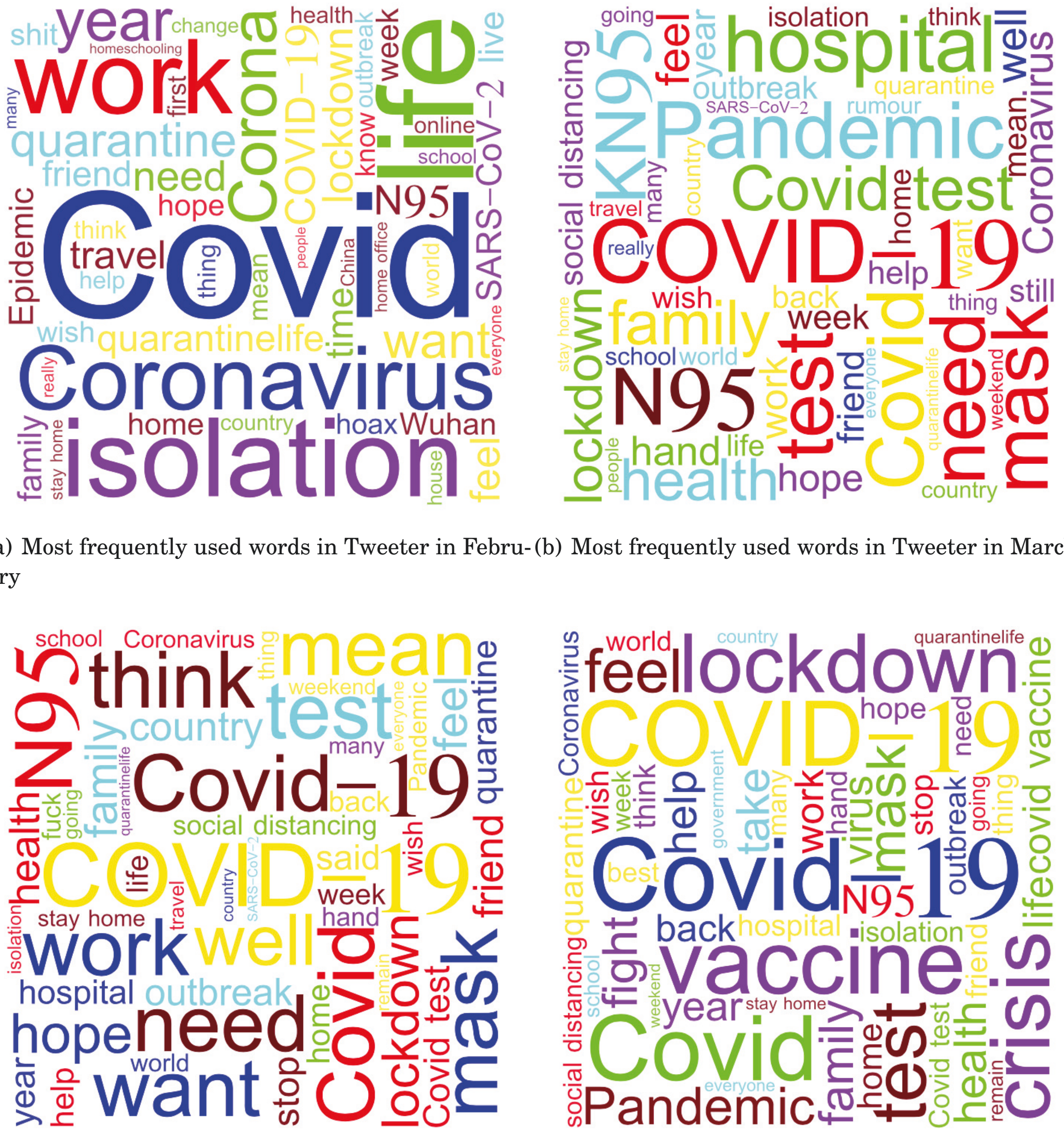
- The number of words in *text* is larger than 10.

Table 7.4 gives the summary of the total number of tweets we collect from each month after cleaning. Combining Figure 7.1 and Table 7.4, we can tell that March 2020 is really a big month as the WHO declared COVID-19 a pandemic. Meanwhile, a lot of countries have issued travel ban and stay-at-home order since March 2020. April 2020 seems more quiet than March as the number of tweets decreased sharply in April 2020. It is possible because there were not too many emergencies in April, so the number of tweets in April will also be significantly lower than the number of tweets in March. While in May 2020, the number of tweets enjoyed a second increase. It is also reasonable, because major breakthroughs have been made in the vaccine research process, and the public's discussion of vaccines has also increased.

7.2.2 Initial Analysis

After accounting for background noise and performing lemmatization, we use word clouds to disclose the most frequently used word stems across Twitter users' post descriptions related to COVID-19. Fig. 7.2 shows four word clouds formed from each month's tweets, i.e., February 1, 2020 to May 31, 2020.

