Novel Text Mining Methods and Applications

A thesis submitted to University of Hyderabad in partial fulfillment

for the degree of

Doctor of Philosophy

by

BANGARI SHRAVAN KUMAR

Reg.No. 12MCPC07

SCHOOL OF COMPUTER AND INFORMATION SCIENCES

UNIVERSITY OF HYDERABAD

HYDERABAD -500046

Telangana

India

July, 2018

CERTIFICATE

This is to certify that the thesis entitled "**Novel Text Mining Methods and Applications**" submitted by **Bangari Shravan Kumar** bearing registration number **12MCPC07** in partial fulfillment of the requirements for award of Doctor of Philosophy in the School of Computer & Information Sciences is a bonafide work carried out by him under my supervision and guidance at IDRBT, Hyderabad.

This thesis is free from plagiarism and has not been submitted previously in part or in full to this or any other University or Institution for award of any degree or diploma.

Parts of this thesis have been published online in following publications:

1. Proceedings of SEMCCO-2014 (Chapter 3), ICAIECES-2016 (Chapter 4) .
2. Proceedings of FICTA-2016 (Chapter 5), COMPUTE-2017 (Chapter 6).
3. Knowledge-Based Systems-2016 (Chapter 2).

Further, the student has passed the following courses towards fulfillment of coursework requirement for Ph.D:

	Course Code	Name	Credits	Pass/Fail
1	BT701	Data Structures and Algorithms	4	Pass
2	BT702	Operating System and Programming	4	Pass
3	BT712	Machine Learning and Soft Computing	4	Pass
4	BT717	Advanced Data Mining	4	Pass

Supervisor	Director	Dean
Prof. Vadlamani Ravi	Dr. A.S. Ramasastri	Prof. K. Narayana Murthy
Center of Excellence in	IDRBT	School of Computer and
Analytics, IDRBT	Hyderabad-500 057, India	Information Sciences, UOH
Hyderabad-500 057, India		Hyderabad-500 046, India

DECLARATION

I, **Bangari Shravan Kumar**, hereby declare that this thesis entitled **"Novel Text Mining Methods and Applications"** submitted by me under the guidance and supervision of **Prof. Vadlamani Ravi**, is a bonafide research work and is free from plagiarism. I also declare that it has not been submitted previously in part or in full to this University or any other University or Institution for the award of any degree or diploma. I hereby agree that my thesis can be deposited in Shodganga/INFLIBNET.

A report on plagiarism statistics from the University Librarian is enclosed.

Signature of the Student

Date:

(B. Shravan Kumar)

Reg. No.: 12MCPC07

//Countersigned//

Signature of the Supervisor:

Dedicated to My Family for their support and encouragement

Acknowledgements

First and above of all, I praise God Almighty. I am very much grateful to him for the gift of everlasting life and the energy to work hard towards my goal.

This Ph.D. work and dissertation would have been impossible without the help and support of my supervisor **Prof. Vadlamani Ravi**, Professor, IDRBT, Hyderabad. I thank him for having faith in me and my work. For his precious foundational & advanced lectures of text mining and related topics, grammar corrections for my papers and timely support. He guided me through every step of my research including problem definition and solution to the problem. I am grateful to him for encouraging and helping me in developing as a researcher.

Later, I would like to thank my parents **Sri. Bangari Vaikuntam** and **Smt. Bangari Vijayalakshmi** for their best support and encouragement all over in my life. They have given me the freedom of choice for what you like you can do and always granted the privilege to my life. Because of their essence only I came to Doctoral studies. I would also like to thank my younger brother (**Anil**), younger sister (**Ramya**) for their uncountable love. I express my sincere gratitude my family members standing by me at all critical junctures of my life so far and to be a constant source of inspiration and motivation.

I specially acknowledge **University of Hyderabad** (UoH) for its academic support. I also extend my acknowledgments to **Institute for Development and Research in Banking Technology** (IDRBT), a Research and Development (R&D) center established by Resrve Bank of India (RBI), for its financial support and beautiful environment for conducting the research. I express my sincere thanks

to **Centre of Excellence in Analytics** (CoEA), IDRBT for making me part of experimenting with datasets of various Banks and presenting the results to bankers throughout Ph.D.

It is my privilege to thank **Prof. K. Narayana Murthy**, Dean, School of Computer and Information Sciences (SCIS), UoH, Hyderabad for his academic support. I also extend my thanks to **Prof. Arun Agarwal**, Pro-Vice-Chancellor, UoH, Hyderabad. It is an honor for me to thank **Dr. A.S. Ramasastri**, Director, IDRBT for reviewing the work and timely suggestions throughout my research. I also thank **Mr. B. Sambamurthy**, Former Director, IDRBT for extending his cooperation at the preliminary stage of my research.

Also, I would whole-heartedly thank my Doctoral Research Committee members **Dr. G. R. Gangadharan**, Associate Professor, IDRBT, **Prof. B.M. Mehtre**, Professor, IDRBT, and Late Dr. Mahil Carr, Assistant Professor, IDRBT for their extreme support and timely appreciations.

