Towards better prediction and content detection through online social media mining

Chen, Weiling

2018

Chen, W. (2018). Towards better prediction and content detection through online social media mining. Doctoral thesis, Nanyang Technological University, Singapore.

<http://hdl.handle.net/10356/75925>

<https://doi.org/10.32657/10356/75925>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

**TOWARDS BETTER PREDICTION AND
CONTENT DETECTION THROUGH ONLINE
SOCIAL MEDIA MINING**

CHEN WEILING

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

2018

**TOWARDS BETTER PREDICTION AND
CONTENT DETECTION THROUGH ONLINE
SOCIAL MEDIA MINING**

CHEN WEILING

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

A thesis submitted to Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2018

*To Dad and Mom,
Cao Pi and Furuya Rei,
for their encouragements and love.*

Acknowledgements

This thesis would not be possible without many people who have helped me and changed my life deeply during my study at Nanyang Technological University (NTU).

First and foremost, I would like to express my sincerest gratitude to my supervisors, Associate Professors Lau Chiew Tong and Yeo Chai Kiat for giving me all the support and freedom to carry out the research which I am interested in. In addition, I would also like to extend my appreciation to Associate Professor Lee Bu Sung who has given me many useful suggestions for my research. Their precious and warm help in my research and studies, infectious enthusiasm and kindness, unlimited patience and inspiring guidance have been the major driving force during my candidature at NTU. Under their supervision, I have developed skills in critical thinking, research methodologies and integrity, technical writing, communication and leadership, which are invaluable for my future career.

Moreover, I would like to thank the students and staff of Computer Networks and Communications Graduate Lab (CNCL) in the School of Computer Science and Engineering (SCSE) at NTU. I would like to thank my seniors, Yang Yiqun, Pham Thi Ngoc Diep; my lab peers, Zhang Yan and Chen Zhaomin; my junior, Yean SeangLidet and my good friends, Liang Yuhan, Chu Zhaowei and Li Qiye. They have not only provided me with their valuable suggestions for my studies and research, but also enriched my life at NTU with unforgettable experiences.

Last but not least, I cannot end without giving my special thanks to my parents for their continuous support and endless love. I am also grateful to Cao Pi and Furuya Rei whom I always admire and give me all the courage to overcome the difficulties I have met. This thesis cannot be finished without them and is dedicated to them.

Contents

Acknowledgements	i
List of Figures	viii
List of Tables	ix
List of Abbreviations	x
Abstract	x
1 Introduction	1
1.1 Background	1
1.1.1 Good Aspects of OSN	2
1.1.1.1 Social Functions	2
1.1.1.2 Marketing and Business	3
1.1.1.3 Data Analytics	4
1.1.2 Bad Aspects of OSN	5
1.1.2.1 Low-quality content	6
1.1.2.2 Misinformation	7
1.1.3 Discussions	9
1.2 Motivation and Scope	9
1.2.1 Content Polluters Detection	9
1.2.2 Rumor Detection	11
1.2.3 Stock Index Prediction	14
1.3 Methodology	16
1.4 Major Contributions	16

1.4.1	A Real-time Low-quality Content Detection Framework	17
1.4.2	An Unsupervised Rumor Detection Model based on Users' Behaviors	18
1.4.3	RNN-Boost - A Hybrid Model for Predicting Stock Market Index	19
1.5	Thesis Organization	19
2	Literature Review	22
2.1	Low-quality Content Detection	22
2.1.1	Definition of Low-quality Content	22
2.1.2	Spam Detection	24
2.1.2.1	Content feature based filters	24
2.1.2.2	Non-content feature based filters	25
2.1.3	Phishing Detection	26
2.1.3.1	Blacklists	26
2.1.3.2	Heuristics	27
2.1.3.3	Data Mining	29
2.1.4	Low-quality Content Detection	29
2.1.4.1	Content-based methods	30
2.1.4.2	Non-content based methods	31
2.1.4.3	Real-time detection	32
2.1.5	Discussions	33
2.2	Rumor Detection	33
2.2.1	Studying Rumors	33
2.2.1.1	Definition of Rumor	33
2.2.1.2	Taxonomy of Rumors	34
2.2.2	Theories Related to Rumors	35
2.2.2.1	User Behaviors	35
2.2.2.2	Propagation	36
2.2.3	Feature Selection	37
2.2.4	Detection Methods	39
2.2.4.1	Classification	39
2.2.4.2	Anomaly Detection	40

2.2.5	Discussions	41
2.3	Stock Index Prediction	42
2.3.1	Market Efficiency	42
2.3.1.1	Definition	42
2.3.1.2	Limitations	43
2.3.1.3	Adaptive Market Hypothesis	44
2.3.2	Feature engineering in financial prediction	45
2.3.2.1	Technical Analysis	45
2.3.2.2	Fundamental Analysis	46
2.3.2.3	Combination of Technical and Fundamental Analysis	48
2.3.3	Machine learning in financial prediction	49
2.3.4	Discussions	51
2.4	Evaluation metrics	51
2.4.1	Content Detection	51
2.4.2	Trend Prediction	52
2.5	Summary	53
3	Real-time Low-quality Content Detection Framework from the Users' Perspective	54
3.1	General Overview	55
3.1.1	Terminology	55
3.1.2	Overview of the Framework	56
3.2	A study on low-quality content from users' perspective	57
3.2.1	Cluster analysis of low-quality content	57
3.2.2	Design of the survey	60
3.2.3	Results of the survey	61
3.3	Identifying features characterizing low-quality content	66
3.3.1	Direct features	66
3.3.2	Indirect features	68
3.3.3	Word level analysis	69
3.4	Pre-implementation tweet processing	70
3.4.1	Data collection and preprocessing	70

3.4.2	Labeling tweets	71
3.4.3	Training and testing classifiers	72
3.5	Implementation results and evaluation	73
3.5.1	Word level analysis	73
3.5.2	Feature rank	74
3.5.3	Detection performance	76
3.5.4	Comparisons with other methods	77
3.5.4.1	Blacklists and Twitter policy	77
3.5.4.2	Other spam/phishing detection methods	78
3.6	Conclusions	79
4	Unsupervised Rumor Detection Model based on User Behaviors	81
4.1	General Overview	82
4.1.1	Definition of Rumors	82
4.1.2	Descriptions of the Problem	83
4.1.3	Overview of the Model	84
4.2	Data Collection Methods and Detection Models	85
4.2.1	Data Collection	85
4.2.2	Feature Selection	87
4.2.3	Recurrent Neural Networks	89
4.2.4	Autoencoder	92
4.2.5	Combination Model of RNN and AE	93
4.2.6	Rumor Detection	95
4.3	Results and Comparisons	97
4.3.1	One Hidden Layer vs. Multiple Hidden Layers	98
4.3.2	Standard Autoencoder vs. Proposed Variant Autoencoder	99
4.3.3	One Aggregated Model vs. Individual Models	99
4.3.4	Proposed Model vs. Other Methods	100
4.4	Conclusions	101
5	Leveraging Social Media News to Predict Stock Index Movement Using RNN-Boost	102

5.1	Overview of the Model	103
5.2	Methods and models	104
5.2.1	Data collection	105
5.2.2	Feature engineering	107
5.2.2.1	Technical features	107
5.2.2.2	Content features	108
5.2.3	Recurrent Neural Networks	110
5.2.4	Adaboost	113
5.2.5	RNN-Boost	114
5.3	Results and Comparisons	116
5.3.1	Single RNN with different feature sets	117
5.3.2	Single RNN vs. RNN-Boost	118
5.3.3	Comparisons with other methods	119
5.4	Conclusions	119
6	Conclusions and Future Work	121
6.1	Conclusions	121
6.1.1	Real-time Content Polluter Detection Framework from the Users' Perspective	121
6.1.2	Unsupervised Rumor Detection Model based on User Behaviors	123
6.1.3	Hybrid Model for Predicting Stock Market Index	124
6.2	Future Work	125
6.2.1	Content Polluter Detection	125
6.2.2	Rumor Detection	126
6.2.3	Stock Index Prediction	126
6.2.4	Other Research Directions	128
Appendix A Survey About Users' Opinions on Low-quality Content		130
Appendix B Examples of blacklist keywords		135
Appendix C Examples of stock sensitive sentiment dictionary		137

Bibliography	139
Author's Publications	153

List of Figures

3.1	Overview of the low-quality content Detection Framework	57
3.2	Users' habits about following and cleaning up friends	63
3.3	Users' definition for low-quality content (Abstract categories)	63
3.4	Users' definition for low-quality content (Specific examples)	64
3.5	F1 measure with/without stemming for different dictionary size	74
3.6	Accuracy of different subsets of features	75
4.1	Overview of the proposed rumor detection model.	85
4.2	Proposed RNN module.	91
4.3	Proposed autoencoder module.	92
4.4	The combination of RNN and AE.	94
4.5	Standard deviation of rumors' and non-rumors' error on recent Weibo set	96
4.6	Performance of learning models with different number of hidden layers.	98
5.1	Overview of the proposed model.	104
5.2	One-hidden-layer RNN module	110
5.3	Structure of GRU unit	111
5.4	An overview of Adaboost.R2	114

List of Tables

2.1	Confusion matrix	52
3.1	How much do content polluters affect your user experience when using social network sites?	61
3.2	If someone follows you, will you follow back?	62
3.3	How often will you clean up your followees/friends?	62
3.4	What's the maximum threshold (as a percentage of your recently received messages) you can bear before considering unfollowing him/her?	65
3.5	Direct features	67
3.6	Indirect features	68
3.7	Feature rank	75
3.8	Detection performance of different feature subsets	76
3.9	Comparisons of different methods	79
4.1	Weibo based features	88
4.2	Comment based features	89
4.3	Comparisons of Detection Performance of Different Methods	97
5.1	Basic Features and the Formulas	107
5.2	Prediction results for different feature subsets	117
5.3	Comparisons between RNN and RNN-Boost	118
5.4	Comparisons with other methods	119

List of Abbreviations

ABBREVIATIONS	FULL EXPRESSIONS
AE	AutoEncoder
AMH	Adaptive Market Hypothesis
AUC	Area Under the ROC Curve
CNN	Convolutional Neural Networks
CSI	Chinese Stock Index
EM	Expectation Maximization
EMH	Efficient Market Hypothesis
FinTech	Financial Technology
FPR	False Positive Rate
GA	Genetic Algorithm
GRU	Gated Recurrent Units
HS300	Shanghai and Shenzhen 300 stock index
HMM	Hidden Markov Models
IG	Information Gain
KNN	K Nearest Neighbors
LDA	Latent Dirichlet Allocation
MACD	Moving Average Convergence/Divergence
NBC	Naive Bayes Classifier
NLP	Natural Language Processing
OSN	Online Social Networks
RF	Random Forest
RFE	Recursive Feature Elimination
RNN	Recurrent Neural Networks
RWT	Random Walk Theory
SNS	Social Networking Sites
SVM	Support Vector Machines
TF-IDF	Term Frequency-Inverse Document Frequency

Abstract

With the astronomical growth of Online Social Networks (OSN), they have become the new target of many cyber criminals like spammers and phishers and many advertisers which have resulted in worrying issues. These issues range from low-quality content to phishing and frauds. Rumor diffusion is another problem causing serious social issues. Since information can propagate much faster than ever on OSN, the negative impact of rumors is thus much worse. However, we would not stop using OSN to interact with our friends and acquaintances, to share news and information, and to take part in other interesting online activities just because of the issues it may cause. As a matter of fact, with the content collected from OSN, data analysts would be able to predict box office, terrorism and even the stock price and a lot of other interesting topics.

OSN is like a double-edged sword. Therefore it is necessary to reduce the negative effect of it and benefit as many individuals and organizations as possible. In this thesis, the author carries out research on making detection and prediction tasks more accurate through mining the different aspects of the content collected from OSN.

Detection techniques of malicious content like spam and phishing on OSN are common while in contrast little attention is paid to other low-quality content which actually impacts user browsing experience most. The author proposes a framework to detect low-quality content from the users' perspective in real time. Based on preliminary studies, a survey is carefully designed to gather users' opinions on different categories of low-quality content. Both direct and indirect features including newly proposed features are identified to characterize the different types of low-quality content. The author then combines word level analysis with the identified features and builds a keyword blacklist dictionary to improve the detection performance. The author labels an

extensive Twitter dataset of 100,000 tweets and performs low-quality content detection in real time based on the characterized significant features and word level analysis.

Since information can spread rapidly and widely more than ever on OSN, they have become new hot beds of misinformation diffusion. Owing to the potential harm the false information may bring to the public, rumor detection has become a significant but challenging research topic. In order to detect the few but potentially harmful rumors to prevent the public issues they may cause, the author proposes an unsupervised learning model combining Recurrent Neural Networks (RNN) and Autoencoders (AE) to distinguish rumors as anomalies from other credible microblogs based on users' behaviors. In addition, some features based on comments posted by other users are newly proposed and are then analyzed over their posting time so as to exploit the crowd wisdom to improve the detection performance.

The reason that people sometimes read rumors from OSN is because today, OSN play a significant role as a platform for information sharing especially news updates. News from traditional media has been used to facilitate the prediction of stock movement for a long time. This inspires the author to exploit the news content collected from OSN to predict stock index movement. In this work, the author carefully selects official accounts from China's largest OSN, i.e. Sina Weibo and analyzes the news content crawled from these accounts by extracting sentiment features and Latent Dirichlet allocation (LDA) features. The author then inputs these features together with technical indicators into a novel model called RNN-boost to predict the stock volatility in the Chinese stock market.

The work presented in this thesis demonstrates the boon and bane of OSN and provides the methodologies and applications to exploit the good aspects while minimizing OSN's potential negative impact. The author would expect that this thesis could give some insights into the future work of related research.

Chapter 1

Introduction

In this chapter, the author first presents the background introduction of Online Social Networks (OSN, also known as Social Networking Site or SNS). Based on the merits and limitations of OSN, the motivation and scope of the research are specified. Finally, key contributions of the thesis are listed.

1.1 Background

OSN are platforms which help people to build social relations with those who share similar interests and background as well as to enable real-life connections. Each user has a profile which displays his or her personal information for other users to search, browse and then build connections. Users on the same OSN can interact with one another via updates, messages etc [1]. Furthermore, OSN in a Web 2.0 era have developed from monotonous social interactions and communication platform into an integration of social media functions for all kinds of services [2].

In the last decade, more and more social network sites have sprung up and at-

tracted millions of users. Among them, Facebook, QQ, Sina Weibo, Twitter are the most popular, with 2,061 million, 850 million, 361 million and 328 million active users respectively as of September 2017 [3].

These web-based social networking services enable people to connect with others regardless of borders of politics, geography and background [4]. Through the features and functions provided by these OSN such as email and instant messaging, online communities are founded and a large amount of new content is being created every day. Such content has been collected and analyzed for research or commercial purposes and has been proven to be of real value. The use of OSN has changed the contemporary world in all aspects of life. On the other hand, with the fast growth of OSN, they have become the new target of many cyber criminals and has caused many worrying issues.

1.1.1 Good Aspects of OSN

Nowadays more and more people use social media to connect with others, engage with news content, share information and entertain themselves.

1.1.1.1 Social Functions

As is indicated in its name, the most important feature of OSN is to enable social interactions. With an OSN account and the access to the Internet, one can easily have a worldwide connectivity. Even though there are already many popular online social platforms like Facebook, LinkedIn, Pinterst, etc., new websites with their own features are popping up daily to help people build connections over the Web.

On these social networking sites, users can build business connections, make new friends, or just enlarge their friend circles by connecting and interacting with friends' friends - which may have a huge effect in the future. These contacts facilitate a vari-

ety of functionalities such as match making, job hunting, building communities with common interests or opinions.

It is very convenient for users to share their stories and opinions on OSN and comment on one another's posts among friend circles. From this perspective, OSN fulfill the emotional needs of individuals and enhance the social relationship between friends and acquaintances.

1.1.1.2 Marketing and Business

What is worth mentioning here is that not only individual users create accounts on OSN. Nowadays, more and more organizations also create their official accounts as can be seen on the social networking platforms. These organizations range from news agency to institutes, from companies to governments.

These organizations interact with their followers via the official accounts, answer their questions and collect their feedbacks which are helpful for building a positive image, increasing brand awareness, achieving better customer satisfaction and finally improving brand loyalty and leadership.

OSN are especially useful for marketing campaigns. Regardless of whether the companies have online or offline business, they can promote their products and services to a large number of audience. The best point of marketing on online social media is its low cost compared to advertising on traditional media and this in turn makes the business more profitable. In addition, some OSN also provide fee-based options for advertising and marketing which employ machine learning and data mining techniques to deliver specific content to the target groups. This method maximizes the targeted audience while minimizing the potential waste of resources.

Another aspect which justifies the unique feature of OSN in marketing and business

is its amazing news cycle speed. It is beyond doubt that the information can propagate faster than ever via OSN. This has led to a revolution in traditional journalism. Nowadays, most news agencies rely on OSN to collect and share the latest news and information. Websites like Twitter and Weibo are gradually becoming the main sources for breaking news. When some emergencies happen, it is very useful for authoritative organizations to verify news or debunk rumors on OSN. OSN have provided the most efficient way to publish such announcements as information has the largest outreach within the shortest time period. This helps the organizations and individuals to reduce the potential negative social impact that the misinformation might cause.

1.1.1.3 Data Analytics

Owing to its “social” nature, social media sites have become mainstream sources for mining peoples’ thoughts and opinions about certain topics which can be further processed to understand the likes, dislikes, behavioral patterns of users and even predict their following actions.

Depending on the different purposes, data analytics based on OSN can be roughly divided into two categories. One is to detect potential risks so that the relevant parties can act in advance to reduce the possible negative impact.

Gerber [5] applies text analysis and topic modeling to identify particular discussions on Twitter in the United States and incorporates these topics into a crime prediction model. Boni et al. [6] further analyze the spatio-temporally tagged Tweets and their methods can even infer the routine activity patterns of crimes. Another application is done by Fan et al. [7]. The authors propose an innovative model called AutoDOA to detect the opioid addicts from Twitter automatically, which can help understand the behavioral patterns of opioid abuse and addiction. Similarly, Tsugawa et al. [8]

extract some features from the activity history of Twitter users so as to build models for estimating the presence of potential depression.

The other category of applications of data analytics on OSN is to predict the future based on historical data so as to support decision making. Many organizations have benefited from such data analytics based on OSN.

One successful use case is targeted marketing which has been discussed a little bit in the last subsection. Social networks sites have gradually turned into a significant source for sentiment analysis, opinion mining in fields like customer relationship management, customer opinion tracking, etc [9].

In addition, data scientists collect and analyze content posted by a large number of users to understand their opinions about specific topics and attempt to predict real-world outcomes. Some successful predictions include [10] whose authors predict the box office of 24 movies based on Twitter and [11] whose authors predict the results of the German Federal Election, etc.

These are good aspects of online social media. From the content posted, data scientists can predict the future and act in advance or to gain more profits using data analytics.

1.1.2 Bad Aspects of OSN

However, things may start well and end less well sometimes. With the fast growth of OSN, many cyber criminals seize the new opportunity offered by OSN to engage in spam and phishing.

1.1.2.1 Low-quality content

Spam is often considered as junk mail or junk postings on websites, also known as unsolicited email or any message that is unwanted or unrequested by the recipient. Botnets and virus-infected computers are commonly used to send the majority of spam messages, including job-hunting advertisements, promotions of free vouchers, testimonials for some pharmaceutical products, etc [12].

Recently, targets of spam messages have been shifted to some online websites especially OSN which own a large number of active users. The biggest advantage of social spam is its outreach. A spam posted on OSN can reach hundreds of thousands people if it is designed skilfully. What makes the conditions worse is that OSN spam is more difficult to detect because the boundary between spam and some legitimate content is vague.

Phishing can be recognized as a special type of spam intended to trick the recipients into revealing their personal information especially some sensitive ones like login and password details. After obtaining this personal information, the phishers can breach the victims' accounts and commit identity theft or fraud. Considering 70% of Internet users use the same password for all the online services they use, the damage is not limited to one breached account, but all the emails, online bank accounts and so on. This is the reason why phishing is quite efficient compared to other attacks, as the cyber criminals, by repeatedly using the same account information, can access multiple online accounts and manipulate them for their own purposes. Recent research reveals that identity theft affects millions of people every year, costing victims huge amount of time and money in identity recovery and repair.

Like spam, phishing is not limited just to emails. They have become more and more prevalent on social networking sites. These phishing messages look genuine because

the phishing pages look almost identical to a legitimate source, for example, amazon, online banking, schools etc. The page may contain the legitimate organization logo and seems to be sent from the organizations' email addresses. If the users are not careful enough, it is easy for them to reveal their credentials.

Apart from spam and phishing content, OSN also suffer from large amount of low-quality content. User timelines are filled with these content polluters which significantly decreases the overall experience when using the OSN. As the content of real value is overwhelmed by the low-quality content, users are impeded from browsing meaningful and interesting content.

1.1.2.2 Misinformation

The relatively recent development of the Internet in the history of humanity has resulted in a Copernican revolution in the way that information is distributed in society. With the explosion in the popularity of social media platforms, social media platforms and other complementary digital news channels have increasingly encroached upon traditional print media. According to the Media Consumption Report: Q3 2014¹, it is found that people spend 5 to 6 hours online, compared to a paltry 2 to 3 hours spent on the next highest medium, traditional TV. It is also found that out of the 32 countries surveyed, people of 26 countries spent more time online than on traditional media.

Major news streams such as BBC, Al Jazeera, and New York Times have also augmented their presence and reach by utilizing social media platforms, in response to the general trend that people are increasingly relying on digital media as their source of news. Sources of information on the Internet include not only traditional news streams with digital avenues, but also from online discussion platforms such as Quora

¹Media Consumption Insight Report: Q3 2014 <http://insight.globalwebindex.net/media-consumption-q3-2014>

and Reddit, and adhoc discussion platforms such as Facebook and Twitter. These trends are also in line with the general evolution of the Internet towards Web 2.0, an Internet that is characterized by user generated content.

The increasing pervasiveness and influence of digital media have only exacerbated the importance of being able to ascertain the veracity of information on the Internet. In recent history, the Internet's role in spreading information of questionable veracity has been brought to the attention of the key stakeholders of society, such as governments, organizations, companies, and community leaders, recognizing the risks of prevalent false information in society. Fake news has been widely flagged for influencing Brexit and the American presidential election in 2016. A landmark bill by German legislators which, if passed, will compel social media outlets to quickly remove fake news which incite hate or face fines of up to €50m. German officials had provided data showing that Facebook "rapidly deleted" just 39% of the criminal content it was notified about while Twitter acted quickly to delete only 1% of posts cited in user complaints [13]. Countries are increasingly calling for new laws to curb fake news.

The need to devise automated means for the legions of social platform users to verify the veracity of information cannot be more urgent given the speed, scale and reach of the highly connected OSN and the sheer volume of information. With such a technology, government, companies and other organizations can identify fake news at an early stage and nip them in the bud to prevent the potential harm.

What should be noted here is that misinformation looks similar to content polluters at the first glance but they are different in essence. The purpose of misinformation is to convince its readers and manipulate public opinions so as to achieve certain specific goals. Compared to content polluters, they are more like normal posts in styles and formats and are thus more difficult to detect.

1.1.3 Discussions

To summarize, Online Social Media is like a double-edged sword. When used properly, the benefits of it can outweigh its disadvantages to some extent. OSN are changing our world gradually in many aspects which cannot be overlooked. OSN more or less encourage more interaction and communication by sharing users' opinions in a more efficient way. In other words, they act as platforms for the different users' voices. People from all over the world who share the same interests and opinions can easily find one another on OSN. In particular, the author focuses on the exploitation of OSN to improve anomaly detection and prediction accuracies which are two of the most important applications of intelligent analytics.

OSN are both 'curses' and 'blessings' in the new era. Whether they become a 'curse' or a 'blessing' not only rests on the shoulders of their users, but also that of data scientists. This fact has inspired the author to enlarge the benefits brought about by OSN and reduce the potential issues they may cause with machine learning and data mining techniques.

1.2 Motivation and Scope

1.2.1 Content Polluters Detection

Technically, spam or phishing is not a new problem with Internet but when the malicious content is integrated into OSN, they bring about more severe damages.

According to Networked Insights' research, as of fall 2014, 9.3% of content on Twitter is spam [14]. Apart from the spam and phishing content, OSN also suffer from large amount of low quality content including advertisements, automatically generated

content by third party applications, etc. Users are hampered from browsing meaningful and interesting content by the overwhelming amount of low-quality content, resulting in significant decrease in the overall user experience of using the OSN. In some extreme cases, they can even affect the physical condition of some vulnerable users with a syndrome called “Twitter psychosis” [15].

Researchers have paid much attention to the detection of malicious content such as phishing or spam while in contrast little attention has been given to the large quantity of repeated low-quality content which bothers users most. Very little of the previous work is carried out from the users’ perspective. Thus it is important to develop a unified technique to filter all the low-quality content so as to improve the overall user experience instead of focusing on spam or phishing alone. Herein lies the motivation for the research carried out in this work. The reason that the author is using the term “low-quality content” instead of the more familiar term “spam” is because the definition for spam is diverse and is often used to indicate malicious content. However, the malicious content only accounts for a small proportion of all low-quality content. In other words, there are other types of low-quality content besides spam. Hence, to avoid the potential misunderstanding, the author uses the term low-quality content instead of spam.

Considering that there is not a general consensus about the definition of low-quality content on OSN, this adds to the difficulties of detection as well as the evaluation of different detection methods. A further question is whether the features selected for detecting spam or phishing can still be efficient when detecting other types of low-quality content. In addition, even if these features can achieve a high detection rate, can they be extracted in real time? The consideration is once a tweet is posted, it will be delivered to all the followers immediately. Hence, the real-time requirement is quite necessary for protecting and improving user experience when they are using the

OSN. As a matter of fact, most of the current detection work is done offline. Graph features adopted in [16] such as betweenness centrality and the redirection information adopted in [17] are too time consuming, making it difficult to apply them in an online context. The work done by Fu et al. [18] also consumes much time when calculating the carefulness of users' behaviors.

Different from existing research work whose attention is focused on the detection of malicious content such as spam and phishing messages, the research objective of the work presented in this thesis is the detection of low-quality content on OSN which has a wider range than just malicious content. Another highlight the author would like to emphasize is that the features proposed to characterize low-quality content are time efficient to compute which facilitates the real-time detection once the offline training is completed. Based on these time efficient features, the author proposes a real-time content polluter detection framework which can be applied on OSN.