From Fig. 7.2, We can obviously find that though there are many keywords that reappear every month, the focus of Tweeter users changes every month. To be more specific, in February, 2020, words like "isolation", "coronavirus", "covid" take the most important position. It might be because in February, 2020, coronavirus started to attract the attention of the public all around the world, and the public is more concerned about whether quarantine can really alleviate the epidemic. In March, 2020, the most frequently used words in Tweeter change to "pandemic", "mask", "N95", "hospital", etc. We can see that after the WHO announced the COVID-19 pandemic, the public begin to



(a) Most frequently used words in Tweeter in February

(b) Most frequently used words in Tweeter in March

(c) Most frequently used words in Tweeter in April

(d) Most frequently used words in Tweeter in May

Figure 7.2: Word clouds of each month from February 1, 2020 to May 31, 2020.

pay more attention to the issue like, whether the masks are effective, whether the global supply of masks is sufficient, and whether medical resources can keep up. While in April, 2020, words like “work”, “need”, “want”, “think” which reflect the spiritual needs of the public appear more frequently. We can find that after prolonged isolation and working from home, the public began to talk more about psychological needs. When it comes to May, 2020, “vaccine” is the most frequently discussed keyword that suddenly appeared in the public. Because the COVID-19 vaccine finally made a major breakthrough in May 2020 with the unremitting efforts of medical researchers around the world.

In general, we can observe that what the public discusses on Tweeter is always following the real-time news. More interestingly, we can also find that after roughly three months of isolation, the public began to pay attention to mental issues. This gives us an alert that we need to pay attention to the mental health caused by the pandemic after a period of time of the outbreak.

7.3 Topic Modeling and Sentiment Analysis

Topic modeling is used to discovery hidden semantic structures in the tweet text. In this work, we use Latent Dirichlet Allocation (LDA) to help with extracting topics of COVID-19 related tweets.

The LDA is an unsupervised machine learning method which is suitable for performing topic modeling [38] . It groups frequently used words into multiple topics, and performs well for both short and long texts. In this work, we use multiple sets of topic modeling, and each set contains 5 to 10 topics. Then we choose topic sets that looks more reasonable and interpretable. After selecting a set of topics, we reviewed the first 10 words of each topic and unanimously developed a topic name for each topic, i.e. Family Situation, Economic Situation, Social Situation, Healthcare Environment and Mental Health.

As for sentiment analysis, we employed Valence Aware Dictionary and Sentiment Reasoner (VADER) [38] to evaluate whether a COVID-19 related tweet reflect a positive, negative or neutral sentiment, expressing as sentiment score. Sentiment score is calculated for each topic, ranging from -1 to 1, where -1 representing the most negative sentiment and 1 representing the most positive sentiment. In VADER, positive sentiment is categorized by having sentiment scores ≥ 0.05, sentiment scores ranging from -0.05 to 0.05 is categorized as neutral sentiment, and negative sentiment is recognized by having sentiment scores ≤ -0.05. During the experimentation, we manually coded a random sample of 200 tweets as having positive, neutral, or negative sentiments, and checked them based on the sentiment classification output of the machine.

There is a total of 1,420,546 tweets remaining after cleaning over 4 months (Table 7.4). Through sentiment analysis, there are 639,481 tweets (45.0%) classified as having positive COVID-19 sentiment, 354,084 tweets (30.1%) classified as having negative COVID-19 sentiment, and 426,981 tweets (24.9%) classified as having neutral COVID-19 sentiment. Overall, during the first 4 months of COVID-19 global outbreak, positive tweets outweighed negative tweets with a ratio of 1.81 to 1.

To be more specific, Fig. 7.3 shows the frequency distribution of tweets on five main topics across sentiment types from February 1, 2020 to May 31, 2020. In general, there are more positive tweets on the five topics than negative ones, indicating that the public's attitudes were relatively positive in the first four months of the COVID-19 outbreak. Furthermore, the public has the largest proportion of positive views on topics related to the healthcare environment (48.9%), and the least positive attitudes on topics related to Economic Situation (42.1%). It can be seen that in the early stages of the COVID-19 outbreak, the public still had confidence in the healthcare environment. On the contrary,