I also take a minute to thank **Prof. B.L. Deekshatulu**, **Prof. D.K. Subramanyam**, **Prof. B. Yegnanarayana**, **Prof. S. Ramachandram**, and **Prof. Venu Govindaraju** for their periodical reviews of my work and their inputs. I would also like to express my sincere thanks and gratitude to the faculty including Prof. V.N. Sastry, Dr. M.V.N.K. Prasad, Dr. V. Radha, Dr. Rajarshi Pal, Dr. Shakthi Mishra, Dr. S. Nagesh Battu and Mr. Sandilya for their lectures as part of my coursework and timely suggestions. I also thank to Mr. Ratna Kumar, Mr. Prakash Dhavale, and Mr. Srinivas for their support in utilizing library resources to conduct research. I also thank to Administarive Assistants Pavani, and Namdev for their support in Academic activities. I also special thanks to Pradeep, Ms. Girija, and Rohit for help in grammar correction.

It is always pleasure to express my sincere thanks to the research colleagues including Pradeep, Ghanashyam, Srinu, Veeru, Gopal, Anup, Tiwari, Ilaiah, J. Chandrashekar, Hiran, K. Ravi, Sriramulu, Sandhya,

Manu, Jaya Krishna, Kamaruddin, U. Ravi, Dinesh, Sitara, Satyakrishna, Avinash, Mahesh, Mallikarjun, Johnpaul, Praveen, T. Chandrashekar, Jayarao, Manoj, and Suresh for their interactions with me and their support.

Throughout my doctoral study, many research associates of IDRBT accompanied me. I would like to express my special thanks to my lunch team as well as tour batch Siddeshwar, Tejasviram, Shiva Krishna, Sai Kiran, Venkata Krishna, Narasimharao, Rakesh, Murali, Gopi, Murthy. I thank K. Rakesh for his help in University works. I also extend my sincere thanks to summer project intern namely Rishabh who contributed in one of my works and Nishanth.

I thank UoH M.Tech students Pradeep, Ganesh, Krishna, Mayank, Kshitij, Shukla, Satish, Kansama, Vikash, Bhaskar, and Rohit who worked along with me in CoEA. I also thank, Dr. K. Ramachandramurthy for his discussions and suggestions.

I also specially thank my hostel members Pradeep, Ghanshyam, Gopal, Anup, Tiwari, and Avinash who not only were great critics of my work but also made my stay at IDRBT a pleasant and memorable experience.

I also extend my thanks to my M.Tech friends Ravindra Prasad, Shivareddy, Harish, Vishnu, Venkat, Ajay, Sridhar, Madhavi, Vinaya, and Satheesh. My Engineering friends Bharat, Gangadhar, Alikhan, Imranuddin, Shiva, Ramu, Satyakiran, Sridevi, Padma, and Rajakumar. My school friends Dr. Kumaraswamy, V. Sridhar, Saleem, E. Kumar, Dr. Shafiuddin, Thirupathi, K. Shekar, Santhosh, D. Sridhar, K. Satish, Khaleel, A. Satish, B. Srinivas, Chandramouli, T. Srinivas, Shiva, D. Srinivas, Kiran, and O. Srinivas. I also extend my thanks to intermediate friends Ashok, Ch. Sridhar, Dr. Santhosh Singh, Karunakar, and Ravinder.

I also extend my thanks to OU faculty Prof. S. Srinivasa Rao, Prof. S. Sameen Fatima, Prof. P. Premchand, Prof. M. Venkatdass, Prof.

A. Venugopal Reddy, Prof. Lakshmi Rajamani, Dr. K. Shyamala, Prof. Rambabu, Prof. Suresh, and Dr. P.V. Sudha for their support in my Master's program.

I also extend my thanks to my school teachers Satyavathi, Sharma, Sudha Bindu, Late. Krupadanam, Ramulu, Late. Maruthi, Lokabiram, Vijayalaksmi, Damodhar Reddy, Subramanyam, Jaganmohan Rao, A. Satyanarayana, Majid, and J. Satyanarayana who shaped my childhood. I especially, thank to my tuition teachers Dr. Venu, Dr. Prabasini, Rajender, and Rangachary.

I also extend my thanks to college teachers Mr. M. Satyam Babu, Dr. Raghotham Reddy, Somashekar Reddy, Prof. Krishna Reddy, and Krishnaja.

B. Shravan Kumar

Abstract

In the real world most of the data is generated in unstructured format, for example, text, audio, images and video. Nearly 80% of the data is in the unstructured format. Rather than structured databases like RDBMS etc., we require sophisticated techniques to process the unstructured data. Text mining involves transforming unstructured text data into a structured format, thereby making it amenable for knowledge extraction (data mining) and decision making thereof. The fundamental tasks of text mining are [1] [2] as follows: Classification, Clustering, Association Rule Mining, Topic detection and Summarization. Once the text is converted into a structured format common data mining tasks are performed. Text mining has a lot of applications including, document classification, text clustering, social network analysis, plagiarism detection, phishing/ malware/ spam detection, financial market prediction, financial market movement (directional i.e. up or down), plagiarism detection and Customer Relationship Management (CRM).