The author carries out a survey to investigate user opinions about low-quality content and based on the survey results, the author provides a clearer definition from the users' perspective. Then the author proposes some features for low-quality content detection and verifies the significance of these features. Both the detection rate and time performance are adopted as the evaluation metrics so as to fulfill the requirement of an online environment.

1.2.2 Rumor Detection

OSN have become one of the most popular ways for daily communication among people. While OSN have significantly facilitated our life, they have also brought about some bad impacts. Among them, false rumors and different kinds of misinformation have become one of the most serious problems which bother both normal users and OSN

service providers.

Social psychology field defines a rumor as a controversial and fact-checkable statement [19,20]. Rumors carrying misinformation will cause severe harm especially under some emergency situations.

Most of the previous research work treats rumor detection as a classification task. One of the earliest work is [21] which groups all these features into four categories namely Message, User, Topic, and Propagation. More recent work further studies the content of the microblogs and incorporates the analysis of LDA features [22] and sentiment factors [23].

An obvious limitation of the previous work is that they just consider the overall differences between rumors and normal posts but not the behavioral differences among users. For example, according to [20] and [19], misinformation receives more skepticism than true information. When a celebrity posts a tweet, he or she always receives a lot of replies (more than 100) and many of these replies include question marks. Can we therefore claim that every tweet of this celebrity is suspicious of being a rumor? On the other hand, a normal user seldom receives a large number of replies from others but receives 10 replies for a tweet and most of them contain question marks. Can we say this tweet is less suspicious of being a rumor because the number of question marks in the comments are not that many?

Therefore, the author argues that it is more meaningful to observe such differences between rumors and non-rumors on an individual user's level. The differences between rumors and non-rumors will be diluted when performing rumor detection if the behavioral differences of different users are ignored.

The author is the first to view it as an anomaly detection task. The theoretical support is that according to [21], the user behaviors of posting rumors diverge from

those posting genuine facts. In order to exploit such differences, the author builds a user behavior model based on the recent microblogs posted by the user within a time period. Since rumors only account for a very tiny proportion of all the posts, most of the microblogs can be regarded as credible posts. Thus rumors are regarded as anomalies.

A pilot work about rumor detection is presented in [24]. The author proposes a PCA-based model to profile individual user's behaviors. However, PCA method assumes a linear system which is not always applicable for rumor detection task. Considering this, the author proposes to substitute the PCA model with the AutoEncoder (AE) which can ignore such assumption and can learn more features of the original data set. The model is able to profile the normal behaviors of a user and represent the deviation degree of every post he or she posts.

Apart from using only cues from microblogs themselves, in this work, the author also exploit crowd wisdom (i.e. other users' comments on the suspicious microblog). Previous research discovers that rumor tweets are questioned much more than credible tweets [25,26]. On top of that, the author proposes comment based features to describe such differences in user behaviors when commenting on rumor posts and genuine posts.

In addition, the time at which these comments are posted implies the diffusion pattern of the original microblog. To take into consideration of the changes of these features over time, the author employs Recurrent Neural Networks (RNN) to analyze the features as inspired by previous research work ([27,28]). Thereafter, all features mentioned above are then input into a variant AE for further anomaly detection.

To determine whether a post is actually a rumor, Chen et al. [24] calculate its rank of deviation degree in its recent posts. This method has a limitation because every recent post of a non-rumor should have similar deviation degree. Thus it is difficult

for the rank to reflect whether a post is a rumor or not. To overcome the problem, the author proposes several self-adapting thresholds to help facilitate rumor detection and discusses the experimental results accordingly.

1.2.3 Stock Index Prediction

Efficient Market Hypothesis (EMH) [29, 30] and Random Walk Theory (RWT) [31] have discouraged early research on stock prediction. This is because EMH indicates that market prices should only react to new information or changes instead of the past or present prices and that news follows a random walk pattern which defies prediction. Therefore, it is almost impossible for one to consistently achieve returns exceeding average market returns. However, the assumptions behind the theory cannot be met all the time and even the author himself revises the theory to incorporate 3 levels of efficiency. Since then, EMH has been challenged by many researchers in behavioral economics [32, 33], behavioral finance [34] and so on [35, 36].

Adaptive Markets Hypothesis (AMH) [37] is then proposed to reconcile EMH with behavioral finance. Urquhart et al. [38] further improves the AMH to describe the behaviors of stock returns in a better way. The results illustrate that the price of the stock market does not follow a random walk and thus can be predicted to some extent [39–41].

Since then, stock market prediction has attracted much attention in many research fields. These attempts range from technical analysis to fundamental analysis. Technical analysis relies more on feature engineering with the purpose of selecting highly discriminative technical indicators. With the development of machine learning, fundamental analysis is attracting the attention of more and more researchers. Fundamental analysis usually involves the processing of unstructured data. Some of the possible sources

include financial reports, official documents and even online news and discussions [42].

Many researchers have used news analytics and sentiment analysis to predict the stock price [43, 44]. Others analyze public moods by crawling public posts on OSN like Twitter, Weibo or other forums [45–47] to facilitate stock volatility prediction. According to a recent survey conducted by Pew Research Center, 67% of Americans get at least some news from social media [48]. This indicates that OSN has worked as a more and more important platform for spreading news. This inspires the author to utilize the news content posted on OSN for the analysis of public sentiment.

There are many differences between the news reported by traditional media and online social media. News from social media is usually more succinct and easier to spread. This makes it easier to have a large number of readers. These features turn OSN into a better source for analyzing sentiment and other content features for stock movement prediction. In this work, the author carefully selects official accounts with a large number of followers and analyzes the news content they post. Apart from sentiment features, the author further computes the LDA features which are rarely used in previous research for stock index prediction.

In addition, lots of previous research work has been done to predict the U.S. stock market [49] while not so much attention has been paid to the Chinese stock market given that China is the second largest economy in the world. Therefore, the author carries out the research work to analyze news content from Chinese OSN and predicts the Chinese Stock Index (CSI), i.e. the Shanghai-Shenzhen 300 Stock Index (HS300).

Nassirtoussi et al. and Garcke et al. [42, 50] report that a prediction accuracy above 55% can already be considered as a report-worthy result, which the author believes, is still far from satisfactory. Given that the performance of machine learning models in such application has been relatively weak at only slightly better than random guess-

ing, the author is inspired to adopt Adaboost to improve the prediction performance. Based on this, the author proposes a hybrid model named RNN-Boost which achieves promising experimental results.

1.3 Methodology

This thesis demonstrates methodologies and applications in detection and prediction using content collected from OSN. The research methodologies adopted in this work are as follows:

- Firstly, the author conducts comprehensive literature reviews on previous research work using OSN with a focus on detection and prediction tasks.
- Secondly, the author identifies the particular unsolved challenges from previous research work.
- Thirdly, the author applies data mining and machine learning techniques and proposes corresponding frameworks and models to tackle the aforementioned challenges. Feature engineering is also employed to facilitate the detection and prediction purposes for the proposed frameworks and models.
- Fourthly, the author runs extensive experiments to evaluate the proposed schemes and compares the results with previous work.

1.4 Major Contributions

The author has made the following contributions to address the content detection and prediction tasks identified in Section 1.2.

1.4.1 A Real-time Low-quality Content Detection Framework

The author proposes a real-time low-quality content detection framework which can be applied on real OSN. Details of the contributions are listed as follows:

- The author performs EM algorithm on harvested low-quality content tweets to divide them into 4 categories. Based on the preliminary classification results, the author creates a survey with 211 participants to study their opinions about low-quality content. The author then provides a clearer definition of low-quality content on Twitter based on the survey results. At the point of research, the author has not come across any similar preliminary studies conducted from the users' perspective.
- The author crawls and manually labels 100,000 tweets so as to verify the correctness of the preliminary studies and the proposed definition of content polluters. Examples of tweets and labeling guides are provided so as to make the experiments replicable.
- The detection techniques for malicious content on OSN are quite mature but little attention is paid to other types of low-quality content such as low-quality advertisements and automatically generated content which actually bothers users most. Thus the author unifies the detection of different types of low-quality content and provides an in-depth study of the features commonly used for detection of malicious content to understand their applicability for other low-quality content.
- The author provides a word level analysis on the original tweet texts and builds a keyword blacklist dictionary to facilitate low-quality content detection. The author is the first to build the dictionary to help detect low-quality content.

- The author applies traditional classifiers (SVM and random forest) based on the proposed dominant features as well as word features for real-time low-quality content detection and it achieves a high accuracy and F1 measure as well as good time performance.

1.4.2 An Unsupervised Rumor Detection Model based on Users' Behaviors

The author proposes an unsupervised rumor detection model based on users' behaviors.

The contributions include:

- The author exploits crowd wisdom to perform rumor detection on OSN. To be more specific, the author proposes new features extracted from the comments of the suspicious microblogs to help improve detection performance.
- The author further uses RNN to study the comment based features over time to better capture the differences between rumor posts and credible posts.
- The author's model is based on individual user's behavior and the author views rumors as anomalies in the recent posts by that user. In other words, the author treats rumor detection as an anomaly detection task.
- The author, for the first time, adopts Autoencoder to detect rumors on OSN and experiments show that the proposed model can achieve a good accuracy and F1 measure.
- The proposed model is unsupervised and thus does not need labeled data which makes it especially useful when rumor data are difficult to obtain for training.

1.4.3 RNN-Boost - A Hybrid Model for Predicting Stock Market Index

The author proposes a hybrid model named RNN-Boost to predict stock market index based on the collected news content from OSN. Details of the contributions are as follows:

- The author exploits news content from online social media instead of traditional news media for sentiment analysis to facilitate the prediction of stock index.
- The author further proposes LDA features to improve the performance of the prediction model and combines them with sentiment features and other technical indicators.
- The author, for the first time, proposes a hybrid model incorporating RNN and Adaboost to predict stock index and experiments show that the proposed model can achieve a good prediction performance.

1.5 Thesis Organization

This thesis discusses the good and bad aspects of OSN and how to break the ‘curse’ while enjoying the ‘blessing’. The focus is on the mining of OSN to improve anomaly detection and prediction accuracies with use cases. Chapter 2 introduces and discusses the previous work. The following two chapters present details on how to combat the potential issues caused by OSN which address the use cases, namely, low-quality content detection and rumor detection respectively. Chapter 5 gives an example of utilizing the content collected from OSN to boost the accuracy of predictive analytics using stock market index prediction as the use case. The last chapter concludes the whole thesis

and provides directions for future work.

In summary, the rest of the thesis is organized as follows:

- **Chapter 2** reviews the related work on content polluter detection, rumor detection and stock market index prediction.
- **Chapter 3** presents the details of a real-time low-quality content detection framework from the users' perspective. The author first conducts a survey to understand users' opinions about content polluters and gives a clearer definition of them. Direct and indirect features are proposed to distinguish content polluters from normal posts and they can fulfill real-time requirement at the same time. The performance of the framework is evaluated and the significance of the different feature subsets is also discussed.
- **Chapter 4** introduces an unsupervised rumor detection model which combines AutoEncoder (AE) and Recurrent Neural Networks (RNN). The author views the rumor detection task as an anomaly detection problem. Therefore, AE is adopted to detect the rumors. In addition, RNN is employed to analyze the comments on potential rumors so as to capture the spread pattern which in the end facilitates the recognition of rumors. The variants of the model are discussed and the comparisons of performance are presented as well.
- **Chapter 5** describes a hybrid model named RNN-Boost for stock market index prediction. The sentiment features and LDA features are combined with technical features to perform stock market index prediction. The significance of the proposed features are evaluated. The performance of the individual RNN and RNN-Boost are also compared to demonstrate the improvement.
- **Chapter 6** concludes the whole thesis. Future work to expand the proposed

frameworks and models are presented. In addition, the author further lists the future research directions on how to analyze the information collected from OSN to better serve the individuals and the society.

Chapter 2

Literature Review

This chapter provides the literature review for each of the topics outlined in Section 1.2. Prevalent approaches, their advantages as well as limitations are discussed. How the author's work is different from the previous research work is highlighted as well.

2.1 Low-quality Content Detection

In the last decade, the growth of OSN has provided a new hotbed for spammers and phishers. Significant efforts have been paid to analyze and detect the malicious content on OSN websites like Facebook, Twitter, etc.

2.1.1 Definition of Low-quality Content

Spam on OSN (also known as social spam) is usually regarded as a message which is unsolicited by legitimate users [51]. However “unsolicited” is quite a vague description. Different research work has different definitions for spam and phishing. Yang et al. [16] define tweets containing malicious content as spam and do not regard advertisements as

spam. Thomas et al. [52] and Sridharan et al. [53] label a tweet as spam if the account is suspended by Twitter in a later validation request. However, the definition in [16] is closer to that of phishing while [52] and [53] also have drawbacks as Twitter itself initially only focuses on spam or phishing according to Twitter Rules [54] while showing generosity to mainline bot-level access and some advertisements as long as they do not break Twitter rules [55]. Twitter has recently introduced a quality filter which aims to filter out low-quality content [56]. This testifies to the usefulness of the author’s work. It is to be noted that Twitter’s quality filter is applied on the notification timeline (i.e. tweets mentioning the user) while the author’s work is applied on the user’s home timeline (i.e. all the tweets of the user’s followees). In other words, only tweets mentioning the user will be processed by Twitter’s quality filter while the method proposed in this thesis does not have such limitations. From Twitter policy, it can be concluded that accounts which persistently post low-quality content are less likely to be suspended. Moreover, account suspension may not only be due to the delivery of spam, thus making the judging yardstick even less convincing.

One thing in common among these definitions is that they try to characterize the features or behaviors of the unsolicited content themselves instead of defining them from the users’ perspective. In addition, not much work is focused on low-quality content detection. They either focus on simplex spam or phishing detection instead of proposing a unified detection technique which also aims at other low-quality content. Lee et al. [57] first propose the term “content polluters” and divide them into several categories. However, in their work, the term “content polluter” is used to refer to spam accounts while the author uses “low-quality content” to refer to tweets which contain only valueless and trivial content in this thesis. Their work and the work in this thesis actually represent the two mainstream methods adopted, namely, user based methods and tweet based methods respectively.

2.1.2 Spam Detection

The problem of undesired electronic messaging has become a more and more serious issue in the last few years. According to [58], the percentage of spam in emails is 66.76%. These large proportion of email spam has caused different problems and some of them even leads to economic losses. From the perspective of service providers, these losses include waste of storage space, computational power as well as bandwidth. While from the perspective of users, these unwanted electronic messages waste users' time and lower their work productivity. Some of the users even claim that the spam makes them feel irritated and violates their privacy. In this section, the author provides a structured overview of the existing countermeasures against spam in both emails and OSN.

2.1.2.1 Content feature based filters

The first efficient method applied to spam filtering system is the Bayes algorithm which is regarded as one of the most prevalent spam filtering methods. Naive Bayes Classifier (NBC) has been proven successful as the solution for different types of tasks especially for text classification. Text content is usually transformed to a bag-of-words model. Some keywords are selected and the presence or absence of these words is used to represent the characteristics of a text message. If a particular term t_i is present, the corresponding weight of the characteristic vector will be $w_i = 1$, otherwise $w_i = 0$. The NBC is then applied to this characteristic vector and the classification task is performed.

A similar method, k-nearest neighbor, is proposed in [59] for spam filtering. The label of the message is decided by the majority among the k nearest training samples which are selected according to a predefined similarity function. Another popular classifier commonly used is Support Vector Machine (SVM) [60]. The features of

training samples are extracted and projected to a high-dimensional space so that the two classes can be separated through a hyperplane.

Considering the content of spam is written in natural language, methods performed well in text classification are adopted to perform spam filtering task. One of them is to calculate Chi value by degree of freedom [61]. In this method, the original message is exploded into several n-grams in terms of words or characters. The processed message is then compared with labeled messages processed with the same pre-set using Chi value.

2.1.2.2 Non-content feature based filters

Apart from the body of the emails, other non-content features in header and meta-level features are also extracted for the purpose of structured analysis. Leiba et al. [62] analyze IP addresses in the reverse-path and assign reputation to them based on the number of legitimate and spam emails delivered by them. Boykin et al. [63] analyze and leverage the social networks of senders and recipients of emails to fight with spam. Through the From, To, Cc and Bcc fields in the headers, the authors are able to construct a social graph of these users for classifying new messages. The above two methods can be regarded as a variation of whitelists or blacklists.

Another research direction is to analyze user behaviors. The behavioral patterns of a given message are extracted and compared with a set of predefined behaviors of normal users and spammers to make the final judgment. Hershkop et al. [64] propose a set of behavior models among which past activities of users and recipient frequency show discriminative power for spam filtering.

To achieve better spam filtering rate, the collaboration with users is also adopted. The knowledge of spam is shared through gathering spam reports on a server. [65]

proposes a privacy-preserving method to P2P spam filter. In their filtering system, spam reports are sent leaving out the source of the report which protects user privacy. [66] describes a multiple agent system, in which every message is initially labeled as legitimate, spam or suspicious by a local agent. After that, only those suspicious ones need collaborative judgment from users.

These non-content features usually can help improve the detection performance but may not always fulfill the real-time requirement as some of them request long time to compute or collect/analysis.

2.1.3 Phishing Detection

Phishing exploits vulnerabilities of OSN users. According to [67], phishing attacks attempt to lure users to perform certain actions, on most occasions revealing their personal information, for the benefits of the attackers. In this section, the author provides a literature review about the existing anti-phishing techniques which can be divided into two categories, one is to raise user awareness, and the other is software detection solutions. Human factors are very significant during phishing detection as according to [68], end users fail in detecting 29% of phishing attacks. However, the topic is beyond the scope of this thesis as the focus is on software solutions.

2.1.3.1 Blacklists

Blacklists are lists of previously detected phishing keywords, Internet Protocol (IP) addresses or URLs and are updated frequently. A variation of blacklists is called whitelists which, on the contrary, record those keywords, IP addresses or URLs verified to be legitimate. The application of whitelists is for the purpose of reducing false positive rates.

One of the most commonly used blacklists is Google Safe Browsing API ¹. The two most prevalent browsers, Google Chrome and Mozilla Firefox, both use Google Safe Browsing API as an in-built function to protect their users from phishing attacks. The API requires its clients to send a request containing the suspicious URL, and the server will respond on whether the given URL exists in the blacklists maintained by Google.

DNS-based Blacklist (also known as DNSBL) leverages on the DNS protocol, making it easy for any DNS server to act as a DNSBL. For each inbound SMTP connection, the DNSBL will verify whether the current source is listed in the phishing blacklists.

Cao et al. [69] propose an automated individual white-list (AIWL) to detect phishing URLs. The whitelist includes a set of features depicting authentic Login User Interfaces (LUI) in which the user submits their credential information. Each LUI will trigger a warning message unless it is trusted, in other words, listed in the AIWL.

However, there is a critical weakness with blacklists in that there is always a time lag between the post of the phishing content and the indexing of it into a blacklist. This problem makes blacklists inefficient in detecting zero-hour phishing attacks. As studied in [70], blacklists are only able to detect about 20% of zero-hour phishing attacks. Although 47% to 83% phishing URLs are indexed in blacklists after 12 hours, a noteworthy issue is that the lifespan of 63% of phishing campaigns only last for 2 hours.

2.1.3.2 Heuristics

Phishing heuristics are features found existing in real phishing campaigns. What is worth mentioning here is although there exist some patterns or heuristics, they are not ensured to happen in every phishing attack. However, it is still safe to conclude

¹Google Safe Browsing: <https://developers.google.com/safe-browsing/>

that when a set of heuristic tests are identified, it is likely to detect phishing attacks not seen previously (zero-hour phishing attacks), which fills the gap of what blacklists cannot fill.

The work in [71] acknowledges that phishing websites usually store user credentials but not verify the correctness of them. Based on this discovery, the authors develop a browser plug-in called PhishGuard. PhishGuard will first send the right user ID and wrong passwords to the suspicious page. If it returns a HTTP 200 OK message, the page will be regarded as phishing. If it returns HTTP 401 Unauthorized, another request carrying the right user ID and passwords will be sent again. If the page returns HTTP 401 Unauthorized again, then it is possible that the site is phishing.

Phishwish puts forward a set of heuristic rules to determine if a message is phishing or not [72]. This solution is also efficient in detecting zero-hour phishing attacks and it only requires minimal computational resources (only 11 rules). The score for a suspicious message is represented by the weighted mean of the rule set. This score will be compared with a predefined threshold to give the final prediction.

CANTINA proposed in [73] is a toolbar developed for Internet Explorer that predicts whether a web page is phishing by analyzing the content of it. During the detection phase, Term Frequency-Inverse Document Frequency (TF-IDF) of each term presented in the page is calculated. The top 5 terms with the highest TF-IDF value will be submitted to a search engine. If the current page is included in the top n returned results, the site will be labeled as legitimate, otherwise it is a phishing site.

The drawbacks of heuristics is that they require experts to conclude such domain knowledge which may be rather time consuming. In addition, these rules tend to evolve over time. Past rules would be invalid after a certain time period.

2.1.3.3 Data Mining

Phishing detection problems can be seen as a classification or clustering task. Thus, algorithms in machine learning or data mining field, such as Support Vector Machine, C4.5, density-based clustering, K-means, can be applied to anti-phishing tasks.

Authors in [74] present an algorithm which can automatically classify large-scale pages and this method is later used to facilitate Google Safe Browsing API service. Similar to PhishTank ², this method also requires human involvement but much less. URLs appeared in junk mails manually classified by GMAIL users will become candidates if the number of this URL exceeds a threshold. Features of these candidate URLs will be extracted for the following classification task. The features extracted include non-content based features like IP address, number of sub-domains etc. as well as content-based features like whether password fields are contained, the existence of high tf-idf terms etc.

Likarish et al. [75] develop a Firefox toolbar for anti-phishing using Bayesian classification. They also use a whitelist to reduce false positives. Stone et al. [76] adopt Natural Language Processing (NLP) techniques for intrusion detection systems including phishing attacks. The highlight of their work is that the messages are processed by the proposed system based on semantics. The core module for detecting mechanism relies on OntoSem to deploy NLP techniques.

2.1.4 Low-quality Content Detection

This subsection provides details of some previous studies of low-quality content detection especially focused on social networks.

²PhishTank: <https://www.phishtank.com/>

2.1.4.1 Content-based methods

Content-based information includes the messages posted by users as well as the user account information which can be directly derived via APIs provided by the social websites.

Tweet based features can usually be divided into three groups. They include tweet content, tweet sentiment and tweet semantics. Tweet content features are usually calculated by counting the number of specific words, symbols or punctuation in tweets [55]. Tweet based features adopt in [77] further include number of #tags, number of @tags, etc. Then they apply Naive Bayes, Decision Trees and Random Forest with the proposed features to perform the detection task.

[78] also uses similar features with [77] but develops more on tweet texts. They choose 95 characters from the complete ASCII set that can be accessed on a standard keyboard and count the number of occurrences of these characters in a tweet. These 95 one-gram features together with other features are combined for later processing. They treat the spam identification problem as an anomaly detection problem instead of a classification task. They develop a variation of the density based clustering algorithm so as to detect those abnormal tweets which diverge from the normal model trained before.

Tweet sentiment features are calculated by using some sentiment lexicons plus sentiment analysis [79]. Tweet semantic features exploit the NLP techniques to facilitate low-quality content detection on OSN. Santos et al. [80] apply compression-based text classification methods to avoid good words attack and improves detection performance for spam tweets. Yang et al. [81] extracts text information from both the web pages and the tags. Then the authors measure the relatedness between the two so as to detect tag spam.

2.1.4.2 Non-content based methods

Lee et al. [57] systematically divide user based features into four groups for the first time. The authors adopt four feature sets, including User Demographics (UD), User Friendship Networks (UFN), User Content (UC) and User History (UH). Then they use different classifiers to perform spammer account detection based on their proposed features.

The first category of non-content based method relies on the URLs which appear in the messages. These messages posted on social networks usually contain URLs which are similar to that of emails. However, there is a difference in OSN context that is online social websites limit the length of the posted messages. Thus, these URLs appear on OSN are often shortened using some URL Shortener like Bitly and Google URL Shortener. [82] exploits the click through data for detection of spam and spam accounts. [77] also leverages on some WHOIS³ based information of the URL presented in the tweet.

The second category of non-content based methods observe the behaviors of both legitimate users and abnormal users. Grier et al. [82] develop two tests to characterize user behaviors. The first measures tweet timing with the assumption that the tweets from normal users roughly follow a Poisson distribution while spammers do not. The latter analyzes the entropy of users' tweeting history to see whether they keep posting similar texts or links.

The third category of non-content based methods analyze the social community formed on social websites. [83] analyses the social interactions between users. These social interactions include the follow and mention relationships. They then illustrate these relationships with relationship graph with the purpose of characterizing spam

³WHOIS is a protocol widely used for querying databases that store the registered users or assignees of an Internet resource (e.g. a domain name, an IP address block, an autonomous system)

tweets as well as spam accounts.

What is worth mentioning here is the term “low-quality content” used in this thesis should be distinguished from “content polluter” used in Lee’s paper [57]. In their work, the term “content polluter” is more similar to a spammer’s account whereas the author refers to low-quality content as a piece of **tweet message** of little value or importance to the users and may erode the users’ experience. The intuition behind is that even a normal user may post low-quality content with or without intention and a malicious user may also post normal messages to avoid the suspension by Twitter.

2.1.4.3 Real-time detection

Some research work has focused on real-time requirement for the detection of spam and phishing. Aggarwal et al. [77] develop a plugin for browsers so as to implement automatic real-time phishing detection on Twitter. Song et al. [84] exploit the sender-receiver relationship. When a user receives a message from a stranger, the system can identify the sender at once therefore ensuring that the clients can identify the spammers in real time. Tan et al. [85] propose a runtime spam detection scheme known as BARS (Blacklist-assisted Run-time Spam Detection) which exploits the different behavioral patterns between normal users and spammers as well as a spam URL blacklist.

A more recent piece of work [55] exploits the inherent features of Twitter. All the features they use can be directly extracted or calculated which meet the real time requirement. However, again the aforementioned studies either focus on spam detection or phishing detection but do not provide a unifying solution for all low-quality content detection especially for the large amount of low-quality advertisements, meaningless content, etc.