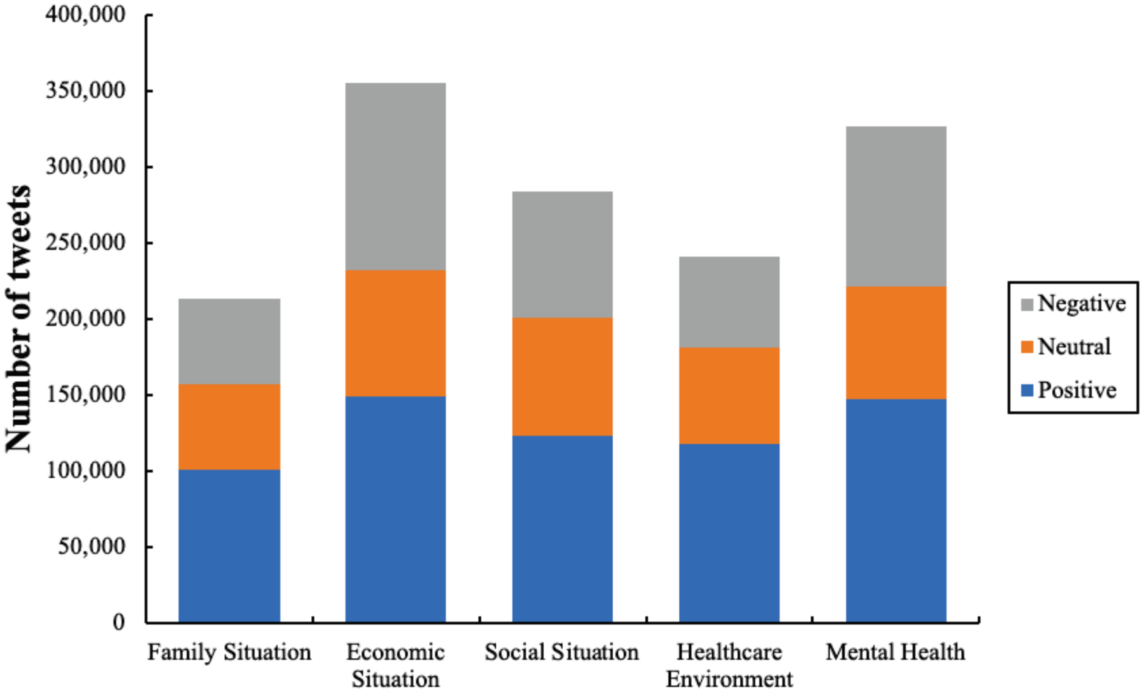
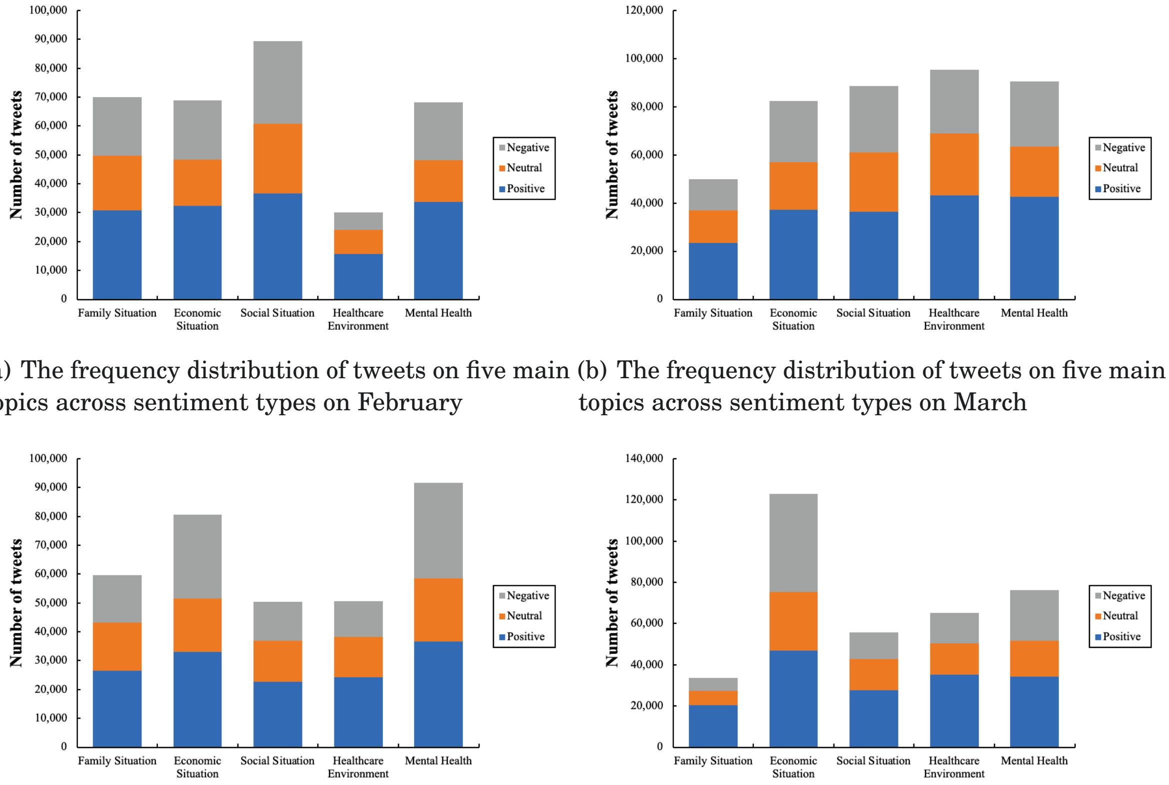


Figure 7.3: The frequency distribution of tweets on five main topics across sentiment types from February 1, 2020 to May 31, 2020

some measures caused by the COVID-19 outbreak, i.e. lockdown, work from home, etc., has aggravated public concerns about the economy.

7.4 Discussion

In order to further explore the distribution of topics and the changes in the public's sentiment trends over time, we shall further analyze the tweets of each month during the first four months of the COVID-19 outbreak in this section.



(a) The frequency distribution of tweets on five main topics across sentiment types on February

(b) The frequency distribution of tweets on five main topics across sentiment types on March

(c) The frequency distribution of tweets on five main topics across sentiment types on April

(d) The frequency distribution of tweets on five main topics across sentiment types on May

Figure 7.4: The frequency distribution of tweets on five main topics across sentiment types on each month from February 1, 2020 to May 31, 2020.

Fig. 7.4 gives a direct visualization on the changing of public's sentiment trends over time. We shall give our analysis as below.

Family Situation Tweets surrounding the family situation are more associated with "quarantine life" and "isolation". It is obviously in Fig. 7.4, family situation topic most often occurs in February (Fig. 7.4(a)) and April (Fig. 7.4(c)). It is reasonable as lockdown began in many places in February. The public began to isolate and control the epidemic at home, thus, many topics related to the family situation generated during this time. And the reason why April became the month of family topic growth, we speculate that it has something to do with mental health being widely mentioned this month.

Economic Situation Another high pressure causing by the lockdown is its impact on the global economy. Thousands of companies closed, and millions of people lost their job. People started to work from home and the unemployment numbers increased rapidly. With the spread of the pandemic and the increase in the time of the world's blockade, there are more and more economic topic related tweets in twitter, but the public‚Äôs attitude towards the economy is becoming more and more negative. Nevertheless, not all posts are related to unemployment. Some tweets indicate that they like to work from home. This finding suggests that strategies to reopen the economy, i.e., working from home, may be both welcome and help reduce the spread of COVID-19.