Document classification is one of the challenging problems in the real world. It consists of various applications like Phishing detection, Malware detection, and Spam detection. Discriminative features play an important role in document classification. In this thesis, we proposed novel hybrid models viz., OCSVM-LSI, PCA-OCSVM, LDA-CARM for document classification which are applicable for binary class as well one-class problems. In the process, we applied various feature selection methods to remove the irrelevant features.

Number of samples in the negative class are more compared to that of the positive class. This motivated us to build a one-class classification

model for textual data that is built based on the samples of one-class, usually negative class and test it on positive class, which is the minority class. Through this approach, we generated a model with less number of document comparisons for classifying the text documents.

Stock market prediction is an interesting problem which involves data mining subsuming applied statistics. Today‘s biggest economies of the world have higher stock market value. Human intervention is limited in stock market prediction. Researchers have been working to predict the stock exchange prices. No methodology has been conceived to predict the price movement accurately. It is therefore difficult to predict the stock market price dynamically. The stock market behaviour depends on the news which is in the unstructured format. This extracted knowledge from the unstructured data is used for effective decision making. The challenging task in stock market prediction is the NEWS collection and its preprocessing i.e. NEWS articles not appeared daily for the particular stock; NEWS appeared on that day but, the corresponding stock may not seem. For this scenario we proposed a novel model for predicting the stock prices. We employed various regression techniques viz., GMDH, GRNN, MLP, RPART, SVR, RF, QRRF.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter presents an introduction to text mining, associated tasks and its applications in various fields. Later, this chapter presents the research motivation, various problems encountered in text mining, followed by problem definition and the contributions made by this thesis.

1.1 Text Mining

In the real world, a majority of the data generated is in an unstructured format. Nearly 80% of the data is in this format. Unlike structured data, sophisticated techniques are required to process the unstructured data. Unstructured data mining deals with image, audio, video and text analysis, whereas text mining deals with text only. We can say that text mining is a subset of unstructured data mining. Text mining facilitates the transformation of unstructured text data into structured data after which standard data mining tasks can be performed on it. The fundamental tasks of text mining [4] are as follows: Classification, Clustering, Association Rule Mining, Topic detection, Summarization, and Document comparison (for plagiarism detection). Text mining has a lot of applications including, document classification, text clustering, social network analysis, plagiarism detection, phishing/ malware/ spam detection, financial market prediction, financial market movement prediction (directional i.e. up or down), and Customer Relationship Management (CRM).

Text mining is an interdisciplinary field [5] which involves Statistics, Machine Learning (ML), Databases, Information Retrieval Systems (IRS), Data Mining (DM), Natural Language Processing (NLP), Computational Linguistics, and Library and Information Sciences. How it differs from data mining? Data mining can be defined as the discovery and extraction of previously unknown knowledge from structured data. Text mining, on the other hand, deals with unstructured, heterogeneous text data. The data handled by text mining typically has a large number of features, hence making feature subset selection a necessary task in text mining. Text mining involves the following challenges:

1. Feature extraction/ generation from text corpus.
2. Handling the large number of features yielded by text mining [6].
3. Identifying the discriminative features is a challenging problem. i.e., an appropriate feature selection method needs to be adopted.
4. In addition to feature subset selection methods, classification performance also depends on the classifiers [7].

1.2 Problem Statement

Document classification is one of the challenging real-world problems. It has various applications like Language identification, News filtering, Genre determination, Sentiment analysis, Phishing detection, Malware detection, and Spam detection. It is intended to formulate the document classification problem as a binary classification task as well as a One-class classification task. The objective of one-class classification is to train a model with one class (negative class) samples and test it with other class (positive class) samples. Through this approach, even with a less number of sample documents available for training, a model can be built for classifying text documents. In binary document classification, unlike the one-class classification approach, the model is trained using samples from both the classes and the model learns the characteristics of each class. Discriminating features play a significant role in the text document classification problem.

Document clustering is one of the primary tasks of text mining. Clustering is an unsupervised learning technique, where class labels of the samples are not known. Based on the intrinsic grouping in the data, clustering algorithms divide the given samples into clusters. This problem was approached by performing feature selection with some unsupervised methods, followed by invoking clustering algorithms.

Stock market prediction is an interesting problem, which is usually solved by data mining. The biggest economies of the world today, have higher stock market index value. Researchers have been working to predict the stock exchange prices. However, no methodology has been proposed to predict the price movement accurately. It is also difficult to predict the stock market price dynamically. The stock market behavior among other things depends on financial news, which is in an unstructured format. This knowledge extracted from unstructured data is used for effective decision making. In this context, a challenging task in stock market prediction is financial news collection and its preprocessing. Some difficulties/challenges include (i) Financial news articles not appearing daily for a particular stock; and (ii) Financial news appears on a day but, the corresponding stock value may not be found. It is intended to address this problem too in this work.

1.3 Motivation

1.3.1 Problems Found in Text Mining

A literature survey of text mining uncovered some missing aspects, which are highlighted in this section. These aspects form the motivation for the current study.

1. Choosing an effective feature selection method is a challenging problem and requires detailed study.
2. Effective One-Class text document classification methods.
3. Class Association Rule Mining on text documents using LDA (Topic Modeling) as a preprocessor.