2.1.5 Discussions

To summarize, blacklists and classifiers are most commonly used for spam and phishing detection. However, they have the following drawbacks:

- There is a time lag for blacklists based methods.
- Real time requirement is not widely considered among previous research work.
- It lacks a unified framework to detect all spam, phishing and other low-quality content.

2.2 Rumor Detection

Many researchers in computer science field have shown interest in rumor detection on OSN in recent years. In the following section, the author will introduce the related work on feature selection for rumor detection tasks. In addition, the author also summarizes some highlights of detection methods in existing research work.

2.2.1 Studying Rumors

2.2.1.1 Definition of Rumor

Before introducing the research work about rumor detection, the author would like to discuss about the definition of rumors first. Basically, there are two major opinions about the definition of rumors. Some research work defines a rumor as a piece of information which is false [86,87]. However, more researchers favor the definition that a rumor is a controversial and fact-checkable statement in circulation [20].

The author hereby supports the latter definition as it is consistent with the definition in social psychological field and the explanation in major dictionaries like Cambridge Dictionary which defines it as “an unofficial, interesting story or piece of news that might be true or invented, and that is communicated quickly from person to person”⁴ and Collins Dictionary which defines it as “a story or piece of information that may or may not be true, but that people are talking about”⁵.

2.2.1.2 Taxonomy of Rumors

According to the definition above, a rumor can be further divided as a true rumor or a false rumor. This is because after the rumor is circulated for a while, it may be verified as a genuine fact (i.e. true rumor) or be debunked as misinformation (i.e. false rumor). Comparing to the true rumors, false rumors usually cause more negative social effects, thus the focus of most research work is on the detection of false rumors. Another similar classification method is based on the level of veracity (i.e. high or low). In this case, the confidence of a rumor is not asserted but given a credibility score.

Apart from credibility, another taxonomy of rumors introduced in [88] is based on active cycle. Rumors are divided into short-term rumors which emerge during breaking news and long-term rumors which are discussed for a long time period. The detection of long-term rumors usually rely on keywords filtering while that of short-term rumors are more difficult. Therefore, more research efforts are given to short-term rumor detection.

⁴<https://dictionary.cambridge.org/dictionary/english/rumor>

⁵<https://www.collinsdictionary.com/dictionary/english/rumour>

2.2.2 Theories Related to Rumors

According to Allport and Postman, the Basic Law of rumor [89] consists of the following equation:

$$R \approx a \times i \quad (2.1)$$

where R represents the strength of a rumor, i represents the importance of the content of the message to a given individual and a represents the ambiguity of evidence of the content. While the foregoing assertion alone may not have the academic rigor for it to be credible, the formula would broadly account for the pervasiveness of fake news. The research findings in [90] suggest that rumors and urban legends thrive on information and emotion selection and could account for the huge amount of the fake news during the period of the US Presidential Election in 2016, where key trending topics included tweets which were crafted to invoke disgust against their political opponents. Examples include invoking feelings of disgust against Hillary Clinton due to her suspected corruption, or against Donald Trump due to his inexperience in political office or against his sexist stances.

2.2.2.1 User Behaviors

It was not until recent years that researchers in computer science started to focus interest on rumor detection especially on OSN. However, rumors and relevant phenomena have been studied in social and psychological fields for a long time. Some previous work in these fields have addressed the differences of user behaviors related to fake news posting and genuine news posting.

According to [20] and [19], people express less skepticism to genuine news and their interest in the genuine news fade away along with time which is unlike fake news that

many people recursively retweet them within a time window. In addition, the doubts about rumors raised by the readers or audience usually lead to a different comment style in rumors compared to genuine news [21].

In addition, sentiment polarity in rumors and non-rumors are also different. Rumors usually contain stronger characteristic sentiments (e.g. anger) compared to non-rumors. It is also commonly believed that most of the rumors contain negative sentiment.

2.2.2.2 Propagation

Examining rumor spread pattern is almost impossible before the emergence of OSN. Therefore, previous studies about rumor propagation in psychology are relatively theoretical and are based on small case studies.

Kwon et al. [91] summarize some insights of existing research on rumor propagation. A rumor can remain a hot topic within a short time period and the rumor statements are usually very ambiguous. In addition, rumors spread more easily and widely in a sparse social network. In other words, denser network structures have higher resistance to rumors than sparser network structures.

More studies about rumor spread on OSN have been carried out recently. Wang et al. [92] describe the important cascading shapes of information propagation from Weibo and Twitter. They detect rumor propagation patterns in streaming trending topic data based on user opinions (i.e. support, deny, question). According to their experimental results, rumors and genuine news show different spread patterns and such properties can be used to distinguish the two types of information.

Yang et al. [93] also examine the topology property of the user network for rumor detection. Their experimental results indicate that the relationship between the au-

thor and the audience of the post actually provides a negative contribution to rumor debunking. Moreover, a bigger size of a cluster (i.e. network structure) would lower the independence of judgment.

These previous work inspires the author to exploit such differences in the creation and spread of fake and genuine news to distinguish them automatically.

2.2.3 Feature Selection

The rumor detection task on OSN originated from the analysis of information credibility. One of the earliest work in computer science field is [94]. The authors propose several features and group them into four classes - message, user, topic, and propagation. Message features can be both Twitter dependent or independent and both of the features characterize the text content of the original tweets. User features take the characteristics of the user profile into consideration. Topic features are calculated based on the message and user features consider the posting history of the user. Propagation features characterize the propagation tree by rebuilding the retweets network.

Other research work has extended this idea by proposing their own properties and features. Zhang et al. [95] employ implicit features and combine them with shallow features for detecting social rumors. Some of the novel implicit features they propose include internal and external consistency, social influence as well as match degree of messages.

Yang et al. [96] propose an innovative method to identify rumors automatically based on a combination of hot topic detection and bursty term identification. Specifically, they introduce a term weighting scheme which characterizes both frequency and topicality of terms for identifying bursty keywords. In addition, they demonstrate a sentence model which further considers entities to facilitate the rumor detection.

Ito et al. [22] conduct a study on how people judge the credibility of tweets. Based on the aforementioned studies, they propose a method which can assess the credibility of information on Twitter by utilizing the “tweet topic” and “user topic” features which are computed based on Latent Dirichlet Allocation (LDA). Moreover, they describe two additional features which are based on the expertness and the bias of users using two relevant hypotheses.

Kawabe et al. [23] propose a method to analyze the information credibility based on sentiment analysis and opinion mining. The credibility of the information is evaluated by computing the percentage of similar opinions on a specific topic. For topic identification, they build topic models based on LDA. For opinion polarity classification, they employ Takamura’s sentiment orientation dictionary and carry out sentiment analysis.

Liu et al. [97] further include the streaming features of OSN and propose a real time rumor debunking method. They utilize crowd wisdom by aggregating the tweets’ comments and investigative journalism of users. They first identify rumor events which attract a large number of relevant posts. Then they present a method to understand users’ opinions, derive underlying belief and other meta features to detect rumors.

The authors of [91, 92] identify the differences in the spread patterns of rumors and genuine facts which are introduced in the previous subsection. Frigger et al. [98] carry out studies on rumor cascades. They discover that rumor cascades have a deeper structure in the social network than that of reshare cascades. In addition, they find that if a reshare receives a debunking url, it is more possible for the cascade to be deleted in the following time periods. Finally, rumors themselves evolve over time with different dominated bursts in popularity. These findings provide some insights for rumor detection from another perspective.

However, the aforementioned features are mainly extracted from the original mi-

croblog itself or the user. There is not much research focusing on features based on comments. Cai et al. [86] try to analyze the crowd responses like comments and reposts of the original microblogs but they only consider the word features. Zubiaga et al. [99] observe the rumorous conversations on OSN and propose 4 related features but do not actually perform detection tasks. In this thesis, the author proposes features based on comments to fill in the gap so as to improve the detection performance.

2.2.4 Detection Methods

2.2.4.1 Classification

Most of the previous work mentioned above treats rumor detection task as a classification problem. Therefore, many traditional classifiers are applied to tackle the challenge.

Yang et al. [100] train an SVM classifier on datasets collected from Sina Weibo to automatically detect the rumors. Sun et al. [101] specially target on rumors about social events which they believe would bring more negative social effects. They divide these event rumors into 4 different categories and propose a method to identify one major type which is called text-picture unmatched rumor. In their experiments, they train different classifiers including Naive Bayes, Bayesian Network, Neural Network and Decision Tree.

Liang et al. [87] observe that users' behaviors for posting rumors are different from normal users and the responses to rumors diverge from that of normal posts. Based on the observations, they propose 5 behavioral features and train their classifiers on Logistic Regression, SVM, Naive Bayes, Decision Tree and KNN respectively.

Vosoughi [102] tackles the task by characterizing rumors using linguistic, user ori-

ented and temporal propagation features. Unlike previous methods, they employ machine learning algorithms like Dynamic Time Wrapping (DTW) and Hidden Markov Models (HMM). The author performs the experiments on the datasets built from Twitter and the performance of the models are compared. The experimental results demonstrate that HMM can have a better detection performance than DTW. The author further evaluates the features proposed and discovers that the most significant ones are temporal propagation features.

Among these previous work, what the author would like to emphasize is the work done by Kwon and Ma et al. who incorporate a time series fitting model into traditional classifiers in their serial research work [27, 91, 103] and try to capture the variation of these features over time. A more recent piece of work [28] presents a RNN-based model to better understand the variation of aggregated information across different time intervals. However, their RNN-based model only processes and uses the words appearing in microblogs as features and does not consider other features.

2.2.4.2 Anomaly Detection

In data mining, anomaly detection is a technique used to identify items or observations which do not conform to an expected pattern in a data set. It has already been used in the context of OSN in some research work.

In [104], for the first time, Miller et al. treat the spam detection task as an anomaly detection problem and propose a variation of clustering methods to detect spam from normal tweets. In [105], Guzman et al. exploit the anomaly detection idea and propose a scalable and fast online method to address bursty keyword detection problem. These methods all achieve good performance compared to traditional classification methods.

The requirement of detecting spam and bursty keywords is similar to that of ru-

mor. This inspires the author to apply anomaly detection technique to address rumor detection problem. Another advantage of anomaly detection methods is most of these methods are unsupervised. Since rumors only make up a small proportion of all the posts, it is challenging to obtain enough labeled data for training. For rumor detection field, only the author's previous work [24, 106] view it as an anomaly detection problem. The underlying assumption is that for an individual user, his or her posting style remains consistent within a period of time. When the user posts a rumor, the posting behaviors deviate from the normal ones and thus can be regarded as anomalies.

In one piece of previous work [24], the author uses Principal Component Analysis (PCA) to perform feature selection and the results are then combined with the proposed strategy (Euclidean Distance and Leave One Out) to detect anomalies but even the best accuracy is not very high (82.28%). Based on [24], the author further proposes a different distance-based rumor detection strategy and achieve slightly better result. However, the issue with PCA is that it is premised on a linear system which is not applicable all the time.

In this thesis, the author applies AutoEncoder (AE) to achieve the anomaly detection purpose. AE does not have to assume a linear system and the hidden layer in an AE can be of greater dimension than that of the input whereby the data in the new feature space can be disentangled from the hidden factors of variation. Therefore AE can usually learn more features of the original data, in other words, perform better than PCA in feature selection which contributes to a higher detection rate of rumors.

2.2.5 Discussions

To summarize, traditional classifiers are most commonly used for rumor detection task. Considering that rumors only account for a small percentage of all posts, the dataset

can be extremely imbalanced which would have a negative effect on the detection performance. Therefore the author proposes to apply anomaly detection method instead of classification.

In addition, a lot of previous work have paid much attention on behaviors and propagation which has proven to be significant in rumor detection. Therefore the author has incorporated these features as part of the inputs into the proposed model. However, not much attention has been paid to the analysis of the comments on potential rumors. The few relevant work only use bag-of-word model without using more advanced NLP techniques. The author suggests this should be a direction which needs further exploration.

2.3 Stock Index Prediction

2.3.1 Market Efficiency

2.3.1.1 Definition

Stock market prediction has always been a popular and challenging task in financial time-series forecasting. However, is it really possible to predict the stock market? Early research work about stock prediction was carried out based on Efficient Market Hypothesis (EMH) [29,30] and Random Walk Theory (RWT) [31].

According to EMH, an asset's prices fully reflect all available information. In other words, market prices should only react to new information or changes instead of the past or present prices. Considering that news always follows a random walk pattern which defies prediction, it is nearly impossible to predict stock price with an accuracy of more than 50%. In other words, the future price of a stock is as unpredictable as a

sequence of random numbers. In statistics, it is concluded from EMH that the future prices of stock is completely independent from the past prices, that is, the sequence of stock prices has no past memories.

If EMH and RWT hold, any effort that investors dedicate to predict the stock market is meaningless. This is because it is impossible for one to beat the market benchmark consistently. Even though high return may be achieved sometimes, they are always accompanied by high risks accordingly.

2.3.1.2 Limitations

These facts have discouraged the researchers and investors from predicting the stock market. However, what should be noted here is that there are two assumptions lying behind both theories. Firstly, the investors are rational when they make decisions to buy or sell the stock. Secondly, the complete information about the stock market is available to all investors.

However, in the practical stock market, the two assumptions do not always hold and critics rise to blame the belief in “rational markets” especially during the late 2000 financial crisis [35,36]. Even Fama himself who proposed EMH revised his theory later to further distinguish 3 levels of efficiency which are strong, semi-strong and weak. He conceded that his theory is more applicable in markets whose information is more transparent to all market traders and investors.

A large amount of research studies have examined the EMH and RWT from aspects of behavioral economics [32], behavioral finance [34] and so on. The results illustrate that the price of the stock market does not follow a random walk and thus can be predicted to some extent [39–41]. This indicates the possibility of predicting some weakly efficient markets and has encouraged more researchers to further explore the

performance of different machine learning methods in predicting the financial market.

2.3.1.3 Adaptive Market Hypothesis

The discussion about the applicability of EMH on different financial markets is still ongoing with very different results and conclusions. Lo et al. [37] propose a theory named Adaptive Market Hypothesis (AMH) which tries to reconcile EMH with Behavioral Finance. In a more recent research work, Urquhart et al. [38] have verified the effectiveness of AMH in US, UK and Japanese financial markets over a long time period. Their experimental results show that AMH is able to better describe the behaviors of stock price movement than EMH.

The core of EMH is that the market will absorb the emerging new information and adjust accordingly to achieve new equilibriums and become efficient again. The process occurs consistently which makes the previous predictive standard invalidated. Nevertheless, past observations about financial markets indicate that the investors are not always rational and cannot always respond quickly to new information. The financial market therefore becomes less efficient and it is possible to predict the stock market to some extent. For example, Poti et al. [107] discover there exists predictability in the foreign exchange market (FOREX) in their experiments.

According to Fama's revised EMH [29], some financial markets are more predictable than other markets. Yu et al. [108] has discovered in their research work that the financial markets of developing countries like Thailand and the Philippines are less efficient and are thus easier to predict than those of developed countries. In addition, they further conclude that the efficiency of the markets is dynamic over time implying that short-term technical indicators of the market usually have a better predictive power.

2.3.2 Feature engineering in financial prediction

The previous subsection has addressed the inefficiency of stock markets and the market predictability exist to some degree. For researchers and investors who hold the belief that stock markets are predictable, they usually devote their efforts to technical analysis and fundamental analysis.

2.3.2.1 Technical Analysis

In financial field, technical analysis is a methodology for predicting the movement direction of prices through the study and analysis of statistics gathered from trading history including price, volume, etc.

There are two underlying assumptions of technical analysis. The first assumption is that the market price at any given time reflects all available information explicitly. This assumption is derived from the assumption of EMH. The second assumption is that the price changes are not stochastic. This encourages the belief that market trends can be recognized in both short run and long run which enables the market investors to make a profit by analyzing the historic information.

The core methodology of technical analysis is to use charts which reflect the values of technical indicators like price and volume. Technical analysts believe that there exist visual patterns in the market charts. Experimental results of previous research work have indicated that such patterns exist in historic market movements [109]. Based on this, technical analysts attempt to identify movement patterns and market trends in the markets and seek to utilize those patterns.

Apart from market charts, historical prices are often used for technical analysis in the prediction of stock price movement. Closing price is one of the most frequently used technical indicators. Some other simple indicators include open price, highest price,

lowest price, etc. More complicated technical indicators are usually mathematical transformations of price, volume and other inputs.

These indicators are used in analyzing the probability of its movement direction and of continuation. Technical analysts are also interested in seeking the underlying relation between the market price and the technical indicators. Some of the examples include the relative strength index, the moving average, and MACD. Technical rules like moving average rules, filter rules and relative strength rules are also proposed to facilitate stock price prediction.

2.3.2.2 Fundamental Analysis

However, the two assumptions mentioned in the previous subsection cannot always hold. More recent work [108] indicates that the predictive power of technical analysis is quite limited and inefficient. Therefore, other researchers have tried to explore additional information which is regarded as fundamental data for fundamental analysis.

Fundamental analysis, when applied to markets like stock and FOREX, is the analysis of a business's overall financial statements (e.g. assets, profits, competitiveness, etc). Other factors like interest rates, employment, GDP which can reflect the overall economic environment would also be taken into consideration. In conclusion, fundamental analysis includes 3 levels which are economic analysis, industry analysis and company analysis.

There are two basic methods when performing fundamental analysis which are bottom-up analysis as well as top-down analysis. The top-down traders initiate their analysis with macroeconomics considering both international and national indicators. Then they gradually narrow their analysis down to industry and company analysis. The bottom-up method is performed in a reverse process. They initiate their analysis with

specific business, regardless of macroeconomics information, and proceeds in reverse of the top-down approach. Similar to technical analysis, fundamental analysis is also carried out on the available market data but with the purpose of not only predicting future market trend but also understanding its intrinsic value.

According to [42], there are 5 main sources of traditional fundamental data. They are the financial data of a company, the financial data of a market, the government and bank activities, political circumstances as well as geographical and meteorological circumstances. Fundamental data are usually more unstructured which adds to the difficulty of fundamental analysis.

Recently, unconventional fundamental data have attracted more attention from researchers and investors for predicting financial markets. Among different sources of unconventional fundamental data, financial news has demonstrated a strong impact on stock market. Heston et al. [44] discover that companies with news show difference from those without news in terms of future returns. In a more recent piece of work [110], the authors propose an approach which is able to detect events from news and learn the impact of these events on stock volatility. On the other hand, Chatrath et al. [111] use more specific macro news to predict foreign exchange market.

Other work leverages crowd wisdom by analyzing content collected from online forums or communities discussing about stock market [112, 113]. They crawl text content posted on these websites and apply sentiment analysis because public moods have an influence on investors with respect to sell/buy decisions. Furthermore, Si et al. [114] propose a Semantic Stock Network (SSN) which can conclude the discussion topics about stocks and their relations. Their results show that close neighbors in the proposed network can help improve the prediction performance.

2.3.2.3 Combination of Technical and Fundamental Analysis

There is little doubt that combining the strengths of both technical and fundamental analysis can help traders better understand the financial markets and predict the future direction so as to support their investment decisions. Therefore most market participants attempt to take the both technical and fundamental data into consideration for analysis.

Some technical analysis methods work well with fundamental analysis and can provide additional information to traders. One of the most popular methods for analyzing market sentiment is to consider trade volume. Large spikes imply that the stock has attracted a good attention from the trading community and that the shares are under either accumulation or distribution. The trade volume actually reflects the opinions of the majority of market participants and some of them may have additional insights about the particular business. Volume indicators can be very useful as they help confirm whether other traders agree with the trader's expectation on the specific business and give the trader a better understanding of it.

The combination of technical and fundamental analysis can be regarded as the combination of short term and long term analysis. Fundamental analysis usually focuses on a longer period of time but fundamental analysts would also like to obtain a favorable buying or selling price during trading. Technical analysis can be useful under these contexts. Many fundamental analysts will look at the chart of a specific business to understand its performance over a period of time when big news is released. They believe patterns tend to repeat themselves, and the traders have a tendency to respond to the news in a similar way as well.

However, the combination of the two strategies sometimes leads to negative results. For the examples mentioned above, it is significant to note that the crowd would some-

times be wrong because the traders may be irrational and emotional especially when big news is released. In addition, while the certain market movements are predictable based on patterns observed from charts, it is not guaranteed due to the efficiency of the financial market. These charts are heavily focused on the historical data and usually cannot reflect or anticipate the macro trends. The subjectivity of different traders also prevents the traders from making wise decisions when reading these charts.

2.3.3 Machine learning in financial prediction

Machine learning has already been used for time-series prediction tasks for a long time. Support Vector Machine (SVM) is used as one of the most popular financial market predictors due to its strong classification ability. Kercheval et al. [115] apply SVM in high frequency trading using 5 groups of different features to predict short term price changes. Their experimental results verify that SVM is a good tool for financial time series forecasting tasks.

Choudhry et al. [109] further combine Genetic Algorithm (GA) and SVM then propose a hybrid machine learning system. They mainly rely on technical analysis and use technical indicators as input features. In addition, they utilize the stock prices of other companies in the same industry and attempt to understand the correlations between these stocks. GA is used for feature selection in their method. Their experiments are done to predict the price of 3 stocks in the Indian stock market and show promising results.

Patel. et al [116] compare four models which are Artificial Neural Network (ANN), SVM, random forest and Naive Bayes using two methods. The first method relies on the analysis of 10 technical indicators and the second method relies on converting the technical indicators into trend deterministic data. Their experiments are carried out

in the time period from 2003 to 2012 on the two stocks. All the models improve when the technical indicators are represented in the trend deterministic data and random forest achieves the best performance.

Apart from traditional machine learning methods, neural networks also play important roles to forecast stock prices over time. Early work like [117, 118] utilizes traditional neural networks to predict the movement of stock index. Liu et al. [113] generate a sentiment dictionary of financial related keywords and build a model to calculate the sentiment score of online posts discussing different stocks. Thereafter, they apply Elman Network to deal with two sentiment indicators to facilitate stock volatility prediction.

Other researchers incorporate neural networks as part of a hybrid model. Kwon et al. [119] combine RNN and GA to predict NASDAQ and 36 stocks. Their method has a better result than traditional financial methods. Rather et al. [120] propose a novel and robust hybrid model to forecast stock movement. The proposed model includes two linear models (autoregressive moving average model, exponential smoothing model) and a non-linear model (RNN). The proposed hybrid model combines the prediction results of the 3 models and the weights are optimized by Genetic Algorithms. Their experiments show that the hybrid model can outperform the single RNN model.

More recently, deep learning is attracting more attention in the financial forecasting tasks. Ding et al. [110] implement a deep constitutional network to predict S&P index. They extract events from news and convert them into vectors using embedding methods. Then they implement a Convolutional Neural Network (CNN) to model both the long term and short term effects on stock volatility. Peng et al. [121] utilize word embeddings and deep neural networks to analyze financial news so as to predict stock price movements. The method is easy to implement but effective and can improve the prediction accuracy significantly.

2.3.4 Discussions

To summarize, the mainstream methods to predict financial market can be divided into two categories, namely traditional machine learning methods and artificial intelligence methods. Compared to traditional machine learning methods like regression, artificial intelligence, especially neural networks, has demonstrated more potential in the aforementioned prediction task.

Regarding feature engineering, the author favors the opinion that one can achieve better prediction results when combining both fundamental analysis and technical analysis. Although there is previous work to predict stock price based on traditional news, considering people are relying more on OSN as their news sources instead of traditional media, the author proposes a method to predict stock market index based on news content collected from OSN.

2.4 Evaluation metrics

2.4.1 Content Detection

Content detection tasks addressed in this thesis can be regarded as binary classification problems. To evaluate the performance of the proposed methods and to compare with other models, commonly used evaluation metrics are adopted in this thesis.

Before introducing specific matrices, the author would like to introduce the basic concept of a confusion matrix, also known as an error matrix [122] (See Table 2.1). Each row of the matrix represents the predicted classes of the instances while each column represents the ground truth of the instances (or vice versa).

The formula for some most commonly used metrics are as follows:

Table 2.1: Confusion matrix

Total population	Condition positive	Condition negative
Predicted condition positive	TP (True Positive)	FP (False Positive)
Predicted condition negative	FN (False Negative)	TN (True Negative)

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.2)$$

$$Precision = \frac{TP}{TP+FP} \quad (2.3)$$

$$Recall = \frac{TP}{TP+FN} \quad (2.4)$$

$$FPR = \frac{FP}{FP+TN} \quad (2.5)$$

$$F1 = \frac{2TP}{2TP+FP+FN} \quad (2.6)$$

2.4.2 Trend Prediction

Apart from content detection, this thesis also discusses about trend prediction. The performance of trend prediction can be evaluated both qualitatively and quantitatively. If one is only interested in predicting the trend direction, the prediction task can be regarded as a binary classification problem. Therefore, the metrics introduced in the previous subsection can be adopted to evaluate the performance of the model. On the other hand, if one is interested in predicting the real values instead of the direction, metrics used to evaluate regression models are considered.

3 metrics are used to evaluate the quantitative performance of the prediction in this thesis. Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are two scale-dependent metrics. On the other hand, Mean Absolute Percentage Error (MAPE) is a scale-independent measure. Formulas used to calculate the 3 metrics are listed below:

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N (y_t - o_t)^2} \quad (2.7)$$

$$MAE = \frac{1}{N} \sum_{t=1}^N |y_t - o_t| \quad (2.8)$$

$$MAPE = \frac{1}{N} \sum_{t=1}^N \left| \frac{y_t - o_t}{y_t} \right| \quad (2.9)$$

2.5 Summary

In this chapter, the author presents previous research work on detection and prediction using content collected from OSN. Firstly, the author introduces spam detection and phishing detection work with a special focus on low-quality content detection. The definition of low-quality content is presented followed by two mainstream detection methods, namely, user based and tweet based methods. Secondly, the author describes background information about rumors as well as relevant studies in social psychology field. Different strategies of rumor detection are further presented and significant features used for detection are discussed. Thirdly, fundamentals about market efficiency are introduced so as to demonstrate the possibility of predicting financial markets. Both feature engineering and machine learning techniques are discussed in details which are proven essential in financial market prediction. Finally, the author introduces some commonly used metrics in content detection and prediction.