Social Situation At the time when COVID-19 broke out, not only has the public's daily life been disrupted, but also the social network of them been messed up. The pandemic and its associated social unrest seem to leave individuals in a state of uncertainty. It can be seen from Fig. 7.4 that the public's reaction to the social situation is stronger in the first two months with more negative emotions. But it eases in the next two months. This finding indicates that although the outbreak of the pandemic has disrupted people's daily life and social circles, people will find new social lifestyles over time.

Healthcare Environment Discussions surrounding the healthcare environment overlap with politics. Important materials such as personal protective equipment and intensive care resources are related to the demand for government support. It is interesting to find that in February, 2020 (Fig. 7.4(a)), when the COVID-19 just started to break out, the public discuss a little about healthcare environment. Besides, the public has a very positive attitude towards healthcare environment. However, when it comes to March, 2020 (Fig. 7.4(b)), the discussions around healthcare environment has a rapid increase, as the WHO announced the COVID-19 pandemic. The most frequently used words in this month are “hospital” , “mask”, “N95”, “KN95”, etc, which actually imply the rising of public's concern about healthcare environment.

Mental Health While COVID-19 is challenging our physical health, it is also challenging our mental health. 23% of all tweets discuss about mental health during February 1, 2020 to May 31, 2020. It is worth noting that there are more positive sentiment about mental health in February and March (49.4%, 47.0%). However, in April, the number of mental health related tweets suddenly increase along with a rising of negative sentiment (36%). And this negative sentiment improves in May. This finding suggests that when a pandemic occurs for about three months, people's spiritual needs will have an outbreak period. At this time, we need to be more proactive in paying attention to mental health. One possible reason of the improvement in May is the emergence of vaccines. The breakthrough in the research of vaccine has given people a shot.

7.5 Summary

In this chapter, we collect the COVID-19 related dataset from Twitter to investigate public's sentiment toward COVID-19 during different period of time. Firstly, we utilize LDA to help with the topic modeling and give the five main topics of COVID-19 related tweets, i.e., Family Situation, Economic Situation, Social Situation, Healthcare Environment and Mental Health. Then, we employed VADER to evaluate the sentiment of a tweet. At last, we give the analysis on the changing of the public's sentiment on each COVID-19 related topics over time.

Part III

Part III   Conclusion

Chapter 8

Conclusion and Future Work

8.1 Contribtions

To solve the problem of misbehaviours in social networks, some solutions are proposed in this thesis. The conclusions and main contributions of the works in this thesis are listed.

- This thesis presents a complete multi-relational detection framework, which can be used to detect spammers in multi-relational social networks. The detection framework does not rely on traditional content data, and avoids the shortcomings that social network content data are not rich and difficult to obtain. From the perspective of relations, the hidden information between relation-relation, relation-user, user-user is fully explored, so as to detect spammers in social networks more accurately.

- Taking the needs of integrating various relations and the differences of spammers in different roles into account, this thesis takes the *Tagged.com*, a multi-relational weighted network, as an example. It gives a statistical analysis of each relation attribute, and deeply analyzes each relation based on the differences in user behaviour characteristics. The behaviour characteristics indicators of spammers based on non-content data are proposed. Meanwhile, this provides the feature work on multi-relational feature construction with a necessary data support.

- This thesis proposes a new "Send-Receive" Role Separable Graph-Embedding Model (*RS- GEM*) based on probability matrix factorization. *RS- GEM* build a graph in a shared embedding space first, where nodes represent for users and edges represent for relations between users. Second, the number of interactions between the sending and receiving users is extracted as interaction vectors. Third, the sending user feature matrix and receiving user feature matrix are constructed, and the user-user interaction vector is represented by dot product. The difference between these two vectors is used to fit the probability matrix decomposition model, and the constraint conditions are added to prevent the overfitting problem in the optimization process. Finally, the hidden features of each user in multi-relational social networks are obtained through multi relational mosaic.

- This thesis conducts experiments on a large-scale real-world social network dataset from *Tagged.com*. The experimental results show that the hidden features taken by *RS-GEM* can effectively reflect the difference between spammers and normal users. Besides, cross validation results show a significant improvement over the other baselines in the literature.

- This thesis propose a Multi-level Dependency Model (*MDM*) which exploits user's behaviours in terms of long-term and short-term dependency from both individual-level and union-level. The individual-level dependency considers only a single recent behaviour that may trigger subsequent behaviours. In contrast, the union-level dependency considers the collective influence among a union of relations that are involved in the user's short-term behaviour sequence. *MDM* is capable of exploiting user's long-term behaviours hidden in their multi-relational sequential behaviours along with short-term relational behaviours from multiple perspectives, which largely overcomes the limitation of one-sided exploration of sequences for social spammer detection.

- To model the short-term dependency, *MDM* exploits the relational sequences from both individual-level and union-level perspectives. Besides, *MDM* also utilize the residual network, which can learn high-order sequential dependency among multi-relations and help to improve the accuracy of social spammer detection with relational sequences. Extensive experiments on real-world data demonstrate that *MDM* outperforms the state-of-art baselines of spammer detection.