4. Stock market prediction with News articles using psycholinguistic features has not been explored.

1.3.2 Contributions of Thesis related to Problem Statement

Contributions of this thesis aredepicted in the Figure 1.1 Following are the contributions of this thesis with respect to the problem statement.

1. Text mining has a variety of applications. One of the applications is in the financial domain. An overall review of the state-of-the-art techniques of text mining in financial and other applications are categorized with respect to their tasks: Regression (FOREX rate prediction, and Stock Market prediction), Classification (Cyber Security, and Customer Relationship Management (CRM)), Text Summarization. Each of these applications is discussed separately (see Chapter 2).

2. A novel binary document classification model using data mining techniques is proposed. In this work, various feature subset selection methods were explored for finding discriminative features. Later, SVM, DT, NB, k-NN, MLP, GRNN, GMDH, and RIPPER were used for classification (see Chapter 3).

3. A novel one-class classification approach for text document classification using One-Class Support Vector Machine (OCSVM) and Latent Semantic Indexing (LSI) is proposed. In this study, first, a document-term matrix was formed from a corpus of text documents. This was followed by feature selection using the t-statistic method. Then, OCSVM was invoked on the dataset corresponding to the negative class and Support Vectors (SV) were extracted. Then, in the test phase, LSI was employed on the query documents from the positive class to compare them with the SVs extracted from the negative class and then the match score was computed using the cosine similarity measure. Then, based on a pre-specified threshold for the match score, positive category of the text corpus was classified (see Chapter 4).

4. A new hybrid model for document classification involving Principal Component Analysis (PCA) and One-Class Support Vector Machine (OCSVM) in tandem is proposed, where PCA helps achieve dimensionality reduction and OCSVM performs classification. Initially, PCA was invoked on the document-term matrix and then the top few principal components were selected. Later, OCSVM was trained on the records of the matrix corresponding to the negative class. Then, the trained OCSVM was tested with the records of the matrix corresponding to the positive class (see Chapter 5).

5. A new model for text document clustering is proposed. Text documents consist of a largenumber of features. This work selects the features through term variance, document frequency, Latent Dirichlet Allocation (LDA), and Significance methods followed by clustering with k-Means, k-Medoids, Self Organizing Maps (SOM), and Fuzzy C-means algorithms. (see Chapter 6).

6. A novel, general-purpose hybrid method comprising Topic Modeling and Class Association Rule Mining (CARM) in tandem for text classification is proposed. While topic modeling performed dimension reduction, association rule mining aspect was taken care of by Apriori and FP-growth algorithms separately (see Chapter 7).

7. Finally, hybrid intelligent models are proposed for stock market prediction using the psycholinguistic variables extracted from news articles and used as predictor variables. For prediction purpose, various intelligent techniques such as Multilayer Perceptron (MLP), Group Method of Data Handling (GMDH), General Regression Neural Network (GRNN), Random Forest (RF), Quantile Regression Random Forest (QRRF), Classification and Regression Tree (CART) and Support Vector Regression (SVR) were employed and their performance was compared on 12 datasets (see Chapter 8).

1.4 Structure of the Thesis

The thesis structure is as follows.

Abstract

Chapter 1 provides the motivation and the necessary background for the work reported in this thesis. The chapter begins with the definition of text mining and the challenges faced in the field. Then, the problem statement and the contributions of the thesis are described.

Chapter 2 reviews the works related to various data mining and machine learning techniques for text mining and its applications.

Chapter 3 presents binary document classification with data mining techniques.

Chapter 4 presents a novel one-class classification approach for text document classification using One-Class Support Vector Machine (OCSVM) and Latent Semantic Indexing (LSI).

Chapter 5 presents a novel document classification approach based on Principal Component Analysis (PCA) and One-Class Support Vector Machine (OCSVM).

Chapter 6 presents a model for document clustering using unsupervised feature subset selection methods.

Chapter 7 presents a hybrid model comprising Topic Modeling and Class Association Rule Mining (CARM) for text classification.

Chapter 8 presents hybrid intelligent models for stock market prediction using the psycholinguistic variables extracted from news articles as predictor variables.

Chapter 9 presents the conclusions and future directions drawn from this study.

APPENDIX A Overview of Techniques Used

APPENDIX B Annexure (Publications Online)