Chapter 3

Real-time Low-quality Content Detection Framework from the Users' Perspective

OSN have gained huge popularity among netizens. Such popularity has attracted many cyber criminals and spammers to hunt for potential targets on OSN. A solution to filter these content polluters is necessary in order to improve user experience.

This chapter presents a real-time framework which filters the low-quality content from the users' perspective. The author proposes direct and indirect features which can fulfill the real-time requirement. These features are combined together with word level analysis to perform low-quality content detection using traditional classifiers Random Forest (RF) and Support Vector Machines (SVM).

Different from existing research work, the author carries out a survey and studies low-quality content from the users' perspective for the first time. A clear definition of low-quality content is thus provided to facilitate further detection work. Then the

author tests the features used in traditional spam or phishing detection to see whether they are applicable to other types of content polluters and characterizes the most significant ones. The work presented in this chapter is valuable in filling the gaps of the current detection methods for low-quality content on OSN and in improving the user experience.

The rest of this chapter is organized as follows. Section 3.1 introduces the overview of the proposed low-quality content detection framework. Section 3.2 presents the results of the survey conducted by the author and defines the low-quality content in a clearer way based on the survey results. Thereafter, Section 3.3 provides a detailed study of features used for real-time low-quality content detection evaluating their significance in terms of time and accuracy. This is followed by a description in Section 3.4 of how the author processes and extracts the various features from the original tweets. Section 3.5 illustrates the detection results using the selected features and discusses the comparisons with other research work. The last section concludes the chapter.

3.1 General Overview

3.1.1 Terminology

The research work introduced in this chapter is focused on the content polluter detection on Twitter. As such, the author would like to define some terms used in Twitter before introducing the framework.

- **Twitter** Twitter is an Online Social Website with hundreds of millions users. Users can share their opinions and comments on other users' posts on Twitter.
- **Tweet** Short messages posted by users are called tweets which are limited to the

length of 140 characters.

- **Retweet** A retweet is the forward or repost of a tweet posted by users on Twitter. There is an indicator “RT” at the beginning of retweets.
- **Favorite** When users like or agree with a tweet, they can click the “favorite” button below the tweet to express their opinions/feelings. The number of favorites of that tweet is next to the “favorite” button and can be seen by everyone.
- **Verified user** Famous individuals or organizations can request verification from Twitter to increase the influence of their accounts. Such users become verified users if Twitter has confirmed the identities/entities behind the accounts. The verified status of a user can be seen by everyone.
- **Follower & Followee** If user A would like to receive user B ’s tweets and updates, user A can “follow” user B . In this case, user A is the follower of user B and user B is the followee of user A . In some previous work, a followee is also called a friend. The number of followers and followees of a user can be seen by everyone.

3.1.2 Overview of the Framework

Fig 3.1 shows the overview of the proposed low-quality content detection system. The framework comprises two portions, the actual real-time detection of low-quality content tweets (refer to the shaded blocks in Fig 3.1) and the out-of-band training process (refer to the unshaded boxes of Fig 3.1).

The training process is conducted out-of-band to train the classifier used for real time low-quality content detection. A user survey is conducted to provide insights on the definition of low-quality content from the users’ perspective. These are then used as label guides to manually label 100,000 tweets crawled via Twitter API. Significant

Figure 3.1: Overview of the low-quality content Detection Framework

features (both direct and indirect) of low-quality content are identified from the 100,000 labeled tweets and these features are combined with word level analysis to train the classifier.

After training the classifier, the classifier is ready to predict the labels of tweets submitted to the system. These tweets experience the same feature extraction phase as the training phase and are then forwarded to the trained classifier for low-quality content detection. The classifier will then predict whether the tweet is low-quality content or not. What is worth mentioning here is that both the feature extraction and low-quality content detection can be done in real time.

3.2 A study on low-quality content from users' perspective

3.2.1 Cluster analysis of low-quality content

It is meaningful to understand users' attitudes and definitions of low-quality content before proceeding with the subsequent research. In order to design a survey which can fully convey users' opinions about low-quality content, the author and two students

manually investigate and verify low-quality content via cluster analysis.

The Streaming API provided by Twitter ¹ gives developers low latency access to Twitter's real-time global stream of tweet data from public timelines. The author uses Streaming API to crawl 10,000 tweets as a preliminary dataset. These tweets were collected in March 2016 randomly. Then three annotators are asked to label the tweets as either low-quality content or normal tweets in the dataset. During this phase, only some general descriptions of low-quality content are provided instead of clear labeling guidelines. If any of the three annotators marks the tweet as low-quality content, it will be regarded as potential low-quality content.

There may be bias due to the limited size of the preliminary dataset and the three annotators' opinions may not represent each and every user. However, labeling during this phase does not need to be that accurate and the purpose is to get a general idea of the low-quality content from the users' perspective so as to design the questions in the survey such that they are more typical and representative.

To perform the cluster analysis, the author represents the tweets with a set of features (described in details in the later section) and then applies the Expectation Maximization (EM) algorithm [123] to group together tweets which have similar characteristics or behaviors. What inspires the author to use EM algorithm to roughly classify the low-quality content is [57] which uses EM to group spammers into several categories. However, the difference between this work and theirs is that the author uses EM to classify the tweets (i.e. low-quality content) instead of accounts (i.e. spammers).

After removing groups with too few tweets and emerging groups with similar tweets, all these low-quality content can be divided into four categories:

1. **Low-quality advertisements.** These advertisements include not only decep-

¹Twitter API: <https://developer.twitter.com/en/docs>

tive or false advertisements but also those valueless advertisements posted by obscure users. Two relative examples are “*take free bit-coin every three minute (URL omitted)*” and “*Hot, my little pony friendship city light curtain. (hm118) - Full read by eBay (URL omitted)*”. Some pornographic and violent content also appear in the form of advertisements which mars users' experience when browsing normal tweets.

2. **Automatically generated content.** This type of content is usually posted by some applications or online services instead of the users themselves, mostly for promotion purposes. Once the user has given authorization to these applications and services, some user behaviors would trigger automatically generated content like “*I've collected 7,715 gold coins! (URL omitted) #android, #androidgames, #gameinsight*” or “*Today stats: 4 followers, No unfollowers via (URL omitted)*”. Content produced by the same application tends to be similar or has limited variations. A large amount of repetitive content significantly erodes the user experience.
3. **Potential meaningless content.** Some of the potential meaningless content is also posted by bots and has different forms. Some of them are readable like quotes of famous people or the created time of the tweet. Some of them are unreadable like mere messy codes (e.g. *g7302t\$u!7#52jgi4o*).
4. **Click baits.** The characteristics of low-quality content falling into this category is not very obvious and they cover a wide range of topics. Many of them look like normal messages but in most cases, the link appearing in the tweet is not related to the tweet text. Furthermore, some of the links lead to malicious sites.

3.2.2 Design of the survey

The author designs a survey according to the cluster analysis and posts it online where participants have to answer two questions related to personal information, namely, age and gender and eight questions related to OSN and low-quality content. A full version of survey is shown in Appendix A. The purpose of the survey is to understand:

- The impact of low-quality content on user experience when using OSN.
- What kind of content is regarded as low-quality content by users.
- To what degree can users tolerate low-quality content before considering unfollowing.

The survey is posted online and is entirely anonymous. At the beginning of the survey, the participants are told that the survey is anonymous and their responses will be used for research purpose. In other words, consent is implicit as in by taking part in this survey, it means the participant has given their consent. It is open to anyone online and participants voluntarily participate in the survey. Hence no participant is harmed physically and mentally.

After posting the survey online, the survey has received 211 responses. All of them are valid as all questions in the survey are compulsory and the participants have to complete all the questions in the survey before they can submit it.

As the survey link is posted on several famous online social websites (e.g. Twitter, Sina Weibo, etc), it ensures the survey results are indeed from OSN users. 88.7% of the respondents use OSN every day and 9.48% of them use OSN at least once a week. These participants are from different age groups with 74.88% in the 18 to 25 age group and 44.55% of them are females.

3.2.3 Results of the survey

For this analysis, the survey focuses on several aspects of low-quality content from the users' perspective. First, the author would like to demonstrate the significance of this work. Technically, the filtering mechanism can be applied on both the Twitter server side and the user client side. The fact is that Twitter only bans the accounts pertaining to abusive behaviors [54]. For obvious commercial reasons, Twitter will not ban advertisements unilaterally from its side, especially those repetitive low-quality advertisements. For other low-quality content, as long as it does not violate Twitter rules, there is no reason for Twitter to filter them as well. However, too much valueless content hampers users from browsing other meaningful content and 97.16% of participants believe such content affects their user experience more or less as shown in Table 3.1. Considering this, a filter applied on the client side would be very meaningful to the Twitter users.

Table 3.1: How much do content polluters affect your user experience when using social network sites?

Options	Number	Ratio
Very much.	48	22.75%
A bit but still bearable.	141	66.82%
A little.	16	7.58%
They don't affect my user experience.	6	2.84%

To better understand the impact on users, the author also asks participants about their responses to such low-quality content. One conjecture is that if one account posts too much low-quality content, its followers tend to unfollow him or her. Surprisingly, contradicting this conjecture, 72.51% of the respondents seldom or never clean up their followees (Table 3.3) although most of them admit that low-quality content does affect their user experience more or less as shown in Table 3.1.

As shown in Table 3.2, 19.91% of the participants admit that, out of courtesy, they

will follow reciprocally if someone follows them. The result corresponds to that in [124] indicating that users tend to reciprocate out of social etiquette.

Table 3.2: If someone follows you, will you follow back?

Options	Number	Ratio
follow back out of courtsey	42	19.91%
follow those I know or share common interests	149	70.62%
don't follow back	15	7.11%
others	5	2.37%

The author then carries out a cross analysis for users' habits about following (Table 3.2) and cleaning up friends (Table 3.3). The results are shown in Fig 3.2. Among the 19.91% of people who easily follow back, 85.71% of them actually do not clean up their friends regularly.

Table 3.3: How often will you clean up your followees/friends?

Options	Number	Ratio
Seldom or never.	153	72.51%
More than once a month.	41	19.43%
At least once a month.	11	5.21%
Almost every week.	6	2.84%

This finding is consistent with [125] that users tend to interact with only a small subset of their friends and pay little attention to as much as half of their friends. This further indicates that once a malicious user tricks a normal user into following him in some way, it is highly possible that this user may continuously deliver low-quality content and affect his followers' user experience because most users feel too bothersome to clean up their followees. Therefore, some users may have a close friendship with malicious users. [126] makes similar conclusions in their research. Thus, it is necessary to set up an automatic filtering mechanism for low-quality content to improve the user experience of using OSN.

To investigate the users' definitions of low-quality content, the author provides

Figure 3.2: Users' habits about following and cleaning up friends

abstract (general) categories in one question (see Fig 3.3) and specific example tweets in another question (see Fig 3.4). The two questions are designed based on the cluster analysis results in the previous phase. Participants are asked to tick what they regard as low-quality content.

Figure 3.3: Users' definition for low-quality content (Abstract categories)

It is generally agreed that deceptive content and messy codes belong to low-quality content but what is interesting here is that the category which receives the third most complaints is automatically generated content by some applications and services. Ac-

Figure 3.4: Users' definition for low-quality content (Specific examples)

cordingly, two automatically generated tweets receive the most complaints by users as shown in Fig 3.4.

In Fig 3.4, it is observed that users feel more upset with low-quality advertisements posted by obscure individuals or organizations rather than high quality advertisements posted by well-known individuals or organizations. Nevertheless, even high quality advertisements receive a large percentage of complaints (61.61%) from users which is consistent with the results shown in Fig 3.3 that 53.55% of users think all advertisements are low-quality content. Again, this underscores the fact that detecting only malicious content is not enough to improve overall user experience.

The survey questions include two contrast tweets as well. Contrast tweets are those regarded as normal by the annotators in the preliminary studies. These contrast tweets are presented so that the survey respondents are being reminded of what normal tweets are and the author expects less people will regard these contrast tweets as low-quality content compared to the real low-quality content. As seen in Fig 3.4, the two contrast tweets still receive more than 40% of the responses of being recognized as

low-quality content and this may be due to the fact that 40.76% of the participants believe uninterested content should also be regarded as low-quality content.

Another factor observed that may affect users' attitudes towards suspicious content is actually the proportion of low-quality content among all the tweets posted by a suspicious user. From Table 3.4, 19.43% of those surveyed will start considering unfollowing if low-quality content makes up more than 25% of all tweets while 35.55% of them will start considering unfollowing if more than 50% is low-quality content.

Table 3.4: What's the maximum threshold (as a percentage of your recently received messages) you can bear before considering unfollowing him/her?

Options	Number	Ratio
Too bothersome to unfollow	28	13.27%
Nearly 100%	15	7.11%
More than 75%	52	24.64%
More than 50%	75	35.55%
More than 25%	41	19.43%

Hence, from previous work [127] and based on the results of the survey, the author provides a clearer definition of low-quality content. The author defines it as a large amount of valueless or fraudulent content which hampers users from browsing useful content. They include low-quality advertisements, automatically generated content, potential meaningless content, click baits, etc. The "low-quality content" being referred to in this paper describes tweets instead of users which differentiates the work in this thesis from existing research work which filter users [57, 128, 129] rather than the "offending" low-quality tweets. This is because some users post low-quality content not out of ill intention. These users still post content which their followers are interested in so it is not acceptable to simply filter all the users straightaway once they have ever posted low-quality content.

Moreover, according to the results shown in Table 3.4, a complement to the definition is that identical content posted by different users will not always get the same

labeling results on whether it is low-quality content or not. This is because whether such content accounts for a large proportion of all tweets of that user is also an important factor. The frequency of such content determines the negative impact on overall user experience. If such content constitutes a huge proportion of a user's tweets, it is regarded as low-quality content. If it is sporadic relative to a user's total number of tweets, their followers will not mind them so much. Thus the definition of low-quality content is not necessarily an absolute unit of content but can be relative and vary from user to user.

3.3 Identifying features characterizing low-quality content

Low-quality content detection is usually viewed as a classification task. A lot of features have been proposed for spam or phishing detection. The question about whether these features can be adopted for detecting the low-quality content defined in this thesis will be addressed in the later section. In this section, the author provides an in-depth analysis of features proposed and the common features presented in existing studies. The author then determines the dominant features from the perspective of both time and accuracy for low-quality content detection.

3.3.1 Direct features

The typical structure of a tweet crawled is in JSON format. All the information included in this raw JSON tweet can be directly extracted almost at the same time it is posted. These features are the most efficient ones in low-quality content detection from the perspective of time performance. Since they can be extracted directly, they

are called direct features (DF) in this thesis. Direct features which can be extracted from the raw JSON tweet are listed in Table 3.5. Features 1 to 10 are Tweet based while the rest are profile based (i.e. account based).

Table 3.5: Direct features

Index	Feature	Comments
1	Source	Tweeting tools
2	Type	Regular, Replies, Mentions and Retweets.
3	Retweet_count	The number of times the tweet is retweeted.
4	Favorite_count	The number of times the tweet is favorited
5	Hashtags_count	The number of hashtags in the tweet.
6	Urls_count	The number of urls in the tweet.
7	Mentions_count	The number of mentions in the tweet.
8	Media_count	The number of media in the tweet.
9	Symbols_count	The number of cashtag in the tweet.
10	Possibly_sensitive	If the tweet possibly contains sensitive content.
11	Location	If the location field of profile is not filled.
12	URL	If the URL field of profile is not filled.
13	Description_len	The length of the description field of.
14	Verified	If the user is verified by Twitter.
15	Ff_ratio	Followers_count / Friends_count
16	Followers_count	The number of followers of the user.
17	Friends_count	The number of friends of the user.
18	Statuses_count	The number of statuses the user post.
19	Favourites_count	The number of tweets the user favorite.
20	Listed_count	The number of lists the user create.
21	Account_age	The lifespan of the account.
22	Default_profile	If the user is using a default profile.
23	Default_profile_image	If the user is using a default avatar.

Since a user can post multiple tweets, the profile based features for different tweets posted by the same user are identical while tweet based features may be different from tweet to tweet but can be the same for tweets posted by different users because of retweets.

3.3.2 Indirect features

However, direct features alone cannot always give the best performance. According to the users' responses presented in the previous sections, the proportion of low-quality content also affects users' attitudes towards low-quality content. Thus indirect features (IF) are also identified. Indirect features are those which cannot be directly extracted from the raw JSON tweet. Instead, a separate request is sent to Twitter to obtain the additional information. Indirect features capture the history information and tweeting behaviors of a user which are proven significant for low-quality content detection in the later section. The purpose for adopting both direct and indirect features is to achieve a balance between detection accuracy and time performance.

The indirect features are listed in Table 3.6. As the indirect features are historical data of a particular user, most of them are profile based except for the last one. The author is the first to use media, symbols and lists related features for similar detection tasks.

Table 3.6: Indirect features

Index	Feature	Comments
1	Source_count	No. of sources used for posting n latest tweets.
2	Type_count	No. of types of the latest n tweets posted.
3	Hashtags_proportion	% of tweets with hashtags in the latest n tweets.
4	Urls_proportion	% of tweets with urls in the latest n tweets.
5	mentions_proportion	% tweets with mentions in the latest n tweets.
6	Media_proportion	% tweets with media in the latest n tweets.
7	Symbols_proportion	% tweets with symbols in the latest n tweets.
8	Sensitive_proportion	% tweets possibly sensitive
9	Nonfriends_interaction	If the tweet is an interaction between non-friends.

3.3.3 Word level analysis

However, both direct and indirect features do not take the semantic meaning of the original tweet text into consideration. Thus word level analysis is designed to capture the content characteristics of the tweet text. Like spam emails, some keywords such as “click”, “free” are more frequently seen in low-quality content than in normal tweets.

Actually, word level analysis is frequently used in spam detection for emails [130] [131] [132] while not that popular in spam detection on OSN. Possible reasons may be the extensive use of informal abbreviations and the limited length of a tweet. [55] uses a wordpress comment blacklist but this blacklist may not be suitable for low-quality content detection on Twitter. Thus, in this study, the author analyzes those tweets labeled as low-quality content and attempts to find the terms which happen most frequently and builds a blacklist keyword dictionary. The author exploits the bag-of-word model to process the original 10,000 tweet texts. There is one word bag for low-quality content and another for normal tweets. Stop words are removed in each bag. For terms in the word bag of low-quality content, the term frequency is used to represent the weight of the term but the weight will be reduced if the same word also appears in the bag of normal tweets. Then the words in the bag of low-quality content are sorted according to their weights and the top N words make up the blacklist keyword dictionary. The selection of N will be discussed in the later section.

Each term in this dictionary can be viewed as one feature and these features together with the direct and indirect features proposed earlier are combined for detecting low-quality content. In Section 3.5, the efficacy of all direct and indirect features as well as the word level analysis proposed in this section is validated.

What is worth mentioning here is that for a real time environment, the dictionary will evolve in order to capture the “hot words” in low-quality content. The update

of the dictionary can be implemented in real time while the reconstruction of the corresponding training set should be done at regular intervals. It is noted that it is not possible for such a dictionary to include all blacklist keywords and that is why it should be updated regularly. In addition, the purpose of such a dictionary is to help improve the detection performance of low-quality content instead of listing as many blacklist words as possible. Further details about the performance improvement are discussed in the later section.

3.4 Pre-implementation tweet processing

3.4.1 Data collection and preprocessing

To collect tweet data, the author uses one thread to crawl tweets through public streams provided by Streaming API. The tweet crawled in this way is in the JSON format. Another thread is run at the same time to parse the raw tweet and then extract the direct features shown in Table 3.5.

Twitter REST APIs provide access to read and write Twitter data such as posting a new tweet, reading the author's profile and follower data, etc. In this case, the author uses a third thread to send a request to function *statuses/user_timeline* so as to obtain the latest tweets of a particular user and calculate the corresponding indirect features listed in Table 3.6. The three threads can work simultaneously in order to save time for detection.

For the preparation of word level analysis, the author exploits the *Text Mining (tm)* Package developed for R [133]. For tweets marked as low-quality content, the author uses regular expressions to remove all RT, @, # tags as well as all URLs in tweets. Then only English characters are preserved and transformed into lower case. These

tweets are then forwarded to the *tm* library to remove all stop words. One consideration here is whether stemming should be done to process the tweets after removing the stop words as the stemming step could help reduce the number of possible terms but with the risk of losing part of the word meanings. The details would be discussed in Section 3.5.

The Twitter dataset created by the author consists of 100,000 tweets generated by 92,720 distinct users. These tweets are collected from 16th May to 17th May 2016. The days are randomly selected with no particular reasons.

3.4.2 Labeling tweets

To develop an automatic low-quality content detection system, it is necessary to build a training set. The author has set up some label guides based on the survey results to ensure the label from annotators can fully convey the users' opinions. If the tweet falls into the four categories discussed in the preliminary studies, the timeline of the user will also be examined. If similar low-quality content appears frequently (usually more than 50% of the latest tweets posted) in the timeline of the user, the tweet will be labeled as low-quality content, otherwise it is regarded as a normal tweet. What should be noted here is that the annotators do not label other tweets appearing in the timeline of the user; they are just used as a reference during the labeling process. In other words, they are not considered as labeled tweets.

Cohen's Kappa coefficient (k) is computed to evaluate the inter-rater agreement of the labeling which is also used in [25] and [93] for similar purpose. The formula to calculate Cohen's Kappa coefficient is as follows:

$$k = \frac{p_o - p_e}{1 - p_e} \quad (3.1)$$

$$p_o = \frac{n_{agree}}{N} \quad (3.2)$$

$$p_e = \frac{n_{1,p} * n_{2,p} + n_{1,n} * n_{2,n}}{N^2} \quad (3.3)$$

where p_o is the relative observed agreement among annotators, and p_e is the hypothetical probability of chance agreement, using the observed data to calculate the probabilities of each observer randomly seeing each category. In the formula above, $n_{1,p}$ denotes the number of tweets that the first annotator marks as low-quality content while $n_{1,n}$ is that of normal tweets. So are $n_{2,p}$ and $n_{2,n}$ for the second annotators. The annotation results reach a high agreement of $k = 0.90$

In total, the annotators have labeled the 100,000 tweets crawled and 9,945 of them are labeled as low-quality content.

3.4.3 Training and testing classifiers

The focus of the evaluation is to show the feasibility of derived features in the real-time detection of low-quality content. Hence the classification method used is not the focus. According to [77] and [134], Random Forest (RF) and Support Vector Machine (SVM) outperform other classifiers in detecting spam and phishing. Thus the author chooses the two classifiers to perform the low-quality content detection task. The classifier is trained using the training set with a 5-fold validation. Then the model is tested on the test set and the prediction results are checked against the labels.

A series of experiments are conducted to evaluate the performance of the proposed low-quality content detection system. 100,000 labeled tweets are used to test the

system to gather the prediction results and to evaluate the computation time. All the experiments are run on a Dell Precision T3600 PC with Intel Xeon E5-1650 processor at 3.20 GHz with 16 GB of RAM.

3.5 Implementation results and evaluation

3.5.1 Word level analysis

To achieve a better performance through word level analysis, two special factors are discussed in this subsection. One is the size of the keyword blacklist dictionary. Usually a larger dictionary will increase the detection accuracy but may fall into the overfitting trap. For each word preserved in the low-quality content corpus, its weight determines whether it can be added into the dictionary. Its weight is represented by its term frequency in low-quality content minus its term frequency in normal tweets. The dictionary size can be varied by setting different thresholds for weight.

The other controlled factor is whether to perform stemming on the tweet texts during the preprocessing phase. In this subsection, the author performs low-quality content detection with different dictionary size and evaluates the performance from the perspective of both time and detection rate. The F1 measure results are shown in Fig 3.5.

It is observed from Fig 3.5 that when the size of the keyword blacklist dictionary becomes larger, the detection performance increases moderately. However, when the dictionary size is further increased, both of them fall into the trap of over-fitting. No stemming performs better than stemming when the dictionary size is not very large but experiences an early and severe drop in detection performance when the dictionary size increases. Another advantage of no stemming is that it can save the time cost which

Figure 3.5: F1 measure with/without stemming for different dictionary size

will otherwise be incurred for the extra stemming step. According to the observations, the dictionary size is set to 150 and the stemming step is skipped in the following experiments. An example of keyword blacklist dictionary can be seen in Appendix B.

3.5.2 Feature rank

The construction of the keyword blacklist dictionary has already included the selection of important word features. In this subsection, the author discusses more about the significance of other direct and indirect features. Initially, the Recursive Feature Elimination (RFE) is applied to test the performance of using different number of features described before and the results are shown in Fig 3.6. It is observed that the accuracy reaches a peak when using 30 features out of a total of 32. This indicates that most of the proposed features are quite efficient in detecting low-quality content. What is worth mentioning here is that even when adopting only 10 features, the accuracy can reach more than 90%. The top 10 features selected by RFE can be seen in the last column of Table 3.7.

Figure 3.6: Accuracy of different subsets of features

Table 3.7: Feature rank

IG	CHI	AUC	RFE
mention_prop	mention_prop	favourites_count	follwers_count
url_prop	url_prop	type_cnt	friends_count
media_prop	media_prop	urls_cnt	statuses_count
type_cnt	favourites_count	url_prop	url_prop
favourites_count	type_cnt	mention_prop	listed_count
friends_count	friends_count	mentions_count	urls_count
urls_count	follwers_count	type	mention_prop
hashtag_prop	urls_count	default_profile	media_prop
follwers_count	hashtag_prop	ff_ratio	favourites_count
Type	type	hashtags_count	hashtag_prop

The author also uses three other popular feature evaluation methods: Information Gain (IG), Chi-square test and Area under the ROC Curve (AUC) to compute the rank of the features and the top 10 features selected via different evaluation methods are shown in Table 3.7. It is to be noted that the focus is to extract the top ranking features. Hence, the relative quantitative performance is not shown.

The results show that most of the indirect features are more efficient in detecting low-quality content than direct features. This is because the indirect features also take the posting history of a user into consideration. Among all the features, mention_prop,

url_prop and favourites_count are selected by all four feature evaluation methods.