- Although it is crucial to verify the authenticity of rumours, the increasing use of social networks leads to a emergency of a large number of rumours. Hence, there is no need to pay much attention on a rumour with little impact in the social network. To the best of our knowledge, most existing works regarding rumour impact are solely based on prior knowledge or various other assumptions or even human power. Hence, being targeted to numerically describe the rumour impact in social networks and thus help government to control social rumours are now a top priority. And this thesis provide a novel measurement on the impact of rumours.

- At the time when the rumour appears, usually there is only rumour content without other related attributes. Therefore, this thesis proposed the *RISM* model, which only extract features based on the content of the rumours. Meanwhile, the features we extracted are made up of two parts, one is the *TF-IDF* that is widely used in text mining, and another one is *Word to Vector* that represents the semantic information hidden in the text. An extensive experiments conducted on real-world datasets demonstrate the effectiveness of the proposed *RISM* model.

- This thesis collect the COVID-19 related dataset from Twitter to investigate public's sentiment toward COVID-19 during different period of time. LDA is utilized to help with the topic modeling, and the five main topics of COVID-19 related tweets are given as Family Situation, Economic Situation, Social Situation, Healthcare Environment and Mental Health. VADER is employed to evaluate the sentiment of each tweet. At last, this thesis gives an analysis on the changing of the public's sentiment on each COVID-19 related topics over time.

8.2 Possible Future Work

Although the solutions proposed in this thesis addressed some research problems in misbehaviour analysis in social networks, there still some problem needed to be researched in the future, e.g., data imbalance, lack of ground-truth label, etc. The specific directions for future research are as follows.

- Although the behaviour of spammers in social networks has a wide range of influences, spammers only occupy a small part of social network users, which leads to the emergence of extremely uneven samples. In order to solve the imbalance problem, we can try to use a single classifier, or design a cost-sensitive classifier to improve the accuracy of detecting spammers.

- Datasets such as *Tagged.com* which provides users ground-truth labels are extremely rare. Meanwhile, most of the datasets from social networks such as Twitter, Weibo, Facebook, etc. are unlabeled. Therefore, how to make full use of these unlabeled or partially labeled data sets to detect spammers in social networks has great research value. The current research which can referred is the use of semi-supervised machine learning models to detect spammers in such data sets with missing labels. In the future, this issue is worthy of further study.

- Nowadays on social networks, the text information is much less than before and is relatively short and refined. Besides, the data containing the content is also difficult to obtain due to privacy protection reasons. Therefore, we need relation-dependent but content-independent methods like RS-GEM and MDM proposed in this thesis. However, the accuracy of these two methods is still not very satisfactory. In the future, if we can find a suitable real-world dataset that contains both relational and content information, the combined approach of relational and content-dependent is worth trying.
- Different languages will result in different types of rumours in social networks. Different languages also produce different kinds of rhetorical techniques such as exaggeration, metaphor, personification, etc. A further analyse on the language style of rumours will have a great value. In addition, how to make the most effective refutation against such rumours is also a very interesting research direction.

BIBLIOGRAPHY

[1] ProMED-mail. Undiagnosed pneumonia-China.
Available at https://promedmail.org/promed-post/?id=20191230.6864153 Published December 30, 2019.

[2] ProMED-mail. Undiagnosed pneumonia,ÄîChina: novel coronavirus identified.
Available at https://promedmail.org/promed-post/?id=20200108.6877694 Published January 28, 2020.

[3] Center for Systems Science and Engineering, Johns Hopkins University. COVID-19 dashboard.
Available at https://gisanddata.maps.arcgis.com/apps/opsdashboard/index.html#/bda7594740fd40299423467b48e9ecf6. Published April 6, 2020.

[4] Twitter. Home page.
Available at https://twitter.com/home. Accessed January 26, 2020.

[5] A. AGRESTI, *Categorical data analysis*, vol. 482, John Wiley & Sons, 2003.

[6] L. AKOGLU, M. MCGLOHON, AND C. FALOUTSOS, *Oddball: Spotting anomalies in weighted graphs*, in Pacific-Asia conference on knowledge discovery and data mining, Springer, 2010, pp. 410–421.

[7] G. W. ALLPORT AND L. POSTMAN, *An analysis of rumor*, Public Opinion Quarterly, 10 (1946), pp. 501–517.

[8] J. I. ALVAREZ-HAMELIN, L. DALL'ASTA, A. BARRAT, AND A. VESPIGNANI, *Large scale networks fingerprinting and visualization using the k-core decomposition*, in Advances in Neural Information Processing Systems, 2006, pp. 41–50.

[9] M. AWAD AND R. KHANNA, *Support vector regression*, in Efficient learning machines, Springer, 2015, pp. 67–80.

[10] S. Bajracharya, *Measures of violence: Rumor publics and politics in the kathmandu valley*, Journal of Material Culture, 20 (2015), pp. 361–378.

[11] F. Benevenuto, T. Rodrigues, V. Almeida, J. Almeida, C. Zhang, and K. Ross, *Identifying video spammers in online social networks*, in Proceedings of the 4th international workshop on Adversarial information retrieval on the web, 2008, pp. 45–52.

[12] S. Y. Bhat and M. Abulaish, *Community-based features for identifying spammers in online social networks*, in 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013), IEEE, 2013, pp. 100–107.

[13] J. Brophy and D. Lowd, *Collective classification of social network spam*, in Workshops at the Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[14] G. Cai, H. Wu, and R. Lv, *Rumors detection in chinese via crowd responses*, in Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, IEEE Press, 2014, pp. 912–917.