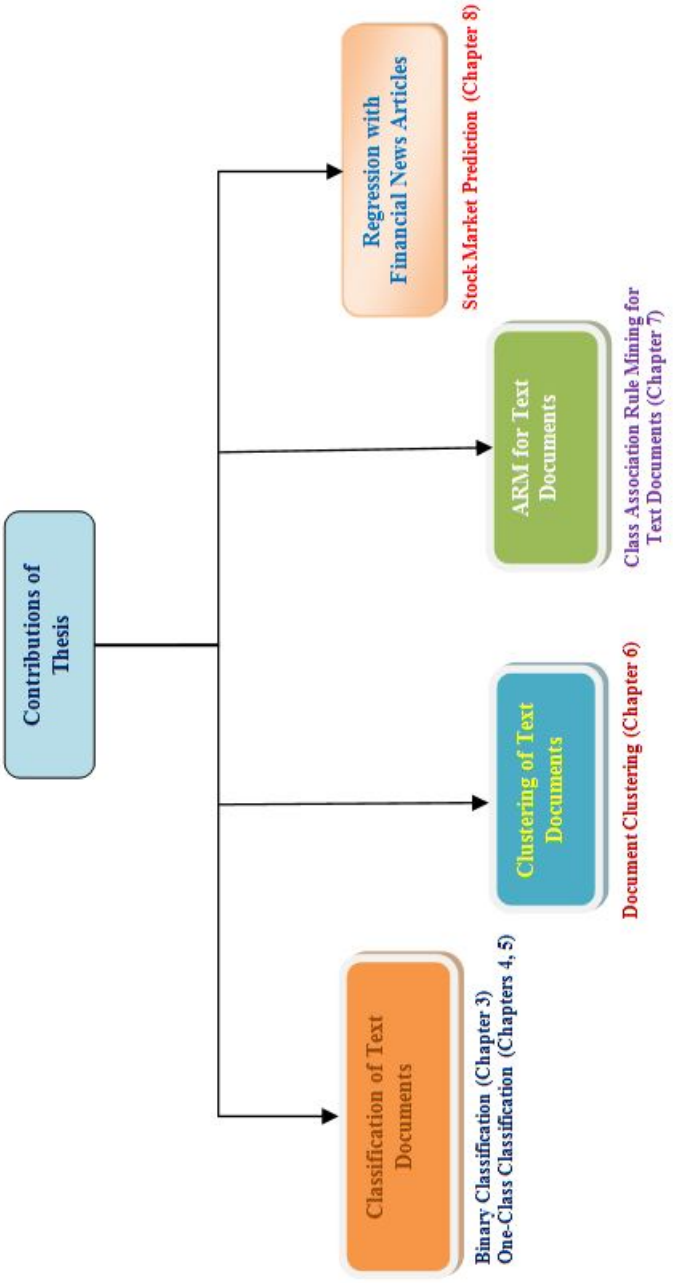

Figure 1.1: Schematic Diagram of Thesis Contributions

Chapter 2

Literature Review

Text mining has found a variety of applications in diverse domains. Of late, prolific work is reported in using text mining techniques to solve problems in various domains. This chapter presents the literature survey of various applications of text mining with respect to the applicable tasks.

2.1 Introduction

Nowadays, in this Internet-dependent world, an enormous amount of data is being generated by several sources. A huge amount of this data is available in an unstructured format. Analyzing the unstructured data using text mining and data mining techniques can enable better decision-making. Text mining can be applied to a broad array of tasks such as document clustering, document classification, text summarization, sentiment analysis, social network analysis, topic detection, web page classification, identification of author, plagiarism detection, phishing/ spam/ malware analysis, patent analysis, financial decision making, etc. The fundamental challenge in text mining is handling the unstructured form of data. It needs to be converted into a structured form before starting the data mining process. The contribution of this chapter is, reviewing the past works comprehensively with respect to dimensions such as

(i) Relevant data mining techniques applied in developing predictive models.

(ii) Data sources used for their analysis.

(iii) Prediction accuracy measures employed.

Previous surveys on text mining dealt with individual applications such as phishing/ spam/ malware detection, financial market etc., and did not cover the entire spectrum. Therefore, this chapter addresses this gap.

2.2 Text mining Tasks

The various tasks of Text Mining are depicted in Figure 2.1. They are as follows: Text summarization, Classification, Topic Detection, Clustering, Association Rule Mining (ARM), and Topic detection/ identification. The applications of text mining listed task wise are as follows: Classification (Phishing, Spam, Intrusion detection) Regression (Forex rate and Stock Market Predictions) The above two tasks address most of the text mining problems. So, we explored these two tasks along with their applications. Apart from this, we also explored the summarization and CRM applications.

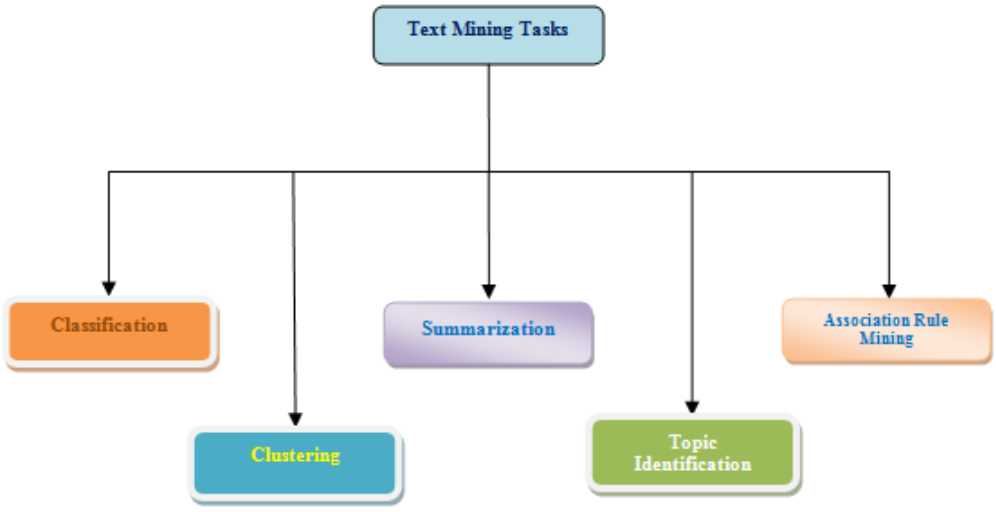

Figure 2.1: Schematic Diagram of Text mining Tasks

Text mining became popular in 1990s, even though it started in 60s, as it was identified as the primary field of information systems. The introduction of