3.5.3 Detection performance

In this subsection, the author presents the performance of the proposed method in detecting low-quality content by adopting different subsets of features. According to the observed results in the last subsection, the dictionary size is set to 150. In the experiments, three subsets of features are adopted. Feature Subset I includes all direct features. Feature Subset II includes all direct and indirect features. Feature Subset III includes all direct and indirect features plus word level analysis. The author performs both RF and SVM for the low-quality content detection task and the detection performance are shown in Table 3.8. Lets test

Table 3.8: Detection performance of different feature subsets

Random Forest				
Feature Subset	Acc(95% CI)	Fpr	F1	Time(s)
Direct	0.9526±0.0013	0.0103	0.7124	0.0002
Direct+Indirect	0.9599±0.0012	0.0089	0.7634	1.9327
Direct+Indirect+Word	0.9711±0.0010	0.0075	0.8379	1.9342
SVM				
Feature Subset	Acc(95% CI)	Fpr	F1	Time(s)
Direct	0.9335±0.0015	0.003	0.4981	0.0003
Direct+Indirect	0.9418±0.0015	0.0074	0.6089	1.9328
Direct+Indirect+Word	0.9562±0.0013	0.0037	0.7199	1.9343

The performance of the proposed framework is evaluated by Accuracy, FPR and F1 which are defined by the formulae presented in Section 2.4. In this case, TP means the amount of low-quality content which has been correctly labeled by the framework as low-quality content and TN means the amount of normal content which has been correctly labeled by the framework as normal content. Similarly, FP means the amount of normal content which has been wrongly labeled as low-quality content and FN means the amount of low-quality content which has been wrongly labeled as normal content.

It can be concluded that RF always performs better than SVM. Direct features alone can help detect roughly 95.26% of the low-quality content and the time performance is more than satisfactory - almost as soon as the tweet is posted. When both direct and indirect features are adopted, the accuracy increases moderately to 95.99%. The detection accuracy goes up to 97.11% when considering word level analysis and the F1 measure also increases significantly to 0.8379. For all 3 subsets of features, the false positive rate remains low at about 0.01.

For time performance, unlike [134], the author does not include the time for building the training model because the training phase can be done out of band. In other words, the time performance is the detection time of content polluters and it includes the time required for extracting features as well as that for prediction.

The processing time for feature subsets II & III is longer than that of subset I. This is because the indirect features incur response time for the additional request to Twitter REST API. This fact notwithstanding, the current time performance for subsets II & III is still acceptable for real-time detection requirement (less than 2s). To summarize, the results show that the proposed features can not only achieve a good detection rate but also are time efficient.

3.5.4 Comparisons with other methods

3.5.4.1 Blacklists and Twitter policy

Blacklists are often used for detecting phishing or spam. The biggest problem with blacklists is that there is always a time lag between the occurrence of the malicious content and the report to the blacklists. This problem makes blacklists less efficient to fulfill the real-time detection requirements. Moreover, most of the blacklists focus on phishing or malware and do not pay much attention to low-quality content. The

focus of Twitter suspension policy is a bit different from the mentioned blacklists but still falls into the same trap. The author further checks the status of the low-quality content collected during the experiments and **60%** of them is still there. One possible reason is that Twitter mainly focuses on content which breaks Twitter rules and pays less attention to other low-quality content. Even if they can detect such content, they would not filter them because of commercial reasons. Owing to the lack of an effective real-time low-quality content detection method, the users' timelines are filled with low-quality content which hampers them from browsing other meaningful content. The framework proposed in this thesis tackles the problem in a holistic manner since the low-quality content detected here covers valueless content of different types from the users' perspective and includes spam and phishing which are commonly covered by existing works. Hence, the proposed framework has been proven to be of great value to improve the overall user experience.

3.5.4.2 Other spam/phishing detection methods

The reason it is unnecessary to distinguish among spam, phishing and other low-quality content is because they share similar characteristics. Furthermore, from the perspective of users, they do not care what categories these low-quality content belong to. To improve overall user experience, the aim of the proposed framework is to filter all the low-quality content regardless of their categories. However, other research work either focuses on spam detection or phishing detection, so it is not that meaningful to compare the proposed framework with them.

Nevertheless, to provide some insight into the performance of the proposed method for detecting low-quality content, the author still selects two related research work for comparison. One is [57] and the other is [55]. The author implements their methods and performs low-quality content detection on the same dataset as described in Section

3.4 and the results are shown in Table 3.9.

Table 3.9: Comparisons of different methods

Method	Acc(95% CI)	FPR	F1
Ours	0.9711 $\pm$ 0.0010	0.0075	0.8379
Wang's	0.9580 $\pm$ 0.0012	0.0056	0.7538
Lee's	0.8514 $\pm$ 0.0022	0.0919	0.7025

For Lee's method, a possible reason which may explain the low detection rate is that the detection method is designed based on accounts instead of tweets. The high false positive rate indicates that some users who are classified as spammers also post normal content which their followers may be interested in. This proves that the detection for low-quality content is better to be carried out on a tweet level instead of an account level. In addition, some of the features used in Lee's method cannot fulfill the real-time requirement.

For Wang's method, their false positive rate is slightly lower than the proposed method while the accuracy and F1 measure are much worse. This is because the proposed method is specially designed for low-quality content detection while their detection is mainly focused on spam.

The comparison results prove that the proposed framework achieves a good performance in both time and detection rate for low-quality content detection.

3.6 Conclusions

In this work, the author proposes a solution to address the problem of detecting low-quality content on Twitter in real time. The author first derives a definition for low-quality content as a large amount of repeated phishing, spam and low-quality advertisements which hamper users from browsing normal content and erode the user

experience. This definition is based on the outcomes of a survey targeting real users of OSN and is thus proposed based on the users' perspective. It is very necessary to detect the low-quality content in real time so as to improve user experience on OSN.

The author has performed a detailed study of 100,000 tweets and identified a number of novel features which characterize low-quality content. An in-depth analysis of these features is provided and the efficiency of using word level analysis for low-quality content detection is validated. The direct and indirect features can actually distinguish most of these low-quality content and the accuracy is about 95%. In addition, when word level analysis is adopted, the accuracy soars to 97.11% while still maintaining a low false positive rate (0.0075) and a good F1 measure (0.8379). The time needed to process all the features proves feasible for real-time requirement.

Through a series of experiments, the author demonstrates that the proposed framework can achieve a good performance for real-time low-quality content detection for OSN. The framework addresses the low-quality content problem holistically. The framework is therefore of great value to the users not just in removing spam and phishing but also serves to detect other types of low-quality content and finally improve the overall user experience in real time.

Chapter 4

Unsupervised Rumor Detection Model based on User Behaviors

Apart from its social functions, OSN is gradually being used as a platform for acquiring all kinds of information and news. This is a natural transformation since OSN provides a convenient and efficient way for its users to share their thoughts and opinions in real time. With a large amount of content created everyday on OSN, there is an absence of a supervision mechanism to guarantee the veracity of the unprecedented source of misinformation.

A widely known example is the Fukushima Daiichi nuclear disaster occurred in Japan in March 2011. A rumor saying that iodized salt can protect people from the radiation effects spread fast on China's OSN - Sina Weibo. People rushed to buy much more iodized salt than they need which increased the salt price by almost 5 to 10 times. Similar examples of the abuse of OSN to spread rumors can be found in [135, 136].

It is observed from these famous rumor cases that the updates with breaking news events are released piecemeal under most circumstances which causes a large percentage

of posts being unverified when they are posted, and even confirmed as false in worse cases [137]. Considering the potential misunderstanding, panic and even hatred caused by rumors carrying misinformation and the infeasibility for human involvement with extra efforts to establish the source and veracity of every post on OSN, an automatic mechanism for detecting rumors is of high practical value.

This chapter presents a hybrid rumor detection model which combines Recurrent Neural Networks (RNN) and AutoEncoder (AE). The assumption behind the model is that behaviors of users when posting rumors and non-rumors tend to be different and rumors only account for a small proportion in all posts. Thus rumors can be regarded as anomalies.

The rest of this chapter is organized as follows. Section 4.1 introduces the overview of the proposed learning model for rumor detection. Section 4.2 illustrates the features adopted for the rumor detection and provides the details of how the RNN and the variant AE are combined to build the detection model. Section 4.3 presents the experimental results as well as the comparisons with other methods. Section 4.4 concludes the whole chapter.

4.1 General Overview

4.1.1 Definition of Rumors

Different research work defines rumors in different ways which leads to some confusion between rumors and false rumors. In this paper, the author favors the definition presented in social psychology field, that is, a rumor is a controversial and fact-checkable statement [138]. It should be noted that a rumor does not necessarily indicate misinformation.

After a while, when external sources are being referred to debunk or verify a rumor, it can turn into an actual rumor (i.e. misinformation) or a genuine fact. Comparing with true information spread as rumors due to lack of first-hand knowledge, those actual rumors are more harmful without doubts. Therefore in this work, the focus is the preemption of actual rumors excluding those later verified as true information. In the later sections, the term “rumor” shall be used to refer to actual rumors for convenience. On the other hand, for those rumors which turn out to be true, they are referred to as genuine facts or non-rumors.

4.1.2 Descriptions of the Problem

In this work, the author uses Sina Weibo as an example to illustrate the research work for false rumor detection. Sina Weibo, sometimes simply called Weibo, is one of the biggest social media platforms in China. Weibo literally means microblog [139]. Sina Weibo can be regarded as an equivalence to Twitter in China. To avoid confusion, thereafter, Sina Weibo shall be used to refer to the web service while Weibo shall be used to refer to the post (i.e. microblog).

Some of the notations used are listed as follows:

- w is used to denote a Weibo (microblog) posted by a user which is equivalent to a tweet in Twitter. A Weibo can be represented by a feature vector $V = \{f_1, f_2, \dots, f_n\}$.
- u is used to denote a user which is equivalent to the user in Twitter. The author of a Weibo w is denoted as u_w
- $S_{w,k} = \{w_1, w_2, \dots, w_k\}$ is called a recent Weibo set regarding Weibo w with the parameter k . Assuming the Weibo w is posted at time t , then $S_{w,k}$ represents

the recent k Weibos of user u_w before t . Note that $w \in S_{w,k}$ (i.e. $w = w_1$).

Based on the above concepts and definitions, the rumor detection task is articulated as follows. For a suspected Weibo w submitted to the proposed system, its recent Weibo set $S_{w,k}$ is extracted and the corresponding features are computed. The anomaly detection is performed on $S_{w,k}$ using a variant of AutoEncoder (AE). The Weibos in $S_{w,k}$ are ranked according to the degree of deviation represented by the errors of AE. Those Weibos whose errors are higher than its self adaptive threshold are regarded as rumors.

4.1.3 Overview of the Model

To illustrate the procedures and performance of the proposed model, experiments are carried out on Sina Weibo. Since Sina Weibo implements many common features of OSN, the model can be applied to other OSN with little refinement needed. Fig. 4.1 shows the overview of the proposed learning model for rumor detection comprising two phases, namely, feature extraction and learning model.

For a suspicious Weibo submitted to the system, the system refers to the profile of its author and crawl a set of recent Weibos posted by the author. In the feature extraction phase, each Weibo in the recent Weibo set is represented by the features extracted from itself and its corresponding comments. In the learning phase, comment based features (i.e. time dependent features) of the recent Weibo set will be first input into the RNN module to be analyzed over time as comments arrive at different time. The output of the RNN module is then combined with those Weibo based features (i.e. time independent features) and forwarded to the variant AE as an input. The variant AE then learns the normal behavior patterns of the author. Finally, the errors between the input and output of AE are calculated to perform anomaly detection and a Weibo

Figure 4.1: Overview of the proposed rumor detection model.

with a larger error usually indicates a high possibility of an anomaly (i.e. rumor).

4.2 Data Collection Methods and Detection Models

In this section, the author first introduces the data collection methods and the features computed from both the original Weibos and their comments. The author then presents the implementation details of RNN and the variant AE in the learning model and how they are combined for rumor detection purpose.

4.2.1 Data Collection

In the experiments, the dataset contains n different original Weibos $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n\}$ including both rumors and non-rumors. For each original Weibo $\mathbf{w}_i$ submitted to the system, it crawls its recent Weibo set $\mathbf{S}_{w_i,k} = \{\mathbf{w}_{i1}, \mathbf{w}_{i2}, \dots, \mathbf{w}_{ik}\}$.

k is the size of a recent Weibo set $\mathbf{S}_{w_i,k}$. There are already some discussions about the selection of k in [24] and [106]. k cannot be too large so as to avoid the gradual change in a user's posting behaviors over time. On the other hand, k cannot be too small either, otherwise, there are too few data samples for the learning model to learn the user behaviors. k is set to 50 based on the experimental results in the aforementioned work.

A big challenge of this work is to build a validated data set, especially the access to confirmed rumors as they only constitute a tiny portion of the total social media posts [97]. Intensive manual labeling work not only costs a lot of human labor and time but also cannot guarantee the accuracy of the labeling results.

Fortunately, Sina Weibo has an official "Weibo Community Management Centre"¹ to deal with the complaints from users. If a user finds that a post is suspected to be a rumor, he or she can click the "report" button at the top right corner of a Weibo.

These complaints are then forwarded to the Weibo Community Management Centre for professionals employed by Sina Weibo to make a judgment. The processing results will be posted publicly. For actual rumors, the justifications, reasons or clarifications on the rumors by the official accounts will be given so as to guarantee the correctness of the judgment at the detailed page. If the rumor is not deleted, the detailed page also provides the URL of the original reported Weibo.

The author does not use the rumor datasets collected in previous research work because these datasets are not recent data. Many of them are already deleted by the OSN thus it is impossible to obtain the comments of these microblogs. The authors of most of previous work also do not provide their datasets for downloading. Therefore it is not feasible to apply the proposed model on their datasets.

¹Weibo Community Management Centre for false rumor category:
<http://service.account.Weibo.com/?type=5>

As Sina Weibo does not provide an API to collect these rumors, a crawler is implemented. As all rumors published on the platform have been verified by professionals, it is guaranteed that they are actual rumors. Then the author uses APIs provided by Sina Weibo² to get these Weibos' recent Weibo Sets. Since rumor posts only account for a small proportion of all the user's posts, other Weibos in the recent Weibo set can be used to model the normal behaviors of that user. In this phase, 1,257 rumors are collected by the crawler.

For the collection of non-rumors, the author collects Weibos from Sina Weibo's public timeline. Similar to what is done with rumor Weibos, their recent Weibo sets are also crawled. However, one Weibo in the recent Weibo set is randomly selected as the original Weibo with the purpose of avoiding the potential bias of Sina Weibo's public timeline APIs. The author then manually goes through these original Weibos to make sure they are not actual rumors. In this phase, 2,325 non-rumors are collected from public timeline.

Both rumors and non-rumors collected at this phase are called original Weibos and are used to predict their labels. Each post is posted by an individual user. For each user, its recent Weibo set as well as the comments of these Weibos are crawled. In total 167,731 Weibos are collected including both rumors and non-rumors with their recent Weibo sets and 1,501,472 comments.

4.2.2 Feature Selection

Each Weibo $\mathbf{w}$ in the recent Weibo set can be described by the features extracted from itself and its corresponding comments. Thus $\mathbf{w}$ is represented as $\mathbf{w} = (\mathbf{f}^w, \mathbf{f}^c)$, where $\mathbf{f}^w$ and $\mathbf{f}^c$ are two vectors containing the features extracted from $\mathbf{w}$ and its comments

²Sina Weibo API: <http://open.weibo.com/wiki/>

Table 4.1: Weibo based features

Feature	Description
atti_cnt	The no. of users who liked this Weibo.
cmt_cnt	The no. of users who commented this Weibo.
repo_cnt	The no. of users who reposted this Weibo.
sent_score	Sentiment score of the Weibo.
pic_cnt	The no. of pictures posted in this Weibo.
tag_cnt	The no. of #hashtags in this Weibo.
mention_cnt	The no. of @mentions in this Weibo.
smiley_cnt	The no. of smileys in this Weibo.
qm_cnt	The no. of question marks in this Weibo.
fp_cnt	The no. of first person pronouns.
length	The length of the Weibo.
is_rt	Whether the Weibo was a repost.
hour	The time the Weibo was posted.
source	How the Weibo was posted.

respectively.

In order to build the behavioral model of user $\mathbf{u}_i$ (i.e. the author of Weibo $\mathbf{w}_i$) from its recent Weibo set $\mathbf{S}_{w_i,k}$, two types of features are taken into consideration viz. $\mathbf{F}_i^w$ and $\mathbf{F}_i^c$, where $\mathbf{F}_i^w = \{\mathbf{f}_{ij}^w, j = 1, \dots, k\}$ and $\mathbf{F}_i^c = \{\mathbf{f}_{ij}^c, j = 1, \dots, k\}$. The author uses the feature set presented in Table 4.1 to build $\mathbf{F}_i^w$ and these features are time independent.

The features in $\mathbf{F}_i^c$ are extracted from comments in $\mathbf{S}_{w_i,k}$. They are time dependent and will be input into RNN for further analysis. Table 4.2 describes all these features. The first 13 comment based features are based on the content while the last two features are based on the relationship between the author of the original Weibo and the author of the comment. These features are proposed with reference to some previous psychology research in [19] and [20]. People tend to express more skepticism to rumors than credible posts. Sometimes they provide figures and links for rumor busting or seek confirmation from friends. The author is interested in such differences in responses to rumors and non-rumors. Hence, the aforementioned features based on comments are

Table 4.2: Comment based features

Feature	Description
has_pic	Whether the comment contains a picture.
face_cnt	The no. of smileys in the comment.
atti_cnt	The no. of likes of the comment.
mention_cnt	The no. of @mentions in the comment.
tag_cnt	The no. of #tags in the comment.
url_cnt	The no. of urls in the comment.
is_reply	Whether the comment is a reply.
is_repo_cmt	Whether the comment is reposted.
positive_pb	The probability the comment is positive.
no.correct_pb	The probability the comment author agrees.
qm_cnt	The no. of question marks in the comment.
fp_cnt	The no. of first pronouns in the comment.
length	The length of the comment.
o_follow_c	Whether u_w follows u_c .
c_follow_o	Whether u_c follows u_w .

proposed.

Before the data are input into the learning model, the author first normalizes them with natural logarithm $y(x) = \ln(1 + x)$. The reason that z-score or min-max method are not utilized to normalize the data is because they tend to condense the deviation in the data which runs counter to the research purpose.

4.2.3 Recurrent Neural Networks

A recurrent neural network (RNN) [140] is a feed-forward artificial neural network in which connections between units form a directed cycle. This builds an internal state of the network allowing it to exhibit dynamic temporal behavior. The reason that the author would like to observe the temporal behaviors is because the propagation pattern of rumors and non-rumors tend to be different. In other words, other users who read the posts respond differently over time depending on whether the original post is a rumor or not.

Before introducing the details of the RNN module, the author would like to introduce the method used to group the incoming comments of a Weibo into different time slots. The author builds time slots for Weibos having at least two comments. The time range between the first and last comment of the Weibo is first calculated, then the time period is divided into T time slots equally. Finally all comments are distributed into these time slots according to their posting time. In this work, the author chooses $T = 7$ as the number of time slots. According to the information diffusion process presented in [141] and [142], 7 phases is enough to capture the diffusion characteristics of a piece of information regardless of whether it is a rumor or a credible post. If T is set too small, there is too little information for RNN to learn the diffusion process. On the other hand, if T is set too large, the model tends to overfit and thus is unable to learn the “common behaviors” of users.

Each Weibo $\mathbf{w}$ in the recent Weibo set has different number of comments and the comments of $\mathbf{w}$ can be divided into T sets of comments $\{\mathbf{C}_t, t = 1, \dots, T\}$, where $\mathbf{C}_t$ denotes the set of comments falling into the t th time slot. For each comment $\mathbf{c}$, m different comment based features are extracted in $\mathbf{f}^c$. In the RNN module, the input value of the t th time slot $\mathbf{x}_t = (x_{t,1}, \dots, x_{t,m})$ is a m -vector. $x_{t,i}$ denotes the i th comment based feature in $\mathbf{f}^c$, calculated at time slot t , by applying function $g(\cdot)$ on the comments in $\mathbf{C}_t$ using Formula 4.1. The author sets $x_{t,i} = 0$ for $\mathbf{C}_t$ which contains zero comment (i.e. $|\mathbf{C}_t| = 0$). The function $g(\cdot)$ describes the general characteristics of all the comments falling into the same time slot. Functions like sum and mean are tested and they perform similarly. In the proposed model, the mean function is selected for the following experiments.

$$x_{t,i} = \begin{cases} g\left(\bigcup_{c \in \mathbf{C}_t} \mathbf{f}_i^c\right), & |\mathbf{C}_t| > 0 \\ 0, & |\mathbf{C}_t| = 0 \end{cases} \quad (4.1)$$

Furthermore, the proposed one-hidden-layer RNN module is formalized as follows (see Fig. 4.2(a)): given an input sequence $\{\mathbf{x}_1, \dots, \mathbf{x}_T\}$, the hidden states $\{\mathbf{h}_1, \dots, \mathbf{h}_T\}$ are updated at every time step so as to generate the outputs $(\mathbf{o}_1, \dots, \mathbf{o}_T)$.

Figure 4.2: Proposed RNN module.

$\mathbf{h}_t$ is the hidden state in the t th time slot and can be regarded as the “memory” of the network. $\mathbf{h}_t$ is calculated based on the previous hidden state and the input at the current step. Similarly, $\mathbf{o}_t$ is the output vector. $\mathbf{h}_t$ and $\mathbf{o}_t$ are calculated using Formula 4.2 in which U_R, W_R, V_R are the input-to-hidden, hidden-to-hidden, hidden-to-output parameters respectively. When training the parameters the errors between $\mathbf{o}_t$ and $\mathbf{x}_{t+1}$ are calculated since it is expected that the output at step t would learn the influence of the comments in the previous time slots and predict the result in the next slot.

$$\begin{aligned}\mathbf{h}_t &= \tanh(U_R \mathbf{x}_t + W_R \mathbf{h}_{t-1}) \\ \mathbf{o}_t &= V_R \mathbf{h}_t\end{aligned}\tag{4.2}$$

In order to capture higher-level feature interactions between different time slots,

the author further develops a multiple-hidden-layer structure of RNN. The output of the multiple-hidden-layer structure is correspondingly calculated based on the values of the last hidden layer. Fig. 4.2(b) shows the details of a two-hidden-layer RNN. In the figure, $h1(t)$ means the hidden unit for the first hidden layer while $h2(t)$ means the hidden unit for the second hidden layer. The structures of the RNN with more hidden layers are implemented in the same manner.

4.2.4 Autoencoder

An AutoEncoder (AE) [143] is an artificial neural network used for unsupervised learning of efficient codings, i.e. setting the target values to be equal to the inputs. Considering the characteristics of AE, it can be used for anomaly detection [144]. Data experience a dimension reduction process in the hidden layer of AE and in this subspace, normal data and anomalies appear significantly different [145].

Figure 4.3: Proposed autoencoder module.

The details of the proposed one-hidden-layer AE module can be seen in Fig 4.3. An Autoencoder takes an input $\mathbf{X}$ and first maps it (with an encoder) to a hidden representation $\mathbf{H}$ through a deterministic mapping. The hidden value $\mathbf{H}$ is then mapped back (with a decoder) into a reconstruction $\mathbf{O}$ of the same shape as $\mathbf{X}$. The mapping happens through a similar transformation. The input data $\mathbf{X}$ is a matrix in the experiments, where the rows of $\mathbf{X}$ denote different Weibos and the columns of $\mathbf{X}$ represent different features. The following formula concludes the calculation process in which W_A, b and V_A, c are the input-to-hidden and hidden-to-output parameters respectively.

$$\begin{aligned}\mathbf{H} &= \text{SoftPlus}(W_A \mathbf{X} + b) \\ \mathbf{O} &= \text{SoftPlus}(V_A \mathbf{H} + c),\end{aligned}\tag{4.3}$$

where the activation function $\text{SoftPlus}(x) = \ln(1 + e^x)$. *SoftPlus* is selected as the activation function because the range of the function corresponds to the range of the input data and the vanishing gradient problem can be avoided as well.

4.2.5 Combination Model of RNN and AE

For each original Weibo $\mathbf{w}$, a hybrid model combining RNN and AE is built based on the recent Weibo set and its corresponding comments. The structure of the combination model is shown in Fig 4.4. The comment based features of the recent Weibo set, namely $\mathbf{X}^i$, are first input into the RNN module to be analyzed along with the time and the detailed method is described in Section 4.2.3. The output of the RNN module is set as $\mathbf{X}^o$. $\mathbf{X}^o$ is then combined with those features in $\mathbf{f}^w$ extracted from the recent Weibos, namely $\mathbf{X}^w$, and forwarded to the Autoencoder as an input $\mathbf{X}$, i.e $\mathbf{X} = (\mathbf{X}^o, \mathbf{X}^w)$. The

Figure 4.4: The combination of RNN and AE.

output of AE is $\mathbf{O}$.

To train a standard AE, $\mathbf{X}$ is set as the target value. However, in the proposed variant AE, the target value is set as $\mathbf{Y}$ which combines the input of the RNN module $\mathbf{X}^i$ and $\mathbf{X}^w$, i.e. $\mathbf{Y} = (\mathbf{X}^i, \mathbf{X}^w)$. In a standard AE, the input itself is used as the target value because the input is already the observed real value. However in this case, the input value is not the observed real value and already accumulated some errors because of the previous RNN model. Therefore it is more meaningful to set the target value as the real observed value $\mathbf{Y}$ which does not contain errors. A performance comparison between the proposed variant AE as well as a standard AE is presented in Section 4.3. Finally, using a proper detection method based on the errors between the input and output, the potential rumors can be identified.

In the experiments, the input dim m (the number of comment based features) of RNN module is 15 and the hidden dimension of RNN module is set to 15 accordingly. In addition, the number of features in $\mathbf{f}^w$ is 14. Given that the number of time slots T

is 7, the input dimension of the AE module (the number of columns of $\mathbf{X}$) is 119³ and the author simply sets the hidden dimension of AE module as 50.

Note that if all the recent Weibos have too few comments (less than 2), the model cannot learn the social wisdom from so few comments. To speed up the model, the whole RNN module is pruned and $\mathbf{X} = \mathbf{Y} = \mathbf{X}^w$. In other words, under the situation of too few comments, the model only takes into consideration of $\mathbf{f}^w$ and omits $\mathbf{f}^c$.