[15] D. Chakrabarti, *Autopart: Parameter-free graph partitioning and outlier detection*, in European Conference on Principles of Data Mining and Knowledge Discovery, Springer, 2004, pp. 112–124.

[16] H. Chalupsky et al., *Discovering and explaining abnormal nodes in semantic graphs*, IEEE Transactions on Knowledge and Data Engineering, 20 (2008), pp. 1039–1052.

[17] E. Chen, K. Lerman, and E. Ferrara, *Covid-19: The first public coronavirus twitter dataset. arxiv 2020*, arXiv preprint arXiv:2003.07372.

[18] T. Chen and C. Guestrin, *Xgboost: A scalable tree boosting system*, in Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.

[19] Z. Cheng, B. Gao, C. Sun, Y. Jiang, and T.-Y. Liu, *Let web spammers expose themselves*, in Proceedings of the fourth ACM international conference on Web search and data mining, 2011, pp. 525–534.

[20] A. Chorus, *The basic law of rumor.*, The Journal of abnormal and social psychology, 48 (1953), p. 313.

[21] K. W. Church, *Word2vec*, Natural Language Engineering, 23 (2017), pp. 155–162.

[22] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics for machine learning*, Cambridge University Press, 2020.

[23] C. Dellarocas, G. Gao, and R. Narayan, *Are consumers more likely to contribute online reviews for hit or niche products?*, Journal of Management Information Systems, 27 (2010), pp. 127–158.

[24] N. DiFonzo and P. Bordia, *Rumor, gossip and urban legends*, Diogenes, 54 (2007), pp. 19–35.

[25] B. Doerr and M. Fouz, *A time-randomness tradeoff for quasi-random rumour spreading*, Electronic Notes in Discrete Mathematics, 34 (2009), pp. 335–339.

[26] P. Donovan, *How idle is idle talk? one hundred years of rumor research*, Diogenes, 54 (2007), pp. 59–82.

[27] S. Fakhraei, J. Foulds, M. Shashanka, and L. Getoor, *Collective spammer detection in evolving multi-relational social networks*, in ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 1769–1778.

[28] R. Fallahpour, S. Chakouvari, and H. Askari, *Analytical solutions for rumor spreading dynamical model in a social network*, Nonlinear Engineering, 4 (2015), pp. 23–29.

[29] A. Fayazi, K. Lee, J. Caverlee, and A. Squicciarini, *Uncovering crowdsourced manipulation of online reviews*, in Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval, 2015, pp. 233–242.

[30] G. Fei, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh, *Exploiting burstiness in reviews for review spammer detection*, in Proceedings of the International AAAI Conference on Web and Social Media, vol. 7, 2013.

[31] J. H. Friedman, *Greedy function approximation: a gradient boosting machine*, Annals of statistics, (2001), pp. 1189–1232.

[32] C. Grier, K. Thomas, V. Paxson, and M. Zhang, *@ spam: the underground on 140 characters or less*, in Proceedings of the 17th ACM conference on Computer and communications security, 2010, pp. 27–37.

[33] S. Hamidian and M. Diab, *Rumor identification and belief investigation on twitter*, in Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, 2016, pp. 3–8.

[34] S. Hamidian and M. T. Diab, *Rumor detection and classification for twitter data*, in Proceedings of the Fifth International Conference on Social Media Technologies, Communication, and Informatics (SOTICS), 2015, pp. 71–77.

[35] A. Heydari, M. ali Tavakoli, N. Salim, and Z. Heydari, *Detection of review spam: A survey*, Expert Systems with Applications, 42 (2015), pp. 3634–3642.

[36] S. Hochreiter and J. Schmidhuber, *Long short-term memory*, Neural computation, 9 (1997), pp. 1735–1780.

[37] B. Hooi, N. Shah, A. Beutel, S. Günnemann, L. Akoglu, M. Kumar, D. Makhija, and C. Faloutsos, *Birdnest: Bayesian inference for ratings-fraud detection*, in Proceedings of the 2016 SIAM International Conference on Data Mining, SIAM, 2016, pp. 495–503.

[38] M. Hung, E. Lauren, E. S. Hon, W. C. Birmingham, J. Xu, S. Su, S. D. Hon, J. Park, P. Dang, and M. S. Lipsky, *Social network analysis of covid-19 sentiments: Application of artificial intelligence*, Journal of medical Internet research, 22 (2020), p. e22590.

[39] T. R. Jensen and B. Toft, *Graph coloring problems*, vol. 39, John Wiley & Sons, 2011.

[40] Z. Jin, J. Cao, Y. Zhang, and J. Luo, *News verification by exploiting conflicting social viewpoints in microblogs*, in Thirtieth Aaai Conference on Artificial Intelligence, 2016.

[41] N. Jindal and B. Liu, *Opinion spam and analysis*, in Proceedings of the 2008 international conference on web search and data mining, 2008, pp. 219–230.

[42] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980, (2014).

[43] R. Krestel and L. Chen, *Using co-occurrence of tags and resources to identify spammers*, in Proceedings of 2008 ECML/PKDD Discovery Challenge Workshop, 2008, pp. 38–46.

[44] K. Lee, J. Caverlee, and S. Webb, *Uncovering social spammers: social honeypots+ machine learning*, in Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, 2010, pp. 435–442.

[45] F. H. Li, M. Huang, Y. Yang, and X. Zhu, *Learning to identify review spam*, in Twenty-second international joint conference on artificial intelligence, 2011.