machine learning algorithms for conducting text mining tasks reduced human intervention drastically while also bringing down the amount of time taken to process the text. The first text mining task i.e. document classification was carried out in 1960's by Maron [8]. Documents have one or more keywords for content description. These words are a subset of a controlled dictionary, which contains the collection of synonyms and concepts. With this concept, one of the initial text mining applications was developed by Borko and Bernick [9] for automation of document indexing.

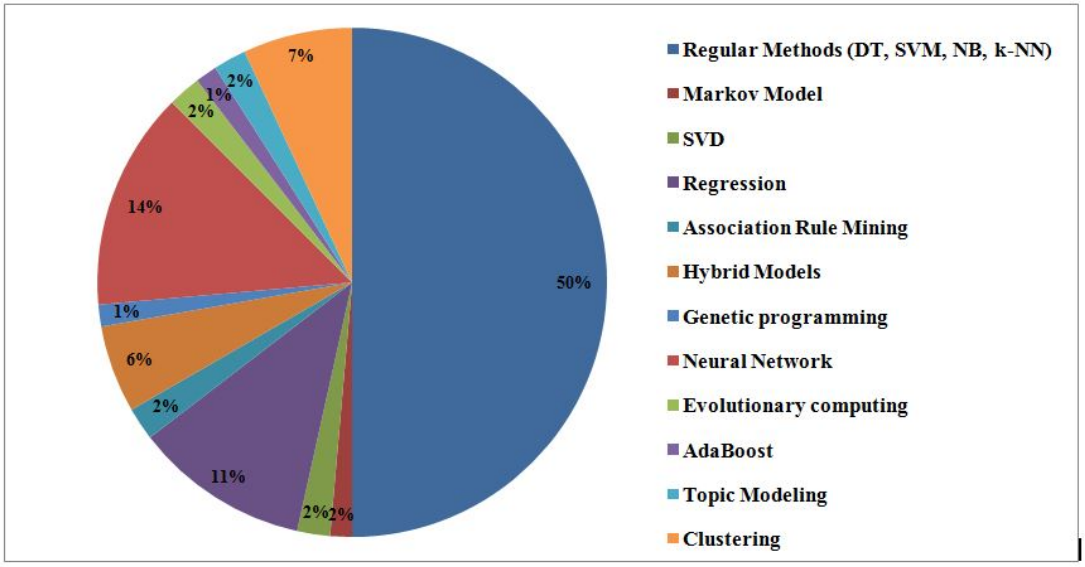

Figure 2.2: Approaches in Various Research Works

A list of the different methods applied in Text mining is depicted in Figure 2.2. The list consists of the regular classification methods i.e. Decision Tree (DT) (Quinlan [10]), Support Vector Machine (SVM) (Cortes and Vapnik [11]), Naive Bayes (NB) (Domingos and Pazzani [12]), and k-Nearest Neighbor (k-NN)(Cover and Hart [13]). Among various approaches, SVM is found to be employed most often and is depicted in Figure 2.2, followed by NB, Neural Networks, Decision Trees, Linear Regression and Logistic Regression etc. Similarly, among the regular classifiers, SVM is the most popular technique followed by NB and DT. The

Figure 2.3 depicts the year-wise distribution of the relevant papers. Similarly, Text mining applications are depicted in Figure 2.4.

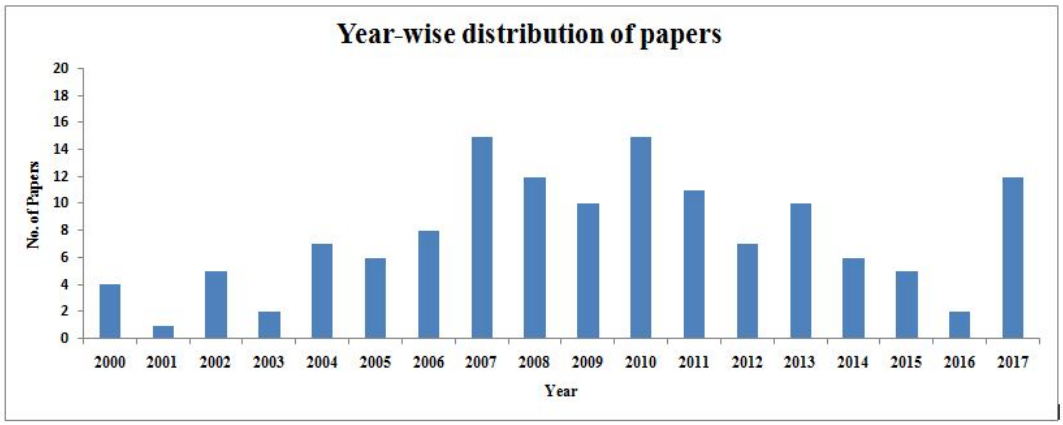

Figure 2.3: Papers Distribution

2.3 Text Mining Applications

2.3.1 Text Mining Applications for Regression

In this section, the works concerning either Forex Rate or Stock Market prediction, or both, have been described.