4.2.6 Rumor Detection

For an original Weibo $\mathbf{w}_i$ and its recent Weibo set $\mathbf{S}_{w_i,k}$, the output and target value of AE module can be represented as $\mathbf{O}_i = (\mathbf{O}_{i1}, \dots, \mathbf{O}_{ik})^\tau$, $\mathbf{Y}_i = (\mathbf{Y}_{i1}, \dots, \mathbf{Y}_{ik})^\tau$, where $\mathbf{O}_{ij}$, $\mathbf{Y}_{ij}$ are the output and target vectors of the j th Weibo in $\mathbf{S}_{w_i,k}$. Using the euclidean norm, the reconstruction error for the j th Weibo can be calculated by:

$$Err_{ij} = \|\mathbf{O}_{ij} - \mathbf{Y}_{ij}\|_2. \quad (4.4)$$

After calculating the error between the input and output of AE, the error is compared with $threshold_i$ to determine whether it is a rumor or not. As what has been mentioned before, user behaviors vary from person to person. Hence the model does not set a fixed threshold for all the users. Instead, the threshold is calculated based on the user's own recent Weibo set and is defined as follows:

$$threshold_i = md_i + \max(1, sd_i) \quad (4.5)$$

where, md_i is the median of all the errors of the recent Weibos $\{Err_{ij}, j = 1, \dots, k\}$

³119=14+7*15. There are 14 Weibo based features and 15 comment based features while each comment based feature has 7 time slots.

while sd_i is the standard deviation. With median and standard deviation, the model can capture the overall characteristics of the behavior habit and the data. The reason that the author does not use $md_i + sd_i$ as the threshold is because if a recent Weibo set does not contain a rumor (i.e. the behaviors of a user remain consistent), the reconstructed errors are stable, thus the sd_i is very small. In this case, $md_i + sd_i \approx md_i$. Therefore half of the weibos in the recent Weibo set shall be regarded as rumors which is not reasonable.

The author further draws the box plot of all the standard deviations of the errors based on rumors and non-rumors in Fig 4.5. It turns out that for non-rumors and their recent Weibo sets, the standard deviations of the errors are smaller and most of them are less than 1. Therefore, by substituting sd_i with $\max(1, sd_i)$ the false positives of the model are reduced.

Figure 4.5: Standard deviation of rumors' and non-rumors' error on recent Weibo set

Then the label of the original Weibo is predicted as follows:

$$isRumor_i = \begin{cases} 1, Err_{i1} > threshold_i \\ 0, Err_{i1} \leq threshold_i \end{cases} \quad (4.6)$$

4.3 Results and Comparisons

A series of experiments are conducted on the dataset described in Subsection 4.2.1 to evaluate the performance of the proposed rumor detection model. All the experiments are run on a Dell PowerEdge R930 Powerful 4-socket 4U Rack Server with 4 Intel Xeon E7-8890 2.5GHz processors and 512 GB of RAM ⁴. In this section, the author presents the results of the proposed learning model as well as some discussions and comparisons. The summary of the results are shown in Table 4.3. In this table, our-ind indicates the proposed model in this chapter. Our-sae adopts a standard AE instead of a variant in the proposed model. Our-agg builds an aggregated model for all users instead of building individual models for individual users. The remaining four are comparison methods. More details are presented in the following subsections.

Table 4.3: Comparisons of Detection Performance of Different Methods

Method	Acc ((95% CI)%)	F1 (%)	Pre (%)	Rec (%)	FPR (%)
[87]	88.52±1.04	80.73	80.00	81.48	8.53
[24]	85.68±1.15	79.00	81.37	76.77	9.51
[106]	85.73±1.15	81.53	74.70	89.74	16.43
[146]	87.07±1.10	81.23	82.81	79.71	8.94
Our-sae	92.10±0.88	88.90	87.69	90.13	6.84
Our-agg	88.72±1.04	84.58	81.29	88.15	10.97
Our-ind	92.49±0.86	89.16	90.36	87.99	5.08

⁴Dell R930: <http://www.dell.com/us/business/p/poweredge-r930/pd>

4.3.1 One Hidden Layer vs. Multiple Hidden Layers

As introduced in Section 4.2, the author implements both one- and multiple-hidden-layer learning models to compare their detection performance. The detection results with different number of hidden layers are shown in Fig 4.6.

Figure 4.6: Performance of learning models with different number of hidden layers.

The only difference among the different models are the number of hidden layers and all other settings are the same. It is observed from the results that the two-hidden-layer model outperforms the one-hidden-layer model in all 5 metrics and the accuracy reaches as high as 92.49%. However, when the number of hidden layers increases, the accuracy and F1 measure slowly decrease and the false positive rate increase dramatically which can be attributed to the overfitting problem. Therefore, the number of hidden layers is set to 2 for the following experiments.

4.3.2 Standard Autoencoder vs. Proposed Variant Autoencoder

It is illustrated in Section 4.2.5 that the Autoencoder used in the proposed model is slightly different from a standard Autoencoder where the target value is concerned. To better support the validity of the variant AE, its performance is compared with a standard AE (i.e. the target value is set to the same as the input which is $\mathbf{X}$). In Table 4.3, Our-sae and Our-ind show the results of a standard AE and the variant AE respectively. It is observed that the variant AE performs better than the standard AE except for the recall.

4.3.3 One Aggregated Model vs. Individual Models

The most significant assumption of the proposed model is that users behave differently to rumors and non-rumors. Similarly, other users' responses to rumors and non-rumors are different. Many previous research work is based on implementing classifiers for rumor detection. However, they do not distinguish these differences among different users whereby the posting behaviors are quite varied which affect the detection performance.

In an observed extreme case, most postings of a user somewhat bear the features of rumors. If the model does not distinguish the user from others, most of the user's postings would be predicted as rumors. However, when observing the recent microblogs posted by the said user, it is discovered that the user's posting behaviors are quite consistent. In other words, these differences of posting behaviors in rumors and non-rumors are more meaningful when they are observed based on individual users.

In this work, the author carries out such a comparison to further support the opinion that individual models are more effective in detecting rumors than a single

aggregated model. In Table 4.2, our-ind represents the two-hidden-layer learning model on individual users while our-agg denotes the two-hidden-layer aggregated model on all users. Other parameters are set as per the description in Section 4.2. It is observed that only the recall is slightly smaller than that of the aggregated model, while all the other 4 metrics of the individual model are much better. The false positive rate of the aggregated model is rather high (10.97%) which is consistent with the observations mentioned above. This further proves that it is more effective to build the learning model on individual users instead of all users for rumor detection.

4.3.4 Proposed Model vs. Other Methods

Since the proposed method is based on user behaviors, 4 other pieces of recent work which also exploit similar features are selected for comparisons. The results are shown in the first 4 rows of Table 4.3. The author implements their methods and applies them on the collected dataset described in Section 4.2.

Liang’s method [87] is based on a SVM classifier which is a supervised learning method. The proposed unsupervised learning model (denoted as our-ind) performs much better than theirs in all metrics. This is because the proposed model further incorporates comments of the original Weibos. In other words, the model learns more information before making a prediction.

The other 3 compared methods are all unsupervised learning methods. For [106], the proposed model achieves a good recall while their precision is less than 80% along with a rather high FPR. For the other two methods, the proposed method outperforms them in all metrics. It is worth mentioning here that [146] is an early work of the author without the RNN module. The comparisons show that the addition of RNN does improve the detection performance significantly. This highlights the huge effect

of crowd wisdom.

To conclude, the proposed model achieves a satisfactory detection performance when compared to other methods.

4.4 Conclusions

Rumors and different kinds of misinformation on OSN have posed one of the most serious problems to both OSN service providers and normal users. Thus it is of great value to detect these rumors automatically to prevent the potential public issues they may cause. However, because rumors only account for a tiny proportion of all the posts on OSN, it is sometimes very difficult to obtain enough data for training. Therefore the author proposes an unsupervised learning model for rumor detection based on users' behaviors.

In this chapter, the author proposes a combination of RNN and variant AE to learn the normal behaviors of individual users. The errors between the outputs and inputs of the model are used to describe the deviation degree of Weibos and are compared with the self-adapting thresholds to determine whether it is a rumor or not. The author implements both the one-layer and multiple-layer structure of the learning model and the two-layer model can achieve a high accuracy of 92.49% and F1 of 89.16% which is much better than the compared methods. The author also compares the performance of one aggregated model and individual models for the different users and conclude that the individual models perform better. This further verifies the assumption that the behaviors of different users vary from person to person and that the proposed model is able to exploit this in the detection of rumors.

Chapter 5

Leveraging Social Media News to Predict Stock Index Movement Using RNN-Boost

In this chapter, to illustrate the good aspects of OSN, the author presents a hybrid model named RNN-Boost to predict the stock market index based on the news content collected from OSN.

In this work, the author keeps an eye on both the technical and fundamental data. For the fundamental data, the news content posted on online social media is utilized to estimate the public moods. This is because more people use social media as an important platform to get news information. In addition, such news is more succinct and is easier and faster to spread. General news instead of specific financial related news is exploited in this work because the non-financial related news also has a latent relationship with the stock price movement over time which can be learned and captured by the Recurrent Neural Networks (RNN) in the proposed model.

The rest of this chapter is organized as follows. Section 5.1 illustrates the overview of the proposed learning model, the RNN-Boost. Section 5.2 explains the details about the data collection process, the adopted features as well as the proposed prediction model. Section 5.3 presents the experimental results as well as the comparisons with other methods. Section 5.4 summarizes the whole chapter.

5.1 Overview of the Model

As what has been introduced in Chapter 2, public moods have a significant influence on stock market. To quantify public moods, the author proposes to analyze the news content posted on social media by official accounts. These official accounts usually have a large number of followers which means the content they post can have a huge outreach. In this work, the author carefully selects official accounts from the Chinese largest online social website - Sina Weibo and collects their posts.

Thereafter, LDA features are computed and sentiment analysis is performed. LDA features and sentiment features are together called content features. These content features are combined with other technical features calculated based on historic stock data and then input into the RNN with Gated Recurrent Units (GRU) to predict the stock price. The errors between the predictions and the real values are calculated so as to adjust the weights of the different training samples and then input into the next RNN for the next round of training. The boosting process is adaptive in the sense that subsequent RNN models are tweaked in favor of those samples wrongly predicted by the previous RNN models. In the end, the final prediction is combined using the weighted median to represent the final output of the boosted RNN, that is, the prediction for the stock index price of the next day. With this predicted output, the model is also able to predict the movement direction of the stock (i.e. whether the stock price goes

Figure 5.1: Overview of the proposed model.

up and down) of the next day.

The overview of the proposed hybrid model can be seen in Fig 5.1. This model can be easily adapted to predict the stock index of other financial markets as long as the news content posted on social media can be obtained. The equivalence of Sina Weibo in this work is Twitter.

5.2 Methods and models

In this section, the author first introduces how the official accounts are selected and how the news content posted by them is collected. Then the author describes the content and technical features adopted for the stock market index prediction. Finally the author presents the details of the proposed hybrid model, the RNN_Boost.

5.2.1 Data collection

In this work, the time period from 1 Jan 2015 to 14 Feb 2017 is chosen to carry out the experiments ¹. The time period includes 513 trading days in total. The reason that the author does not analyze a longer time period is that Sina Weibo started to act as a platform for official accounts around 2012 and became rather popular after 2014. To fully utilize the content features, the aforementioned period is chosen for the experiments.

The data needed include two parts: the historic price of the stock market and the news content from social media.

With China being the second largest economy in the world in 2016 [147], and Shanghai being one of the largest global financial centers, the China Shanghai Shenzhen 300 Stock Index (HS300), also known as Chinese Stock Index (CSI), is gradually becoming a critical financial indicator which is followed closely by investors all over the world. The index is compiled by the China Securities Index Company Ltd and is used to reflect the price fluctuation and performance of 300 stocks traded in the Shanghai and Shenzhen stock exchanges. The historic price of CSI can be downloaded from Yahoo Finance ². The author implements the proposed model with the purpose of predicting CSI.

To utilize the content features which facilitate CSI prediction, Sina Weibo is chosen as the source of the needed news content. Sina Weibo is one of the most popular OSN in China which has about 351 million active users by September 2017 [3]. Sina Weibo is often regarded as an equivalence of Twitter in China because they both have many common OSN features with respect to users and posts. This makes it possible for the

¹The news dataset is available at <http://blogs.ntu.edu.sg/weilingchen/2017/12/13/research-dataset-release-for-stock-index-prediction-paper/>

²Yahoo Finance: <https://www.finance.yahoo.com/>

application of the proposed hybrid model to be migrated from one platform to another.

Before introducing the data collection process, the author would introduce more details about the account system of Sina Weibo. All users are able to register a normal account with Sina Weibo. With the normal accounts, the users are able to view, share and write posts. To increase the influence of posts or attract more followers, it is possible for the normal users to request verification of their accounts from Sina Weibo. There are different types of verified accounts and what should be emphasized here is the blue type verified accounts. Entities behind blue type accounts are usually verified organizations including companies, institutes, media, etc. Once the organization clears the verification process, a blue “V” will be added behind their user names.

In this work, the author collects news content from the Weibos posted by the blue verified accounts. In total, 95 famous blue verified accounts under media categories are selected. These accounts usually have a large number of followers (i.e. more than 90k), ranging from National newspapers, provincial newspapers and magazines to famous TV news programs shown at national TV channels. Most of these accounts post general news and only a small portion focuses on financial related news. The reason the author does not only choose financial news agencies is because the impact of news on the stock market is usually latent. As such, political news and sometimes even the rumors of celebrities would affect the stock market. In total, 808,283 Weibos posted by these accounts are collected during the time period mentioned before.

To avoid losing followers, these verified accounts sometimes also post jokes, greetings, etc. apart from pure news. To filter these non-news content, a filter is implemented to sieve out those Weibos using particular keywords and symbols (e.g. good morning, humor of the day and so on). Some news content might be mistakenly deleted in the preprocessing step. However if the news is significant enough, it is highly possible that it would be posted by other accounts without all being filtered out.

5.2.2 Feature engineering

As per the introduction in Section 5.1, the features used for predicting stock index include content features and technical features which would be explained in details in the following subsections.

5.2.2.1 Technical features

Technical features are calculated based on historic dataset of the stock market. Denote a given trading day as t and the nearest trading day before t as $t - 1$. The description and the formula of the features are listed in Table 5.1. The first 5 basic features of each trading day shown in the table can be directly obtained from the dataset downloaded from Yahoo Finance. The rest of the features are calculated based on the first 5 basic features.

Table 5.1: Basic Features and the Formulas

No.	Feature	Description or Formula
1	Opening Price (O_t)	The price at which HS300 first trades upon the opening of the market on t .
2	Closing Price (C_t)	The final price at which HS300 is traded on t .
3	Highest Price (H_t)	The highest price at which HS300 is traded during day t .
4	Lowest Price (L_t)	The lowest price at which HS300 is traded during day t .
5	Volume (V_t)	The number of shares traded in HS300 during day t .
6	Price change	$C_t - C_{t-1}$
7	Price limit	$(C_t - C_{t-1})/C_{t-1}$
8	Volume change	$V_t - V_{t-1}$
9	Volume limit	$(V_t - V_{t-1})/V_{t-1}$
10	amplitude	$(H_t - L_t)/C_{t-1}$
11	difference	$(C_t - O_t)/C_{t-1}$

5.2.2.2 Content features

Content features can be further divided into sentiment features and LDA features.

Suppose there are n_i pieces of news content collected from the selected accounts we have introduced in the last subsection in day i . We follow the following procedure to calculate the sentiment feature of the day.

Firstly, we follow the method described in [148] to build the sentiment dictionary. There are two categories of keywords in the sentiment dictionary, namely positive and negative. Positive keywords have a positive impact on the stock market in the following day, i.e. the stock price goes up in the following day after the Weibos were posted. Similarly, negative keywords have a negative impact on the stock market, i.e. the stock price goes down in the following day after the Weibos were posted.

Secondly, keywords from the collected Weibos are extracted and the stop words are deleted. The priori probability of every keyword m for each class are calculated as follows.

$$P(m|c) = \frac{1}{total_c} \sum_{i=1}^{n_c} count(m, w_i) \quad (5.1)$$

Where $count(m, w_i)$ calculates the number of times the keyword m occur in Weibo w_i and $total_c$ is the total number of times all the keywords occur in the class.

Thirdly, suppose there are n_w keywords in a suspicious Weibo w , calculate its likelihood to fall in each class.

$$P(w|c) = P(c) \prod_{i=1}^{n_w} P(m_i|c) \quad (5.2)$$

Where $p(c)$ is the prior probability of class c and in this case,

$$P(c_{pos}) = \frac{total_{c_{pos}}}{total_{c_{pos}} + total_{c_{neg}}} \quad (5.3)$$

Fourthly, the sentiment feature of Weibo w is calculated as follows:

$$p(w) = \frac{P(w|c_{pos})}{P(w|c_{pos}) + P(w|c_{neg})} \quad (5.4)$$

Finally, suppose there are n_i Weibos collected in day i , the overall sentiment feature of the day can be calculated as

$$P_i = \frac{1}{n_i} \sum_{j=1}^{n_i} p(w_j) \quad (5.5)$$

The Latent Dirichlet Allocation (LDA) [149] features from the collected Weibos are also computed. LDA is an example of a topic model. It is expected that some topics of news have a higher impact on stock market than others. The LDA features are employed to capture such indicators so as to facilitate stock market prediction.

In this work, Python Library scikit-learn³ is used to calculate the LDA features with online variational Bayes algorithm. Each row of the output matrix represents a topic vector of a Weibo which is denoted as $\mathbf{q}_j$ (j indicates the j th Weibo in day i). Each element in the vector represents the probability of the Weibo belonging to the specific topics. Therefore, the overall LDA feature vector $\mathbf{P}_i$ of news content collected in day i for are calculated as follows:

$$\mathbf{P}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{q}_j \quad (5.6)$$

³scikit-learn: <http://scikit-learn.org/stable/index.html>

5.2.3 Recurrent Neural Networks

RNN are frequently used in time series forecasting tasks. RNN are regarded as recurrent because they perform the same task for every element in a sequence and the current output depends on the previous computations. In RNN, connections between units form a directed cycle. The structure of the one-hidden-layer RNN is shown in Fig 5.2.

Figure 5.2: One-hidden-layer RNN module

In the proposed RNN model, the input value of the t th day $x_t = (x_{t,1}, \dots, x_{t,m})$ is a m -vector indicating features described in the last subsection. The algorithm iterates over the following equation:

$$\begin{aligned} h_t &= \tanh(Ux_t + Wh_{t-1} + b) \\ o_t &= \tanh(Vh_t + c) \end{aligned} \quad (5.7)$$

in which h_t is the hidden state calculated based on the previous hidden state h_{t-1} and the input x_t at the current time step. o_t is the output which is the predicted value.

It can be regarded as an indicator of the stock price for the next day. U, W and V are the input-to-hidden, hidden-to-hidden, and hidden-to-output parameters respectively which are trained in the RNN.

Theoretically, RNN are able to learn information in arbitrarily long sequences. However, they are only able to look back a few steps in actual practice due to the vanishing gradient problem which makes it difficult to learn long-term dependencies. Grated Recurrent Unit (GRU) is then proposed to tackle the issue by employing a gating mechanism. Fig 5.3 shows the simple structure of a GRU unit.

Figure 5.3: Structure of GRU unit

r_t, z_t are respectively the reset and update gates in GRUs. The reset gate determines how to combine the new input x_t with the previous memory h_{t-1} in computing s_t . s_t can be regarded as a “candidate” hidden state. The calculation of h_t uses the update gate z_t to define how much of the previous memory to keep. The following equations are used for calculating a GRU hidden unit:

$$\begin{aligned}
z_t &= \sigma(x_t U^z + h_{t-1} W^z + b^z) \\
r_t &= \sigma(x_t U^r + h_{t-1} W^r + b^r) \\
s_t &= \tanh(x_t U^s + (h_{t-1} \odot r_t) W^s + b^s) \\
h_t &= (1 - z_t) \odot s_t + z_t \odot h_{t-1},
\end{aligned} \tag{5.8}$$

in which $\sigma(x)$ is a hard sigmoid function and $\odot$ denotes component-wise multiplication.

In the experiments, the author applies a two-hidden-layer GRU module to capture the higher-level feature interactions between the different time steps. The units in the second hidden layer are calculated similarly as in the first hidden layer.

The training process of RNN in this work is as follows. The inputs of the RNN model are the feature vectors during a given time period from t_0 to t_n , the training data and the observed values (i.e. target values) are $\{x_t, t = 1, \dots, n\}$ and $\{y_i, i = 1, \dots, n\}$ respectively. What is worth mentioning here is that the closing price is not directly used as the target value because it fluctuates dramatically making it more difficult to predict. Instead, the dependent variable is computed as $y_i = C_i/C_0 - 1, i = 1, \dots, n$, where $C_0, \dots, C_n$ are the closing prices of CSI.

In this work, the historic data of previous s days are used to predict the stock price of the next trading day. The initial parameters of the GRU module are randomly set using a predefined seed to guarantee repetitiveness of the RNN models. To train the parameters, the GRU module uses back propagation technique to minimize the difference between output o_t and the observed y_t . The parameters are updated iteratively when new data come in.

To evaluate the performance of the proposed model, the total time period is divided into two phases. The model uses the data in $t_0 \sim t_{m-1}$ to train the GRU parameters and predicts the dependent data in $t_m \sim t_n$. In the second phase, the GRU parameters are updated after a new prediction is made. In other words, the prediction y_{i+1} is computed based on the updated parameters after $\{x_i, i = t - s, \dots, t - 1\}$ and y_t are input into the GRU module for training. This simulates the real-world situation because new stock price can always be obtained on a daily basis and input into the model for training.

5.2.4 Adaboost

Adaboost is the abbreviation for “adaptive boosting” which is a machine learning technique proposed by Yoav Freund and Robert Schapire [150]. It is often combined with other learning methods to improve the accuracy. The earlier version of Adaboost is conjuncted with a decision tree and works as an ensemble classifier. Later, Drucker proposed an Adaboost regressor which is known as Adaboost.R2 so that it can deal with regression problem [151].

In boosting process, estimators (i.e. learning methods) are trained sequentially. Generally, training data are input into the first estimator and those data samples whose predicted values differ most from the observed values will be noted. The weight of these data samples are then increased when they are input into the second estimator. The weight (i.e. coefficient) of the estimator is also calculated based on its overall errors. In the end, the output of the final model is combined using the weighted median, whereby predictors with more “confidence” are weighted more heavily. An overview of the Adaboost.R2 algorithm can be seen in Fig 5.4.

Since the purpose of this work is to predict the stock price instead of predicting only

Figure 5.4: An overview of Adaboost.R2

the movement direction of the stock price, the author adopts the idea of Adaboost.R2 and combines it with RNN to fulfill the aforementioned requirement. Details of the proposed hybrid model are introduced in the next subsection.

5.2.5 RNN-Boost

According to [148], the prediction performance of a single RNN is far from satisfactory. This inspires the author to use boosting method to improve the prediction accuracy of the stock market index. Algorithm 1 describes the details of the hybrid method. Lines 2 to 10 illustrate the boosting process whereby it iterates multiple times to generate multiple RNN regressors. Thereafter, each estimator (i.e. RNN) and its weight are stored and the details of the boosting process can be seen in Algorithm 2. What is worth mentioning here is that the boosting process would terminate early if the estimator weight becomes 0 as shown in Lines 5 and 6 (i.e. the errors of the estimator is larger than 0.5). This is followed by Lines 11 to 18 showing how to calculate the final prediction output. In this case, the model uses the weighted median (Lines 16 and 17) to combine the estimators which have been stored in the previous step.

Details of the boosting process are illustrated as Algorithm 2. The model first trains

Algorithm 1 The proposed hybrid model RNN-Boost

Require: Features X , labels y , number of estimators n **Ensure:** Predicted stock index prices

```

1: function RNN_BOOST( $X, y, n$ )
2:    $sample\_weight \leftarrow initialize\_weight()$  ▷ Train starts here.
3:   for  $iboot = 0 \rightarrow n - 1$  do
4:      $sample\_weight, estimator\_weight, estimator\_error \leftarrow$ 
        $boost(X, y, sample\_weight)$ 
5:     if  $estimator\_weight = 0$  then ▷ Early termination.
6:       break
7:     end if
8:      $save(iboot, estimator\_weight, estimator\_error)$ 
9:      $normalize(sample\_weight)$ 
10:  end for
11:   $Predictions \leftarrow \emptyset$  ▷ Prediction starts here.
12:  for  $iboot = 0 \rightarrow n - 1$  do
13:     $pred \leftarrow RNN\_predict(iboot)$ 
14:     $append(Predictions, pred)$ 
15:  end for
16:   $median\_index = find\_median\_index(Predictions)$ 
17:   $output \leftarrow predict(Predictions, median\_index)$ 
18:  return  $output$ 
19: end function

```

the RNN as described in Section 5.2.3. Then the square error of each data sample is computed so as to calculate the estimator error (See Lines 3 to 5). What should be noted here is that if the estimator is larger than 0.5, the boosting process would return directly so as to save computing time. Thereafter, the weight of the estimator and data samples are adjusted accordingly (see Lines 9 to 11).

Algorithm 2 Boosting method for RNN

Require: Features X , labels y , $sample_weight$

Ensure: $sample_weight, estimator_weight, estimator_error$

```

1: function BOOST( $X, y, sample\_weight$ )
2:    $estimator \leftarrow RNN\_train(X, y)$ 
3:    $error\_vect \leftarrow RNN\_predict(estimator) - y$ 
4:    $error\_vect \leftarrow square(error\_vect / max(error\_vect))$ 
5:    $estimator\_error = sum(sample\_weight * error\_vect)$ 
6:   if  $estimator\_error > 0.5$  then ▷ Early termination.
7:     return  $sample\_weight, 0, estimator\_error$ 
8:   end if
9:    $beta \leftarrow estimator\_error / (1 - estimator\_error)$ 
10:   $estimator\_weight = learning\_rate * log(1/beta)$ 
11:   $sample\_weight \leftarrow sample\_weight * power(beta, (1 - error\_vect)) * learning\_rate$ 
12:  return  $sample\_weight, estimator\_weight, estimator\_error$ 
13: end function

```

5.3 Results and Comparisons

In this section, the author discusses several important aspects of the proposed model and compares it with other prevalent methods.

To test the performance of the proposed model, experiments are run on a Dell PowerEdge R930 Server. It has 4 Intel Xeon E7-8890 2.5GHz processors and 512 GB of RAM⁴.