[46] X. Li, M. Zhang, Y. Liu, S. Ma, Y. Jin, and L. Ru, *Search engine click spam detection based on bipartite graph propagation*, in Proceedings of the 7th ACM international conference on Web search and data mining, 2014, pp. 93–102.

[47] G. Liang, W. He, C. Xu, L. Chen, and J. Zeng, *Rumor identification in microblogging systems based on users' behavior*, IEEE Transactions on Computational Social Systems, 2 (2015), pp. 99–108.

[48] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, *Detecting product review spammers using rating behaviors*, in Proceedings of the 19th ACM international conference on Information and knowledge management, 2010, pp. 939–948.

[49] C. Lin, J. He, Y. Zhou, X. Yang, K. Chen, and L. Song, *Analysis and identification of spamming behaviors in sina weibo microblog*, in proceedings of the 7th workshop on social network mining and analysis, 2013, pp. 1–9.

[50] Y. Lin, *Twitter statistics every marketer should know in 2020*, Dostupné, 17 (10), p. 2020.

[51] Z. C. Lipton, J. Berkowitz, and C. Elkan, *A critical review of recurrent neural networks for sequence learning*, arXiv preprint arXiv:1506.00019, (2015).

[52] B. Liu et al., *Sentiment analysis and subjectivity.*, Handbook of natural language processing, 2 (2010), pp. 627–666.

[53] J. Liu, F. Wu, C. Wu, Y. Huang, and X. Xie, *Neural chinese word segmentation with lexicon and unlabeled data via posterior regularization*, in The World Wide Web Conference, ACM, 2019, pp. 3013–3019.

[54] S. LIU, G. LI, T. TRAN, AND Y. JIANG, *Preference relation-based markov random fields for recommender systems*, in Asian Conference on Machine Learning, 2016, pp. 157–172.

[55] R. MCCREADIE, C. MACDONALD, AND I. OUNIS, *Crowdsourced rumour identification during emergencies*, in Proceedings of the 24th International Conference on World Wide Web, ACM, 2015, pp. 965–970.

[56] A. MUKHERJEE, A. KUMAR, B. LIU, J. WANG, M. HSU, M. CASTELLANOS, AND R. GHOSH, *Spotting opinion spammers using behavioral footprints*, in Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, 2013, pp. 632–640.

[57] A. MUKHERJEE, B. LIU, AND N. GLANCE, *Spotting fake reviewer groups in consumer reviews*, in Proceedings of the 21st international conference on World Wide Web, 2012, pp. 191–200.

[58] A. J. MURMANN, *Enhancing spammer detection in online social networks with trust-based metrics.*, (2009).

[59] L. PAGE, S. BRIN, R. MOTWANI, AND T. WINOGRAD, *The pagerank citation ranking : Bringing order to the web*, Stanford Digital Libraries Working Paper, (1998), pp. 1–14.

[60] M. PARAMESWARAN, H. RUI, AND S. SAYIN, *A game theoretic model and empirical analysis of spammer strategies*, in Collaboration, Electronic Messaging, AntiAbuse and Spam Conf, vol. 7, Citeseer, 2010.

[61] F. PEDREGOSA, G. VAROQUAUX, A. GRAMFORT, V. MICHEL, B. THIRION, O. GRISEL, M. BLONDEL, P. PRETTENHOFER, R. WEISS, V. DUBOURG, J. VANDERPLAS, A. PASSOS, D. COURNAPEAU, M. BRUCHER, M. PERROT, AND E. DUCHESNAY, *Scikit-learn: Machine learning in Python*, Journal of Machine Learning Research, 12 (2011), pp. 2825–2830.

[62] S. V. PEMMARAJU AND S. S. SKIENA, *Computational discrete mathematics : combinatorics and graph theory with mathematica*, Cambridge University Press, 2009.

[63] F. PENG, D. SCHUURMANS, AND S. WANG, *Augmenting naive bayes classifiers with statistical language models*, Information Retrieval, (2004), pp. 317–345.

[64] B. PEROZZI, L. AKOGLU, P. IGLESIAS SÁNCHEZ, AND E. MÜLLER, *Focused clustering and outlier detection in large attributed graphs*, in Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 2014, pp. 1346–1355.

[65] A. PITSILLIDIS, K. LEVCHENKO, C. KREIBICH, C. KANICH, G. M. VOELKER, V. PAXSON, N. WEAVER, AND S. SAVAGE, *Botnet judo: Fighting spam with itself.*, in NDSS, 2010.

[66] S. POLLETT AND C. RIVERS, *Social media and the new world of scientific communication during the covid-19 pandemic*, Clinical Infectious Diseases, 71 (2020), pp. 2184–2186.

[67] F. Y. QIAN WANG, *Allport and postman posed rumor propagation formula improvement and its verification: Rumor analysis of sina micro-blog based on the tourist casualties caused by northeast tiger*, Chinese journal of journalism and communication, 39, pp. 47–67.

[68] A. RAJARAMAN AND J. D. ULLMAN, *Mining of massive datasets*, Cambridge University Press, 2011.

[69] S. RAYANA AND L. AKOGLU, *Collective opinion spam detection: Bridging review networks and metadata*, in Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining, 2015, pp. 985–994.

[70] T. SCHANK, *Algorithmic aspects of triangle-based network analysis*, Phd in Computer Science, University Karlsruhe, (2007).

[71] G. A. SEBER AND A. J. LEE, *Linear regression analysis*, vol. 329, John Wiley & Sons, 2012.