2.3.1.1 Regression Task for Forex Rate Prediction

The Foreign Exchange (Forex) market changed drastically over the years. The basic idea of this study is to model the human thinking and other aspects that anticipate the direction of financial market movements before making an investment decision. An investor should carefully study the historical trends in financial markets and assess the current situation in order to predict the future (Goodhart [14]). Accurate predictions of Forex rate indeed reduce the market risk emanating from the fluctuations in the Forex rates. Time series data and textual

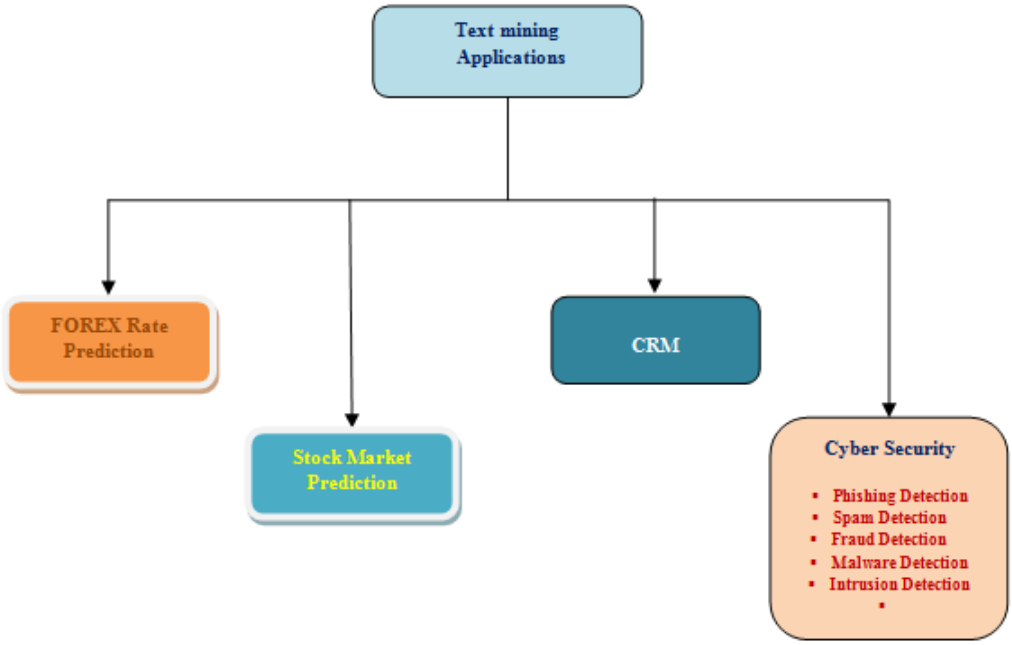

Figure 2.4: Text Mining Applications

data not only contain the effect, but also the possible causes of the event. Better predictions can be made by combining numeric data with text. A recent work on exchange rate forecasting proposed a new model based on the current state of world financial markets. A study of stock market fluctuation based on non-quantifiable information reveals the impact of news reports on the temporal behavior of Forex rates based on an efficient market hypothesis. Fung et al. [15] proposed a new statistical-based partitioning algorithm employing Hierarchical clustering and t-test piecewise segmentation algorithm to identify the trends in time series. This algorithm, clustered the partitioned trends into two groups i.e. rise and drop based on the slope of trends and the coefficient of determination. In order to filter the news articles with the help of clusters obtained from the trends, a guided clustering algorithm was proposed based on incremental k-Means. They also proposed a new differentiated weighting scheme that assigns higher weights

to the features occurring in the rising trend. Evans and Lyons [16] also experimented with macro news for studying the currency flow. In this work, they observed that the arrival of macro news could account for more than 30% of daily price variance and it affected the market participants and market prices. They concluded that macro news impacted two thirds of the exchange rate.

Vu et al. [17] proposed a model for price movements of the stocks based on Twitter messages. Jin et al. [18] proposed a model called Forex-foreteller, which mines news articles and forecasts the movement of foreign currency markets. A combination of language models, topic clustering, and sentiment analysis was used to identify the news articles. The features obtained were combined with historical stock index and currency exchange values. Linear regression model was then invoked for prediction. Yu et al. [19] studied the impact of social media on the stock market performance of a company using NB classifier. A total of 824 firms belonging to six industries (Pharmacy, Software, Health sector, Hotel, and Savings institutions) were analyzed using the postings collected from various sources (forums, blogs, and Twitter). They used stock return value and risk as the performance metrics for evaluation. For each document, sentiment score was calculated based on these values. Chatrath et al. [20] carried out a study on exchange rate prediction based on macro news. They analyzed the rates of four currencies for the period 2005-2010, with intra-day focus and frequency sample of 5-min duration. Currency jumps are a good stand-in for news arrival, and they found that 9-15% of currency changes were directly affected by U.S. announcements. News effect explains 22-56% of the 5-min jump returns, as a negative impact on currency change. Co-jump statistics are strictly dependent on macro news among European currencies, particularly in between Euro and Swiss franc. Some of the works conducted a study on both, Forex rate prediction and Stock Market prediction. They are discussed in the next section.