As detailed above, the experiments are done using the data (i.e. historic stock price

⁴Dell R930: <http://www.dell.com/us/business/p/poweredge-r930/pd>

and news content from OSN) collected from 1 Jan 2015 to 14 Feb 2017.

5.3.1 Single RNN with different feature sets

First, the author tests the usefulness of the proposed features, especially content features. In this phase, the two-layer RNN with GRU are adopted to predict the stock market index in order to test the different configuration of feature subsets. The results are shown in Table 5.2. *Tech* and *Sent* indicate the technical and sentiment features respectively. *LDAx* denotes LDA features with x number of topics as predefined parameters.

Table 5.2: Prediction results for different feature subsets

Feat. Subset		Acc(%)	MAE(%)	MAPE(%)	RMSE(%)
Tech	Avg	53.13	1.74	32.06	2.89
	Max	55.99	1.82	34.73	3.03
Tech+Sent	Avg	64.42	1.39	23.97	2.23
	Max	66.99	1.50	27.57	2.41
Tech+Sent+LDA10	Avg	63.97	1.39	25.30	2.16
	Max	69.68	1.40	24.51	2.19
Tech+Sent+LDA20	Avg	65.28	1.44	26.21	2.17
	Max	70.17	1.50	27.95	2.35
Tech+Sent+LDA30	Avg	64.09	1.47	29.41	2.20
	Max	67.48	1.45	30.72	2.13

The reason that average and maximum accuracies are both provided is because it is observed that when using different initial parameters, the performance of the model varies. The author is more interested in the direction accuracy, thus the “max” here indicates the best performance in accuracy and the other 3 metrics are listed accordingly. To make the experiments repeatable, different seeds are used to initialize the proposed model and the average is calculated based on 100 rounds of experiments using seeds from 0 to 99.

It is observed from Table 5.2 that when the sentiment feature is adopted, the

accuracy increases significantly from 53.13% to 64.42% and all the three errors decrease moderately. For the LDA features, the number of topics is adjusted and multiple experiments are run accordingly. Table 5.2 does not display all the results but only the most representative ones. 20 turns out to be a reasonable choice for the number of topics used as the parameter in the computation of LDA features. This is reasonable because when the topic number is too small, it cannot fully describe the features of the news content while if it is too large, it may cause over-fitting issues.

5.3.2 Single RNN vs. RNN-Boost

To evaluate the improvement of the proposed RNN-Boost over the single RNN, the topic number is set to 20 for both models. The experimental results are shown in Table 5.3.

Table 5.3: Comparisons between RNN and RNN-Boost

Model		Acc(%)	MAE(%)	MAPE(%)	RMSE(%)
Single RNN	Avg	65.28	1.44	26.21	2.17
	Max	70.17	1.50	27.95	2.35
	Min	61.86	1.43	25.13	2.17
RNN-Boost	Avg	66.54	1.32	24.31	2.05
	Max	70.17	1.32	24.85	2.09
	Min	64.06	1.32	23.70	2.05

Similar to the last subsection, the “min” accuracy is included to Table 5.3 to show the worst results of the model. It is observed from the table that the average accuracy increases moderately after using RNN-Boost and the worst result of the model becomes better. Another advantage of RNN-Boost is that it is not necessary to pay attention to the selection of seed for initializing the RNN-GRU model as it tends to preserve the better prediction results of the different RNN models.

5.3.3 Comparisons with other methods

The author also compares the proposed model with some other prevalent methods. Two artificial intelligence methods are selected. One is the work done by Chen et al. [148] whose model is based on RNN and the other is the traditional Artificial Neural Networks (ANN) based on Multi-Layer Perceptron (MLP). In addition, the author compares with traditional machine learning methods, namely, linear regression and Support Vector Regression (SVR). Cakra et al. [152] adopt multinomial linear regression to deal with the historical price while SVR is often used in regression tasks. The same features described in Section 5.2 are used for the training and prediction. The comparison results are shown in Table 5.4.

Table 5.4: Comparisons with other methods

Method	Acc((95% CI)%)	MAE(%)	MAPE(%)	RMSE(%)
Proposed	66.54±4.57	1.32	22.31	2.05
Chen's	64.42±4.63	1.39	23.97	2.23
Cakra's	51.60±4.84	1.29	25.42	2.13
MLP	54.66±4.82	6.04	78.37	7.12
SVR	53.92±4.82	9.15	133.90	9.71

The experimental results show that the proposed RNN-Boost performs better than the compared methods in Acc, MAPE and RMSE but slightly lesser than Cakra's method in the MAE metric. However, the accuracy of the proposed method is much better than Cakra's.

5.4 Conclusions

Stock volatility prediction is always regarded as a challenging yet attractive research task. The work illustrated in this chapter is an attempt at innovative Financial Technology (FinTech) which applies Artificial Intelligence (AI) to tackle financial problems

and develop novel applications. The author proposes a model called RNN-Boost to process both the fundamental and technical features of historic data and achieves a promising prediction result. With little refinement, the proposed model can be deployed for other intelligent trading purposes.

Chapter 6

Conclusions and Future Work

This thesis discusses the boon and bane of OSN and provides methodologies and applications of alleviating the negative impacts of OSN and benefiting from its merits. In this chapter, the author summarizes the whole thesis and explores the possible future directions for extending the work.

6.1 Conclusions

6.1.1 Real-time Content Polluter Detection Framework from the Users' Perspective

During the last decade, OSN have attracted billions of users. These users share their opinions and comment on other users' posts. However, much of the generated content is meaningless and of little value. Too much low-quality content appearing in the user timelines affect their browsing experience severely and make it difficult for users to browse the content they are really interested in.

In Chapter 3, to define low-quality content comprehensibly, the author first uses EM algorithm to coarsely classify low-quality tweets into four categories. Then the author designs a survey based on this preliminary study and involves 211 participants to understand their opinions about low-quality content from the users' perspective. Thereafter, the author gives a clearer definition of low-quality content as a large amount of valueless or fraudulent content which hampers users from browsing useful content. The low-quality content include advertisements, automatically generated content, meaningless content, click baits, etc.

Based on the proposed definition of content polluters, the author further proposes 23 direct features (i.e. features that can be obtained in one API request to Twitter) and 9 indirect features (i.e. features that cannot be obtained in one request) to distinguish low-quality content from normal posts. The author also applies 3 metrics namely AUC, IG and Chi-square to evaluate the significance of the proposed features and observes that mention proportion, url proportion and favorites count are the most powerful features.

Apart from the direct and indirect features, the author also carries out word level analysis on the text content of posts. In this phase, the author analyzes low-quality content and summarizes the terms with high frequency to create a blacklist keyword dictionary. Each term in this dictionary can be regarded as one feature and these features are combined with direct and indirect features for detecting low-quality content.

Finally, the author implements a Random Forest classifier and a SVM classifier to perform low-quality content detection. The performance of the two classifiers are tested on different feature subsets. The experimental results show that random forest classifier outperforms the SVM classifier in all feature subsets. The author has shown that the proposed framework performs well in detecting low-quality content with respect to both time efficiency and detection rate.

6.1.2 Unsupervised Rumor Detection Model based on User Behaviors

Rumors spread on OSN can sometimes cause serious negative social impacts and it is usually impossible for humans to manually inspect millions of posts created everyday. Thus an automated technique to detect rumors on OSN is of high practical value.

In Chapter 4, the author proposes an unsupervised rumor detection model. Different from previous work, for the first time, the author treats the rumor detection task as an anomaly detection instead of classification problem. There are two assumptions behind this idea. Firstly, the behaviors of users posting rumors and normal posts are different and so are the responses from the other users. Secondly, rumors only account for a tiny proportion in the users' posting history. Therefore, a rumor can be viewed as an anomaly in the user's posting history.

Based on previous work in computer science and social psychology, the author proposes 14 Weibo based features and 15 comment based features to capture the behavior differences related to rumors and non-rumors. These features are then input into the proposed model which is a combination of RNN and AE. The RNN is used to process the comments of the suspicious Weibo and capture its spread pattern over time. The recent posts of the user are represented by feature vectors and then input into the AE for anomaly detection. The square errors between the input and output of AE are indicators of the deviation degree. A larger error implies a high probability of being a rumor. The author also proposes a self adaptive threshold of the error for detecting rumors.

For the dataset used for the experiments, the author writes a crawler to collect rumors from Sina Weibo Service Platform where rumors are debunked by professionals hired by Sina Weibo. Meanwhile, the author collects normal posts from the public

timeline of Sina Weibo. The author applies the proposed model on the dataset and compares with several variant models as well as other prevalent methods. The experimental results demonstrate the proposed model performs well in detecting rumors on OSN.

6.1.3 Hybrid Model for Predicting Stock Market Index

Financial market prediction based on text mining and sentiment analysis is just emerging and has attracted much attention from both the academia and industry. News is observed to have an impact on the stock market but the prediction performance of existing work still has much room for improvement.

Therefore, in Chapter 5, the author proposes a hybrid model named RNN-Boost to predict stock market index. To capture the market trend, the author proposes 11 technical indicators and 2 content features, namely, the sentiment and LDA features. For the sentiment feature, the author builds a sentiment dictionary using posts which are posted the day before when the stock prices have changed significantly. The dictionary is used to calculate the sentiment feature which reflects the public moods from the news content on Weibo.

The author first uses a RNN model with GRU to predict the stock index. To train the model, the stock data of each day are represented with a feature vector. To predict the stock index of the following day, the feature vectors of the previous s days are taken as input. Since the performance of a single RNN is not satisfactory enough, the author further proposes the RNN-Boost which incorporates multiple RNN models using Adaboost. In the boosting process, RNN models are trained sequentially. The weight of each RNN model is computed based on its overall errors. The final output of RNN-Boost is combined using the weighted median.

The experiments are performed on the Chinese Stock Market and the news content is collected from Sina Weibo. The author compares the prediction performance of using different feature subsets and discusses the parameter selection for LDA. In addition, the experimental results of the single RNN and RNN-Boost are also compared. The experimental results demonstrate a significant improvement from previous research work.

6.2 Future Work

6.2.1 Content Polluter Detection

It can be seen in the survey conducted by the author that 40.76% of the participants believe that all the content which they are not interested in should be filtered as low-quality content. This interesting discovery indicates the necessity and value of a content filter for disinterested content on OSN.

In addition, the rule set described in section 3.2 is for the purpose of research but is still too general for individuals. Different people have their own definitions for content polluters. Thus in the future, the author would like to add more customized configuration to the current work to implement a more personalized content filter not only focusing on general low-quality content. It is meant to automatically learn what the user is not interested in and hide them from the users' timeline.

Moreover, the current detection work is based on tweets not on users. Other than detecting content polluters, the author would like to further characterize the behavioral patterns of these malicious users so as to filter them out from the legitimate users and then detect the malicious user community. Efforts will also be put in detecting campaigns started by these malicious communities.

6.2.2 Rumor Detection

Misinformation sometimes can cause serious social issues. An automatic rumor detection model is not enough to tackle the issue it may cause. The author realizes there are at least two directions to extend the work described above.

The first research direction is to predict the influence of rumors. Apart from barely detecting the rumors, it would be meaningful and useful to predict the impact of the rumors on both the society and the individuals. Some metrics can be used to represent the impact of the rumors, e.g. the outreach of a rumor post. Rumors emerge almost every day; thus the prediction of their impact is significant. When the resources are limited, it is better to assign priorities to deal with those which have a larger outreach to minimize potential issues and risks.

The second research direction is to explore strategies to dampen the impact of rumors. This can be rather useful when real-world emergencies happen. In these cases, many relevant rumors emerge during a short period of time and it is difficult for humans to verify or debunk them one by one manually. By adding strategies to reduce the negative effects of misinformation to the current model, it turns from a passive observer to an active fighter to dampen the harm rumors may cause. A naive idea about strategies for dampening rumors is to gain some insights from the spread of epidemics. The core is to find the key nodes in the networks. When it comes to OSN, the strategy should be able to target influential users and take immediate actions (e.g. to delete the misinformation posted by them).

6.2.3 Stock Index Prediction

For the RNN-Boost proposed in Chapter 5, the author summarizes the following aspects which would require further efforts to extend this work.

Firstly, it is observed sentiment analysis has a significant impact on the prediction of stock market. In this thesis, the author adopts a sentiment dictionary created in her earlier research work. The author plans to try other sentiment dictionaries and compare the results in the future work.

Secondly, the author would focus more on feature engineering especially the selection of technical indicators. The current model only uses several simple technical indicators while in the traditional analysis of stock market, more advanced rules like moving average rules, relative strength rules, filter rules, trading range breakout rules and other rules are frequently used. A large number of features can be generated using these rules. In addition, the author would like to develop a mechanism which can automatically select the more significant technical features so as to improve the prediction performance.

Thirdly, the current model only implements the sentiment and LDA features for content analysis. However, the advancement in semantic analysis and syntax analysis still have not been fully utilized in the financial prediction field. Much of the previous work only focuses on word occurrence and has not incorporated more advanced technique like WordNet which would be a good research direction. Meanwhile, syntax analysis has attracted less attention. Techniques like parse-trees used for text mining could be taken into consideration for improving the prediction performance of the model.

Fourthly, the author only uses the proposed model to predict the stock index instead of the individual stocks. This is because it is difficult to seek industry specific news which is more related to particular stocks whereas general news may have little impact on them. In future work, the author would like to incorporate more news sources instead of only using news content collected from OSN to get a better understanding of the public moods in the different industries.

6.2.4 Other Research Directions

OSN have attracted billions of users. A huge number of active users create a huge amount of content every day which makes the OSN a valuable source of information. Apart from the research work which has been done in this thesis, the author proposes some other future directions which may be of value to explore deeper.

- **Astroturf campaign detection.** Astroturfing is a tactic to simulate grassroots support for a message or organization (e.g. political, advertising, religious or public relations) with the purpose of shaping public opinions. This research topic can be regarded as an extension to content polluter and rumor detection. Astroturfing cannot be classified as either a content polluter or a rumor. The astroturfing messages may carry true information which makes it even more difficult to detect. One possible research direction is to study the spread pattern of astroturf campaigns and propose relevant features to distinguish them from the normal posts. With a model which is able to detect astroturf campaign automatically, it alleviates the possibility of regular users being deceived by plausible content which is delicately designed for astroturf campaigns and makes it possible for these users to keep neutral about particular topics instead of being manipulated by some stakeholders.
- **Terrorism/crime anticipation.** It plays a significant role in improving public security and reducing potential loss caused by crimes or terrorism attacks. A more specific use case would be finding street gang members on Twitter. Many street gang members intimidate others through Twitter and sometimes even share recent illegal activities [153]. Therefore, tweets posted by these gang members are useful for predicting potential crimes and to prevent them in advance. In addition, the geospatial and demographic information of tweets can help trace

the criminal route and even locate the criminals in the end.

- **Mental disorder pre-diagnosis.** Recently, with the fast development of the society, more and more people suffer from mental disorders. Some patients with severe depression may even commit suicide which cause much pain and loss to their friends and families. Instead of talking to others, some of the patients are more willing to post their feelings on OSN. It is very meaningful to propose a framework which is able to detect suicide tendency using sentiment and semantic analysis based on the tweet posts. Early diagnosis of severe mental disorder help the families and doctors to take actions immediately to prevent potential tragedies.

Appendix A

Survey About Users' Opinions on Low-quality Content

This survey is designed for a research project related to the filtering of low-quality content. The research project is funded by Nanyang Technological University. Your participation in this survey is voluntary. The survey will take you less than 5 minutes to finish. You may refuse to take part in the research or exit the survey at any time without penalty. You are free to decline to answer any particular question you do not wish to answer for any reason. Your name and identity will not be revealed and will be totally anonymous. There are no foreseeable risks involved in participating in this study other than those encountered in day-to-day life. The survey mainly collects open information, and does not contain sensitive questions. The results of the project are open to the public and your responses may help us learn more about the low-quality content. The survey results may be published in conference and journal papers. If you have questions at any time about the study or the procedures, you may contact me via wchen015@e.ntu.edu.sg. Should you have any query on any ethics issue of this survey, please email IRB at irb@ntu.edu.sg with reference no. **IRB-2017-04-015**.

ELECTRONIC CONSENT: Please select your choice below. You may print a copy of this consent form for your records. Clicking on the Agree button indicates that

- You have read the above information
- You voluntarily agree to participate
- You agree to have the survey results being published in conference and journal papers.

1. Your gender

- Male.
- Female.

2. Your age

- Less than 18
- 18 to 25
- 26 to 35
- 36 to 45
- More than 45

3. How often do you use social network sites (e.g. Twitter, Facebook, Weibo, etc)? (Single choice)

- Nearly everyday.
- At least once a week.
- Less than once a week.

- Seldom or never.

4. How often will you clean up your followees/friends? (Single choice)

- Seldom or never.
- More than once a month.
- At least once a month.
- Almost every week.

5. If someone follows you, will you follow back? (Single choice)

- I usually follow back out of courtesy.
- I only follow those I know.
- I only follow those who share common interests with me.
- I usually don't follow back.

6. What will you regard as low-quality content when you are using social network sites? (Multiple choices)

- Those I'm not interested in.
- All advertisements.
- Advertisements posted by organizations who are not famous.
- Those generated automatically by some applications or services (Not updated by users).
- Meaningless messy codes.
- Deceptive content.

7. Please tick the boxes of those you regard as content polluters. (Multiple choices)

- Today stats: One follower, No unfollowers via (URL omitted)
- I've collected 7,715 gold coins! (URL omitted) #android, #androidgames, #gameinsight
- I posted a new photo to Facebook (URL omitted)
- Hot, my little pony friendship city light curtain .(hm118) - Full read by eBay (URL omitted)
- New Toshiba Encore 7 16GB Intel Atom WiFi tablet - Full read by eBay (URL omitted)
- Anupam Kher completes 31 years in Bollywood (URL omitted)
- 23 Clever Tattoos You Might Not Actually Regret In 50 Years (URL omitted)

8. How much do content polluters affect your user experience when using social network sites? (Single choice)

- Very much.
- A bit but still bearable.
- A little.
- They don't affect my user experience.

9. What's the maximum threshold (as a percentage of your recently received messages) you can bear before considering unfollowing him/her? (Single choice)

- I don't care too much about content polluters and it's too bothersome to unfollow others.
- Nearly 100%.

- More than 75%.
- More than 50%.
- More than 25%.

10. **Are you willing to use an extension/application to help filter content polluters on social network sites? (Single choice)**

- Yes.
- No.
- I'm not sure.

Appendix B

Examples of blacklist keywords

"weather", "updates", "theweatherchannel", "channel", "transponder", "snail", "although", "automatically", "follow", "followed", "libra", "unfollowed", "practical", "tug", "aries", "profess", "capricorn", "conflicting", "checked", "virgo", "embol", "maintaining", "pragma", "prowess", "readily", "gemini", "scorpio", "sides", "apparent", "capable", "strategic", "foresee", "imagination", "unfolding", "approach", "measurable", "taurus", "comprehend", "stellar", "aquarius", "enables", "highly", "pisces", "det", "sagittarius", "leo", "emotions", "financial", "somethin", "fully", "follows", "understanding", "calm", "closest", "planning", "witness", "clearly", "convince", "found", "begin", "creative", "matters", "followers", "huaraches", "presentation", "sex", "attitude", "earning", "seem", "gucci", "cancer", "gain", "giants", "benefits", "checkout", "giveaway", "challenge", "encounters", "custom", "monsters", "tos", "coins", "pips", "wild", "collected", "current", "mgwv", "harvested", "candid", "changes", "enter", "pill", "retweet", "straw", "null", "bikini", "smart", "stats", "upskirt", "blowjob", "masturbation", "unfollowers", "bayonet", "followtrick", "mbf", "camel", "limitless", "anal", "hats", "click", "followback", "teamfollowback", "unf", "vagina", "anotherfollowtrain", "positively", "eurusd", "flashiest", "horny", "lesbian",

"seems", "sexy", "adurabyhenshawblaze", "csgorumble", "fade", "mhmm", "safaree", "supporters", "allegedly", "amateur", "cock", "newborn", "samuels", "thumb", "alubarna", "facial", "healthier", "milf", "reflective", "useless", "wers", "badboy", "baths", "bbw", "busty", "decay", "loaner", "oral", "pussy", "sail"

Appendix C

Examples of stock sensitive sentiment dictionary

The following are some example words in the sentiment dictionary translated from Chinese to English.

Positive:

"China", "people", "up", "big", "origin", "company", "after", "ten thousand", "already", "new", "USA", "thousand million", "mutual", "option", "most", "market", "reporter", "under", "Beijing", "all", "family", "second", "high", "point", "10", "three", "national", "1", "happen", "economy", "even", "3", "years", "one", "more", "Li", "work", "5", "claim", "release", "survey", "before", "time", "report", "9", "among", "personnel", "enterprise", "days", "Chairman", "development", "network", "first", "only", "Shanghai", "Xi Jinping", "Japan", "recently", "news"

Negative:

"policy", "bank", "tourist", "today", "death", "accident", "president", "local", "influence", "Hong Kong", "no", "decrease", "again", "Russia", "reform", "manage-

ment", "department", "yet", "organization", "rest", "hospital", "relationship", "North Korea", "injured", "fixed assets", "South Korea", "expect", "late", "low", "explode", "accept", "conduct", "Trump", "Taiwan", "suspicious", "continue", "micro", "history", "response", "Europe", "hurt", "adjust", "season", "risk", "elderly", "further", "lead to", "investor", "stop", "earthquake"

Bibliography

- [1] D. Boyd and N. Ellison, “Social network sites: definition, history, and scholarship,” *IEEE Engineering Management Review*, vol. 38, no. 3, pp. 16–31, 2010.
- [2] P. Collin, K. Rahilly, I. Richardson, and A. Third, “The benefits of social networking services,” 2011.
- [3] Statista, “Leading social networks worldwide,” <http://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/>, 2017, accessed: 2018-01-01.
- [4] A. Lipsman, “Social networking goes global,” *Comscore. Jul*, vol. 31, p. 2007, 2007.
- [5] M. S. Gerber, “Predicting crime using twitter and kernel density estimation,” *Decision Support Systems*, vol. 61, pp. 115–125, 2014.
- [6] M. Al Boni and M. S. Gerber, “Predicting crime with routine activity patterns inferred from social media,” in *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*. IEEE, 2016, pp. 001 233–001 238.
- [7] Y. Fan, Y. Zhang, Y. Ye, X. li, and W. Zheng, “Social media for opioid addiction epidemiology,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management - CIKM17*. ACM Press, 2017.
- [8] S. Tsugawa, Y. Kikuchi, F. Kishino, K. Nakajima, Y. Itoh, and H. Ohsaki, “Recognizing depression from twitter activity,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 3187–3196.
- [9] P. Ambika and M. B. Rajan, “Survey on diverse facets and research issues in social media mining,” in *Research Advances in Integrated Navigation Systems (RAINS), International Conference on*. IEEE, 2016, pp. 1–6.

- [10] S. Asur and B. A. Huberman, "Predicting the future with social media," in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, vol. 1. IEEE, 2010, pp. 492–499.
- [11] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welp, "Election forecasts with twitter: How 140 characters reflect the political landscape," *Social science computer review*, vol. 29, no. 4, pp. 402–418, 2011.
- [12] I. R. Stanford Medicine, "Spam," <https://med.stanford.edu/irt/security/spam.html>, 2015, accessed: 2016-08-01.
- [13] E. Cashen, "Germany takes aim at fake news and illegal content with 50m fines," <https://www.theneweconomy.com/strategy/germany-takes-aim-at-fake-news-with-e50m-fines>, 2017, accessed: 2018-05-12.
- [14] U. Neal, "Almost 10 percent of twitter is spam," <http://www.fastcompany.com/3044485/almost-10-of-twitter-is-spam>, 2015, accessed: 2016-05-12.
- [15] J. Kalbitzer, T. Mell, F. BERPohl, M. A. Rapp, and A. Heinz, "Twitter psychosis: a rare variation or a distinct syndrome?" *The Journal of nervous and mental disease*, vol. 202, no. 8, p. 623, 2014.
- [16] C. Yang, R. Harkreader, and G. Gu, "Empirical evaluation and new design for fighting evolving twitter spammers," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1280–1293, 2013.
- [17] S. Lee and J. Kim, "Warningbird: A near real-time detection system for suspicious urls in twitter stream," *IEEE transactions on dependable and secure computing*, vol. 10, no. 3, pp. 183–195, 2013.
- [18] H. Fu, X. Xie, and Y. Rui, "Leveraging careful microblog users for spammer detection," in *Proceedings of the 24th International Conference on World Wide Web*. ACM, 2015, pp. 419–429.
- [19] A. J. Kimmel, *Rumors and rumor control: A Manager's Guide to Understanding and Combatting Rumors*. Taylor and Francis, 2004.
- [20] N. DiFonzo and P. Bordia, *Rumor psychology: Social and organizational approaches*. American Psychological Association, 2007.
- [21] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, "Aspects of rumor spreading on a microblog network," in *Social Informatics*. Springer, 2013, pp. 299–308.

- [22] J. Ito, J. Song, H. Toda, Y. Koike, and S. Oyama, "Assessment of tweet credibility with lda features," in *Proceedings of the 24th International Conference on World Wide Web*. ACM, 2015, pp. 953–958.
- [23] T. Kawabe, Y. Namihira, K. Suzuki, M. Nara, Y. Sakurai, S. Tsuruta, and R. Knauf, "Tweet credibility analysis evaluation by improving sentiment dictionary," in *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2015, pp. 2354–2361.
- [24] W. Chen, C. K. Yeo, C. T. Lau, and B. S. Lee, "Behavior deviation: An anomaly detection view of rumor preemption," in *Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2016 IEEE 7th Annual*. IEEE, 2016, pp. 1–7.
- [25] M. Mendoza, B. Poblete, and C. Castillo, "Twitter under crisis: can we trust what we rt?" in *Proceedings of the first workshop on social media analytics*. ACM, 2010, pp. 71–79.
- [26] P. Ozturk, H. Li, and Y. Sakamoto, "Combating rumor spread on social media: The effectiveness of refutation and warning," in *System Sciences (HICSS), 2015 48th Hawaii International Conference on*. IEEE, 2015, pp. 2406–2414.
- [27] J. Ma, W. Gao, Z. Wei, Y. Lu, and K.-F. Wong, "Detect rumors using time series of social context information on microblogging websites," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 2015, pp. 1751–1754.
- [28] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," in *Proceedings of IJCAI*, 2016.
- [29] E. F. Fama, L. Fisher, M. C. Jensen, and R. Roll, "The adjustment of stock prices to new information," *International economic review*, vol. 10, no. 1, pp. 1–21, 1969.
- [30] E. F. Fama, "Efficient capital markets: Ii," *The journal of finance*, vol. 46, no. 5, pp. 1575–1617, 1991.
- [31] M. M. Osborne, "Brownian motion in the stock market," *Operations research*, vol. 7, no. 2, pp. 145–173, 1959.
- [32] V. L. Smith, "Constructivist and ecological rationality in economics," *The American Economic Review*, vol. 93, no. 3, pp. 465–508, 2003.