[72] K. SHU, A. SLIVA, S. WANG, J. TANG, AND H. LIU, *Fake news detection on social media: A data mining perspective*, Acm Sigkdd Explorations Newsletter, 19 (2017).

[73] B. SRIRAM, D. FUHRY, E. DEMIR, H. FERHATOSMANOGLU, AND M. DEMIRBAS, *Short text classification in twitter to improve information filtering*, in Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, 2010, pp. 841–842.

[74] G. Stringhini, C. Kruegel, and G. Vigna, *Detecting spammers on social networks*, in Proceedings of the 26th annual computer security applications conference, 2010, pp. 1–9.

[75] J. Tang and K. Wang, *Personalized top-n sequential recommendation via convolutional sequence embedding*, in Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018, pp. 565–573.

[76] C.-Y. Tseng, J.-W. Huang, and M.-S. Chen, *Promail: Using progressive email social network for spam detection*, in Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2007, pp. 833–840.

[77] N. N. Vo, X. He, S. Liu, and G. Xu, *Deep learning for decision making and the optimization of socially responsible investments and portfolio*, Decision Support Systems, 124 (2019), p. 113097.

[78] C. Wang, *Construction and resolution of rumor spreading model of public emergencies*, Journal of Modern communication, (2010), pp. 45–48.

[79] G. Wang, X. Zhang, S. Tang, C. Wilson, H. Zheng, and B. Y. Zhao, *Clickstream user behavior models*, ACM Transactions on the Web (TWEB), 11 (2017), pp. 1–37.

[80] P. Wang, N. Anderson, Y. Pan, L. Poon, C. Charlton, N. Zelyas, D. Persing, D. Rhoads, and H. Babcock, *The sars-cov-2 outbreak: diagnosis, infection prevention, and public perception*, Clinical chemistry, 66 (2020), pp. 644–651.

[81] X. Wang, Q. Li, W. Zhang, G. Xu, S. Liu, and W. Zhu, *Joint relational dependency learning for sequential recommendation*, in Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2020, pp. 168–180.

[82] S. Weibo, *@ weibopiyao*, 2015.

[83] Q. Wong, *Twitter‚Äôs user growth soars amid coronavirus, but uncertainty remains. cnet*, 2020.

[84] Z. Wu, Y. Wang, Y. Wang, J. Wu, J. Cao, and L. Zhang, *Spammers detection from product reviews: a hybrid model*, in 2015 IEEE International Conference on Data Mining, IEEE, 2015, pp. 1039–1044.

[85] C. J. Wu Zhiang, Wang Youquan, *A survey on shilling attack models and detection techniques for recommender systems (in chinese)*, 59 (2013), pp. 551–560.

[86] Z. Xing, J. Pei, and E. Keogh, *A brief survey on sequence classification*, ACM Sigkdd Explorations Newsletter, 12 (2010), pp. 40–48.

[87] G. Ye, Z. Pan, Y. Pan, Q. Deng, L. Chen, J. Li, Y. Li, and X. Wang, *Clinical characteristics of severe acute respiratory syndrome coronavirus 2 reactivation*, Journal of Infection, 80 (2020), pp. e14–e17.

[88] D. Yin, S. D. Bond, and H. Zhang, *Anxious or angry? effects of discrete emotions on the perceived helpfulness of online reviews*, MIS quarterly, 38 (2014), pp. 539–560.

[89] J. Yin, Q. Li, S. Liu, Z. Wu, and G. Xu, *Leveraging multi-level dependency of relational sequences for social spammer detection*, Neurocomputing, 428 (2021), pp. 130–141.

[90] J. Yin, S. Liu, Q. Li, and G. Xu, *Prediction and analysis of rumour's impact on social media*, in 2019 6th International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC), IEEE, 2019, pp. 1–6.

[91] J. Yin, Z. Zhou, S. Liu, Z. Wu, and G. Xu, *Social spammer detection: a multi-relational embedding approach*, in Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2018, pp. 615–627.

[92] L. Yu, C. Zhang, S. Liang, and X. Zhang, *Multi-order attentive ranking model for sequential recommendation*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 5709–5716.

[93] Y.-Y. Zhao, B. Qin, T. Liu, et al., *Sentiment analysis*, Journal of Software, 21 (2010), pp. 1834–1848.

[94] Z. Zhao, P. Resnick, and Q. Mei, *Enquiring minds: Early detection of rumors in social media from enquiry posts*, in Proceedings of the 24th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2015, pp. 1395–1405.

[95] B. Zhou and J. Pei, *Link spam target detection using page farms*, ACM Transactions on Knowledge Discovery from Data (TKDD), 3 (2009), pp. 1–38.

[96] Y. Zhou, C. Huang, Q. Hu, J. Zhu, and Y. Tang, *Personalized learning full-path recommendation model based on lstm neural networks*, Information Sciences, 444 (2018), pp. 135–152.

[97] F. Zhu and X. Zhang, *Impact of online consumer reviews on sales: The moderating role of product and consumer characteristics*, Journal of marketing, 74 (2010), pp. 133–148.

[98] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, and R. Procter, *Detection and resolution of rumours in social media: A survey*, ACM Computing Surveys (CSUR), 51 (2018), pp. 1–36.

[99] A. Zubiaga, M. Liakata, and R. Procter, *Learning reporting dynamics during breaking news for rumour detection in social media*, arXiv preprint arXiv:1610.07363, (2016).

[100] ——, *Exploiting context for rumour detection in social media*, in International Conference on Social Informatics, Springer, 2017, pp. 109–123.