- [33] E. Bikas, D. Jurevičienė, P. Dubinskas, and L. Novickytė, “Behavioural finance: The emergence and development trends,” *Procedia-social and behavioral sciences*, vol. 82, pp. 870–876, 2013.
- [34] J. R. Nofsinger, “Social mood and financial economics,” *The Journal of Behavioral Finance*, vol. 6, no. 3, pp. 144–160, 2005.
- [35] J. Fox and A. Sklar, *The myth of the rational market: A history of risk, reward, and delusion on Wall Street*. Harper Business New York, 2009.
- [36] J. Nocera, “Poking holes in a theory on markets,” *New York Times*, vol. 5, 2009.
- [37] A. W. Lo, “Reconciling efficient markets with behavioral finance: the adaptive markets hypothesis,” 2005.
- [38] A. Urquhart and R. Hudson, “Efficient or adaptive markets? evidence from major stock markets using very long run historic data,” *International Review of Financial Analysis*, vol. 28, pp. 130–142, 2013.
- [39] B. Qian and K. Rasheed, “Stock market prediction with multiple classifiers,” *Applied Intelligence*, vol. 26, no. 1, pp. 25–33, 2007.
- [40] S. Quayes and A. M. Jamal, “Impact of demographic change on stock prices,” *The Quarterly Review of Economics and Finance*, vol. 60, pp. 172–179, 2016.
- [41] X.-Q. Sun, H.-W. Shen, and X.-Q. Cheng, “Trading network predicts stock price,” *Scientific reports*, vol. 4, p. 3711, 2014.
- [42] A. K. Nassirtoussi, S. Aghabozorgi, T. Y. Wah, and D. C. L. Ngo, “Text mining for market prediction: A systematic review,” *Expert Systems with Applications*, vol. 41, no. 16, pp. 7653–7670, 2014.
- [43] S. G. Chowdhury, S. Routh, and S. Chakrabarti, “News analytics and sentiment analysis to predict stock price trends,” *International Journal of Computer Science and Information Technologies*, vol. 5, no. 3, pp. 3595–3604, 2014.
- [44] S. L. Heston and N. R. Sinha, “News versus sentiment: Comparing textual processing approaches for predicting stock returns,” *Robert H. Smith School Research Paper*, 2014.
- [45] M. Nofer and O. Hinz, “Using twitter to predict the stock market,” *Business & Information Systems Engineering*, vol. 57, no. 4, pp. 229–242, 2015.
- [46] J. Si, A. Mukherjee, B. Liu, Q. Li, H. Li, and X. Deng, “Exploiting topic based twitter sentiment for stock prediction.” *ACL (2)*, vol. 2013, pp. 24–29, 2013.

- [47] P. D. Azar and A. W. Lo, “The wisdom of twitter crowds: Predicting stock market reactions to fomc meetings via twitter feeds,” *The Journal of Portfolio Management*, vol. 42, no. 5, pp. 123–134, 2016.
- [48] E. Shearer, “News use across social media platforms 2017,” <http://www.journalism.org/2017/09/07/news-use-across-social-media-platforms-2017/>, accessed: 2017-11-03.
- [49] X. Zhang, H. Fuehres, and P. A. Gloor, “Predicting stock market indicators through twitter i hope it is not as bad as i fear,” *Procedia-Social and Behavioral Sciences*, vol. 26, pp. 55–62, 2011.
- [50] J. Garcke, T. Gerstner, and M. Griebel, “Intraday foreign exchange rate forecasting using sparse grids,” in *Sparse grids and applications*. Springer, 2012, pp. 81–105.
- [51] M. Chakraborty, S. Pal, R. Pramanik, and C. R. Chowdary, “Recent developments in social spam detection and combating techniques: A survey,” *Information Processing & Management*, 2016.
- [52] K. Thomas, C. Grier, D. Song, and V. Paxson, “Suspended accounts in retrospect: an analysis of twitter spam,” in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. ACM, 2011, pp. 243–258.
- [53] V. Sridharan, V. Shankar, and M. Gupta, “Twitter games: how successful spammers pick targets,” in *Proceedings of the 28th Annual Computer Security Applications Conference*. ACM, 2012, pp. 389–398.
- [54] T. Inc., “The twitter rules,” <https://support.twitter.com/articles/18311>, 2016, accessed: 2016-08-01.
- [55] B. Wang, A. Zubiaga, M. Liakata, and R. Procter, “Making the most of tweet-inherent features for social spam detection on twitter,” in *Proceedings of the 5th Workshop on Making Sense of Microposts*, vol. 1395, 2015, pp. 10–16.
- [56] E. Leong, “New ways to control your experience on twitter,” <https://blog.twitter.com/2016/new-ways-to-control-your-experience-on-twitter>, 2016, accessed: 2017-04-17.
- [57] K. Lee, B. D. Eoff, and J. Caverlee, “Seven months with the devils: A long-term study of content polluters on twitter.” in *ICWSM*, 2011.
- [58] M. Vergelis, T. Shcherbakova, and N. Demidova, “Kaspersky security bulletin. spam in 2014,” <https://securelist.com/kaspersky-security-bulletin-spam-in-2014/69225/>, 2014, accessed: 2017-12-18.

- [59] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. D. Spyropoulos, and P. Stamatopoulos, "Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach," *arXiv preprint cs/0009009*, 2000.
- [60] H. Drucker, D. Wu, and V. N. Vapnik, "Support vector machines for spam categorization," *IEEE Transactions on Neural networks*, vol. 10, no. 5, pp. 1048–1054, 1999.
- [61] C. O'Brien and C. Vogel, "Spam filters: bayes vs. chi-squared; letters vs. words," in *Proceedings of the 1st international symposium on Information and communication technologies*. Trinity College Dublin, 2003, pp. 291–296.
- [62] B. Leiba, J. Ossher, V. Rajan, R. Segal, and M. N. Wegman, "Smtip path analysis." in *CEAS*, 2005.
- [63] P. O. Boykin and V. P. Roychowdhury, "Leveraging social networks to fight spam," *Computer*, vol. 38, no. 4, pp. 61–68, 2005.
- [64] S. Hershkop and S. J. Stolfo, *Behavior-based email analysis with application to spam detection*. Columbia University, 2006.
- [65] E. Damiani, S. D. C. Di Vimercati, S. Paraboschi, and P. Samarati, "P2p-based collaborative spam detection and filtering," in *Peer-to-Peer Computing, 2004. Proceedings. Proceedings. Fourth International Conference on*. IEEE, 2004, pp. 176–183.
- [66] G. Mo, W. Zhao, H. Cao, and J. Dong, "Multi-agent interaction based collaborative p2p system for fighting spam," in *Proceedings of the IEEE/WIC/ACM international conference on Intelligent Agent Technology*. IEEE Computer Society, 2006, pp. 428–431.
- [67] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: a literature survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2091–2121, 2013.
- [68] S. Sheng, M. Holbrook, P. Kumaraguru, L. F. Cranor, and J. Downs, "Who falls for phish?: a demographic analysis of phishing susceptibility and effectiveness of interventions," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2010, pp. 373–382.
- [69] Y. Cao, W. Han, and Y. Le, "Anti-phishing based on automated individual whitelist," in *Proceedings of the 4th ACM workshop on Digital identity management*. ACM, 2008, pp. 51–60.
- [70] S. Sheng, B. Wardman, G. Warner, L. F. Cranor, J. Hong, and C. Zhang, "An empirical analysis of phishing blacklists," 2009.

- [71] Y. Joshi, S. Saklikar, D. Das, and S. Saha, “Phishguard: a browser plug-in for protection from phishing,” in *Internet Multimedia Services Architecture and Applications, 2008. IMSAA 2008. 2nd International Conference on*. IEEE, 2008, pp. 1–6.
- [72] D. L. Cook, V. K. Gurbani, and M. Daniluk, “Phishwish: a stateless phishing filter using minimal rules,” *Lecture Notes in Computer Science*, vol. 5143, pp. 182–186, 2008.
- [73] Y. Zhang, J. I. Hong, and L. F. Cranor, “Cantina: a content-based approach to detecting phishing web sites,” in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 639–648.
- [74] C. Whittaker, B. Ryner, and M. Nazif, “Large-scale automatic classification of phishing pages,” in *NDSS*, vol. 10, 2010, p. 2010.
- [75] P. Likarish, E. Jung, D. Dunbar, T. E. Hansen, and J. P. Hourcade, “B-apt: Bayesian anti-phishing toolbar,” in *Communications, 2008. ICC’08. IEEE International Conference on*. IEEE, 2008, pp. 1745–1749.
- [76] A. Stone, “Natural-language processing for intrusion detection,” *Computer*, vol. 40, no. 12, 2007.
- [77] A. Aggarwal, A. Rajadesingan, and P. Kumaraguru, “Phishari: Automatic realtime phishing detection on twitter,” in *eCrime Researchers Summit (eCrime), 2012*. IEEE, 2012, pp. 1–12.
- [78] Z. Miller, B. Dickinson, W. Deitrick, W. Hu, and A. H. Wang, “Twitter spammer detection using data stream clustering,” *Information Sciences*, vol. 260, pp. 64–73, 2014.
- [79] X. Hu, J. Tang, H. Gao, and H. Liu, “Social spammer detection with sentiment information,” in *2014 IEEE International Conference on Data Mining*. IEEE, 2014, pp. 180–189.
- [80] I. Santos, I. Minambres-Marcos, C. Laorden, P. Galán-García, A. Santamaría-Ibirika, and P. G. Bringas, “Twitter content-based spam filtering,” pp. 449–458, 2014.
- [81] H.-C. Yang and C.-H. Lee, “Detecting tag spams for social bookmarking websites using a text mining approach,” *International Journal of Information Technology & Decision Making*, vol. 13, no. 02, pp. 387–406, 2014.

- [82] 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*. ACM, 2010, pp. 27–37.
- [83] A. Almaatouq, A. Alabdulkareem, M. Nouh, E. Shmueli, M. Alsaleh, V. K. Singh, A. Alarifi, A. Alfari, and A. S. Pentland, “Twitter: who gets caught? observed trends in social micro-blogging spam,” in *Proceedings of the 2014 ACM conference on Web science*. ACM, 2014, pp. 33–41.
- [84] J. Song, S. Lee, and J. Kim, “Spam filtering in twitter using sender-receiver relationship,” in *International Workshop on Recent Advances in Intrusion Detection*. Springer, 2011, pp. 301–317.
- [85] E. Tan, L. Guo, S. Chen, X. Zhang, and Y. Zhao, “Spammer behavior analysis and detection in user generated content on social networks,” in *Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on*. IEEE, 2012, pp. 305–314.
- [86] G. Cai, H. Wu, and R. Lv, “Rumors detection in chinese via crowd responses,” in *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*. IEEE, 2014, pp. 912–917.
- [87] 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*, vol. 2, no. 3, pp. 99–108, 2015.
- [88] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, and R. Procter, “Detection and resolution of rumours in social media: A survey,” *arXiv preprint arXiv:1704.00656*, 2017.
- [89] G. W. Allport and L. Postman, “An analysis of rumor,” *Public Opinion Quarterly*, vol. 10, no. 4, pp. 501–517, 1946.
- [90] C. Heath, C. Bell, and E. Sternberg, “Emotional selection in memes: the case of urban legends,” *Journal of personality and social psychology*, vol. 81, no. 6, p. 1028, 2001.
- [91] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, “Prominent features of rumor propagation in online social media,” in *Data Mining (ICDM), 2013 IEEE 13th International Conference on*. IEEE, 2013, pp. 1103–1108.
- [92] S. Wang and T. Terano, “Detecting rumor patterns in streaming social media,” in *Big Data (Big Data), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2709–2715.

- [93] Y. Yang, K. Niu, and Z. He, "Exploiting the topology property of social network for rumor detection," in *Computer Science and Software Engineering (JCSSE), 2015 12th International Joint Conference on*. IEEE, 2015, pp. 41–46.
- [94] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *Proceedings of the 20th international conference on World wide web*. ACM, 2011, pp. 675–684.
- [95] Q. Zhang, S. Zhang, J. Dong, J. Xiong, and X. Cheng, "Automatic detection of rumor on social network," in *National CCF Conference on Natural Language Processing and Chinese Computing*. Springer, 2015, pp. 113–122.
- [96] Z. Yang, C. Wang, F. Zhang, Y. Zhang, and H. Zhang, "Emerging rumor identification for social media with hot topic detection," in *2015 12th Web Information System and Application Conference (WISA)*. IEEE, 2015, pp. 53–58.
- [97] X. Liu, A. Nourbakhsh, Q. Li, R. Fang, and S. Shah, "Real-time rumor debunking on twitter," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 2015, pp. 1867–1870.
- [98] A. Friggeri, L. A. Adamic, D. Eckles, and J. Cheng, "Rumor cascades." in *ICWSM*, 2014.
- [99] A. Zubiaga, M. Liakata, R. Procter, K. Bontcheva, and P. Tolmie, "Crowdsourcing the annotation of rumourous conversations in social media," in *Proceedings of the 24th International Conference on World Wide Web*. ACM, 2015, pp. 347–353.
- [100] F. Yang, Y. Liu, X. Yu, and M. Yang, "Automatic detection of rumor on sina weibo," in *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*. ACM, 2012, p. 13.
- [101] S. Sun, H. Liu, J. He, and X. Du, "Detecting event rumors on sina weibo automatically," in *Asia-Pacific Web Conference*. Springer, 2013, pp. 120–131.
- [102] S. Vosoughi, "Automatic detection and verification of rumors on twitter," Ph.D. dissertation, Massachusetts Institute of Technology, 2015.
- [103] S. Kwon and M. Cha, "Modeling bursty temporal pattern of rumors." in *ICWSM*, 2014.
- [104] Z. Miller, B. Dickinson, W. Deitrick, W. Hu, and A. H. Wang, "Twitter spammer detection using data stream clustering," *Information Sciences*, vol. 260, pp. 64–73, 2014.

- [105] J. Guzman and B. Poblete, “On-line relevant anomaly detection in the twitter stream: an efficient bursty keyword detection model,” in *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*. ACM, 2013, pp. 31–39.
- [106] Y. Zhang, W. Chen, C. K. Yeo, C. T. Lau, and B. S. Lee, “A distance-based outlier detection method for rumor detection exploiting user behavioral differences,” in *Data and Software Engineering (ICoDSE), 2016 International Conference on*. IEEE, 2016, pp. 1–6.
- [107] V. Potì and A. Siddique, “What drives currency predictability?” *Journal of International Money and Finance*, vol. 36, pp. 86–106, 2013.
- [108] H. Yu, G. V. Nartea, C. Gan, and L. J. Yao, “Predictive ability and profitability of simple technical trading rules: Recent evidence from southeast asian stock markets,” *International Review of Economics & Finance*, vol. 25, pp. 356–371, 2013.
- [109] R. Choudhry and K. Garg, “A hybrid machine learning system for stock market forecasting,” *World Academy of Science, Engineering and Technology*, vol. 39, no. 3, pp. 315–318, 2008.
- [110] X. Ding, Y. Zhang, T. Liu, and J. Duan, “Deep learning for event-driven stock prediction.” in *IJCAI*, 2015, pp. 2327–2333.
- [111] A. Chatrath, H. Miao, S. Ramchander, and S. Villupuram, “Currency jumps, co-jumps and the role of macro news,” *Journal of International Money and Finance*, vol. 40, pp. 42–62, 2014.
- [112] M. Nofer and O. Hinz, “Are crowds on the internet wiser than experts? the case of a stock prediction community,” *Journal of Business Economics*, vol. 84, no. 3, pp. 303–338, 2014.
- [113] Y. Liu, Z. Qin, P. Li, and T. Wan, “Stock volatility prediction using recurrent neural networks with sentiment analysis,” *arXiv preprint arXiv:1705.02447*, 2017.
- [114] J. Si, A. Mukherjee, B. Liu, S. J. Pan, Q. Li, and H. Li, “Exploiting social relations and sentiment for stock prediction.” in *EMNLP*, vol. 14, 2014, pp. 1139–1145.
- [115] A. N. Kercheval and Y. Zhang, “Modelling high-frequency limit order book dynamics with support vector machines,” *Quantitative Finance*, vol. 15, no. 8, pp. 1315–1329, 2015.

- [116] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock market index using fusion of machine learning techniques," *Expert Systems with Applications*, vol. 42, no. 4, pp. 2162–2172, 2015.
- [117] N. Chapados and Y. Bengio, "Cost functions and model combination for var-based asset allocation using neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 890–906, 2001.
- [118] G. P. Zhang, "A neural network ensemble method with jittered training data for time series forecasting," *Information Sciences*, vol. 177, no. 23, pp. 5329–5346, 2007.
- [119] Y.-K. Kwon and B.-R. Moon, "A hybrid neurogenetic approach for stock forecasting," *IEEE Transactions on Neural Networks*, vol. 18, no. 3, pp. 851–864, 2007.
- [120] A. M. Rather, A. Agarwal, and V. Sastry, "Recurrent neural network and a hybrid model for prediction of stock returns," *Expert Systems with Applications*, vol. 42, no. 6, pp. 3234–3241, 2015.
- [121] Y. Peng and H. Jiang, "Leverage financial news to predict stock price movements using word embeddings and deep neural networks," in *Proceedings of NAACL-HLT*, 2016, pp. 374–379.
- [122] S. V. Stehman, "Selecting and interpreting measures of thematic classification accuracy," *Remote sensing of Environment*, vol. 62, no. 1, pp. 77–89, 1997.
- [123] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.
- [124] S. Ghosh, B. Viswanath, F. Kooti, N. K. Sharma, G. Korlam, F. Benevenuto, N. Ganguly, and K. P. Gummadi, "Understanding and combating link farming in the twitter social network," in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012, pp. 61–70.
- [125] L. Jin, Y. Chen, T. Wang, P. Hui, and A. V. Vasilakos, "Understanding user behavior in online social networks: A survey," *IEEE Communications Magazine*, vol. 51, no. 9, pp. 144–150, 2013.
- [126] C. Yang, R. Harkreader, J. Zhang, S. Shin, and G. Gu, "Analyzing spammers' social networks for fun and profit: a case study of cyber criminal ecosystem on twitter," in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012, pp. 71–80.

- [127] W. Chen, C. K. Yeo, C. T. Lau, and B. S. Lee, "Real-time twitter content polluter detection based on direct features," in *Information Science and Security (ICISS), 2015 2nd International Conference on*. IEEE, 2015, pp. 1–4.
- [128] X. Zheng, X. Zhang, Y. Yu, T. Kechadi, and C. Rong, "Elm-based spammer detection in social networks," *The Journal of Supercomputing*, pp. 1–15, 2015.
- [129] S. Fakhraei, J. Foulds, M. Shashanka, and L. Getoor, "Collective spammer detection in evolving multi-relational social networks," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 1769–1778.
- [130] M. Sahami, "Learning limited dependence bayesian classifiers." in *KDD*, vol. 96, 1996, pp. 335–338.
- [131] I. Androutsopoulos, J. Koutsias, K. V. Chandrinou, and C. D. Spyropoulos, "An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages," pp. 160–167, 2000.
- [132] H. Drucker, D. Wu, and V. N. Vapnik, "Support vector machines for spam categorization," *IEEE Transactions on Neural networks*, vol. 10, no. 5, pp. 1048–1054, 1999.
- [133] A. S. I. Ingo Feinerer, Kurt Hornik, "A framework for text mining applications within r," <https://cran.r-project.org/web/packages/tm/index.html>, 2015, accessed: 2015-12-04.
- [134] B. J. Park and J. S. Han, "Efficient decision support for detecting content polluters on social networks: an approach based on automatic knowledge acquisition from behavioral patterns," *Information Technology and Management*, vol. 17, no. 1, pp. 95–105, 2016.
- [135] M. Takayasu, K. Sato, Y. Sano, K. Yamada, W. Miura, and H. Takayasu, "Rumor diffusion and convergence during the 3.11 earthquake: a twitter case study," *PLoS one*, vol. 10, no. 4, p. e0121443, 2015.
- [136] N. DiFonzo, M. J. Bourgeois, J. Suls, C. Homan, N. Stupak, B. P. Brooks, D. S. Ross, and P. Bordia, "Rumor clustering, consensus, and polarization: Dynamic social impact and self-organization of hearsay," *Journal of Experimental Social Psychology*, vol. 49, no. 3, pp. 378–399, 2013.
- [137] C. Silverman, "Lies, damn lies, and viral content. how news websites spread (and debunk) online rumors, unverified claims, and misinformation," *Tow Center for Digital Journalism*, vol. 168, 2015.

- [138] 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.
- [139] M. Koetse, “A short introduction to sina weibo: Background and status quo,” <http://www.whatsonweibo.com/sinaweibo/>, accessed: 2015-04-06.
- [140] Z. C. Lipton, J. Berkowitz, and C. Elkan, “A critical review of recurrent neural networks for sequence learning,” *arXiv preprint arXiv:1506.00019*, 2015.
- [141] S. Finn, P. T. Metaxas, and E. Mustafaraj, “Investigating rumor propagation with twitter trails,” *arXiv preprint arXiv:1411.3550*, 2014.
- [142] Y. Matsubara, Y. Sakurai, B. A. Prakash, L. Li, and C. Faloutsos, “Rise and fall patterns of information diffusion: model and implications,” in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 6–14.
- [143] D. E. Rumelhart and P. M. Todd, “Learning and connectionist representations,” *Attention and performance XIV: Synergies in experimental psychology, artificial intelligence, and cognitive neuroscience*, pp. 3–30, 1993.
- [144] M. Sakurada and T. Yairi, “Anomaly detection using autoencoders with nonlinear dimensionality reduction,” in *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*. ACM, 2014, p. 4.
- [145] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.
- [146] Y. Zhang, W. Chen, C. K. Yeo, C. T. Lau, and B. S. Lee, “Detecting rumors on online social networks using multi-layer autoencoder,” in *Technology & Engineering Management Conference (TEMSCON), 2017 IEEE*. IEEE, 2017, pp. 437–441.
- [147] P. Bajpai, “The world’s top 10 economies,” <http://www.investopedia.com/articles/investing/022415/worlds-top-10-economies.asp>, accessed: 2017-06-14.
- [148] W. Chen, Y. Zhang, C. K. Yeo, C. T. Lau, and B. S. Lee, “Stock market prediction using neural network through news on online social networks,” in *Smart Cities Conference (ISC2), 2017 International*. IEEE, 2017, pp. 1–6.
- [149] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

-
- [150] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.
 - [151] H. Drucker, “Improving regressors using boosting techniques,” in *ICML*, vol. 97, 1997, pp. 107–115.
 - [152] Y. E. Cakra and B. D. Trisedya, “Stock price prediction using linear regression based on sentiment analysis,” in *Advanced Computer Science and Information Systems (ICACSIS), 2015 International Conference on*. IEEE, 2015, pp. 147–154.
 - [153] L. Balasuriya, S. Wijeratne, D. Doran, and A. Sheth, “Finding street gang members on twitter,” in *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*. IEEE, 2016, pp. 685–692.

Author's Publications

Journals

- **Weiling Chen** ,Yan Zhang, Chai Kiat Yeo, Chiew Tong Lau, and Bu Sung Lee, “Leveraging social media news to predict stock index movement using RNN-Boost.”, submitted to *Data & Knowledge Engineering*.
- **Weiling Chen** ,Yan Zhang, Chai Kiat Yeo, Chiew Tong Lau, and Bu Sung Lee, “Unsupervised rumor detection based on users behaviors using neural networks.”, *Pattern Recognition Letters* (2017).
- **Weiling Chen** ,Chai Kiat Yeo, Chiew Tong Lau, and Bu Sung Lee, “A study on real-time low-quality content detection on Twitter from the users perspective.”, *PloS One*, no. 8 (2017): e0182487.

Conferences

- Julien Leblay, **Weiling Chen**, Steven Lynden, “Exploring the veracity of online claims with BackDrop.”, in *Conference on Information and Knowledge Management (CIKM)*, 26th ACM International, ACM, 2017.
- **Weiling Chen**, Yan Zhang, Chai Kiat Yeo, Chiew Tong Lau, and Bu Sung Lee, “Stock market prediction using neural network through news on online social networks.”, in *Smart Cities Conference (ISC2)*, 2017 International, pp. 1-6. IEEE, 2017.
- Yan Zhang, **Weiling Chen**, Chai Kiat Yeo, Chiew Tong Lau, and Bu Sung Lee, “Detecting rumors on Online Social Networks using multi-layer autoencoder.”, in *Technology & Engineering Management Conference (TEMSCON)*, 2017 IEEE, pp. 437-441. IEEE, 2017.
- Yan Zhang, **Weiling Chen**, Chai Kiat Yeo, Chiew Tong Lau, and Bu Sung Lee, “A distance-based outlier detection method for rumor detection exploiting user behaviorial differences.”, in *Data and Software Engineering (ICoDSE)*, 2016 International Conference on, pp. 1-6. IEEE, 2016.
- **Weiling Chen**, Chai Kiat Yeo, Chiew Tong Lau, and Bu Sung Lee, “Behavior deviation: An anomaly detection view of rumor preemption.”, in *Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 2016 IEEE 7th Annual, (pp. 1-7). IEEE. Las Vegas, Nevada, pp. 233-238, Jan 2015.
- **Weiling Chen**, Chai Kiat Yeo, Chiew Tong Lau, and Bu Sung Lee, “Real-Time Twitter Content Polluter Detection Based on Direct Features.”, in *Information Science and Security (ICISS)*, 2015 2nd International Conference on, pp. 1-4. IEEE, 2015.