2.3.1.2 Regression Task for Stock Market Prediction

Stock market prediction is an interesting problem which involves data mining and statistics. Today's biggest economies of the world have higher stock market value. Every market is bounded by supply and demand equilibriums. Human

intervention is limited in stock market prediction. Researchers have been working to predict the stock exchange prices. However, no methodology has been conceived to accurately predict the price movement. It is therefore difficult to predict the stock market price dynamically. The stock market behavior depends on news [21], which is in the unstructured format. The knowledge extracted from the unstructured data can be used for effective decision making. Back et al. [22] proposed a model on clustering approach and they employed unsupervised neural networks called Self Organizing Maps (SOMs) for their study. They experimented with numerical data as well as a combination of the numerical data and text. They concluded that this combination yielded best results. Fung et al. [23] proposed a model for stock market prediction using textual data and time series data mining. Contemporary existing approaches were based on mining only a single time series. They primarily experimented with multiple time series mining with text data in the proposed framework. Through this method, they identified inter-relationships between the time series. For evaluation purpose, they conducted experiments on stock data and news articles of Reuters Market 3000 by employing SVM classifier. By combining the data and text mining techniques, financial reports were analyzed by Kloptchenko et al. [24]. They proposed a model based on SOMs for financial reports analysis. They investigated three financial reports belonging to major telecom companies - Motorola, Nokia and Ericsson companies, which consist of both quantitative (financial ratios) and qualitative (textual data) information.

Koppel and Shtrimberg [25] proposed a model based on news articles, for stock price prediction. They extracted the features from the Multex Significant Development corpus and experimented with the Standard & Poor 500 (S&P 500) stock index. Before the process of modeling, they labeled the news as positive or negative. They reported an accuracy of 70%, by employing SVM for training the model. Wang et al. [26] proposed an Ontology-based model for stock market prediction. In the first phase, they presented a framework based on financial news, market investors, and financial liabilities and in the second phase, they estimated the causality among the news articles and liabilities. They evaluated the proposed approach on the stock trading activity of China Petroleum Corporation (New York Stock Exchange) and 9/11 attack articles.

Dey et al. [27] proposed a model based on LDA for stock market analysis using financial news. The model identifies the events that would affect the stock market and their impact. They extracted topics from the news articles and then clustered them with k-means algorithm. They analyzed groups within Sensex raw data. The prevalent conditions that exist in the economy as a whole are different from those of a particular region. In general, the macro environment includes trends in GDP, inflation, employment, monetary and fiscal policy. This has encouraged researchers to study financial market prediction based on global news and the corresponding works have been mentioned in this review. The key factor of stock market decision making is the selection of the right stock at the right time. To aid in the choice of superior stocks for investment, a finite number of alternatives have been ranked considering several criteria. Multiple Criteria Decision Making (MCDM) has to solve these types of problems. Fasanghari and Montazer [28] proposed a fuzzy-based model for selecting better stocks to model the uncertainty.

Mellouli et al. [29] presented a model for financial headlines representation using ontology. The proposed model had dealt with 201 attributes that belong to 31 concepts. They concluded that the headline articles are categorized with an accuracy of approximately 99% with this approach. They evaluated the model with 227 financial headlines related to the different companies concerning the Toronto stocks. Gilbert and Karahalios [30] performed research on how real world emotions affect the real world things. They demonstrated the estimation of emotions about the future stock market prices, from weblogs information. Anxiety, worry, and fear were estimated from the 20 million posts on LiveJournal site. Based on these emotions, they predicted the movements of S&P 500 index. They analyzed the results of Monte Carlo simulation. Wang et al. [31] proposed a framework for finding the correlation between news articles and financial liabilities. They preprocessed the news articles and financial liabilities using ontology. By invoking the NB algorithm, they figured out the trading function. They classified the news events based on their type using ontology. They concluded that the consideration of news articles and their polarity (positive, negative and neutral) is producing best results compared to considering polarity alone after experimenting with China Petroleum & Chemicals Corporation, SP Power Development, and

Shanghai Composite Index stocks. Chan and Franklin [32] proposed a novel text-based decision support system which extracts event sequences from text patterns and predicts the likelihood of the occurrence of events using a classifier-based inference engine. They investigated more than 2000 financial reports with 28,000 sentences. Experiments show that the prediction accuracy of the model outperformed similar statistical models by 7% for the seen data while significantly improving the prediction accuracy on the unseen data. Further comparisons substantiate the experimental findings. Nizer and Nievola [33] applied text mining techniques with GARCH model for predicting the volatility in the stock market behavior. They built a model on Portuguese news content about companies and their stocks and analyzed its effect on the Brazilian stock market. Moniz and Jong [34] proposed a method based on the counting of negative terms from the dictionary and word counting approach to capture contextual information. LDA model was used to infer the negative effect. They identified the top words associated with the topic clusters. Random Forest algorithm was used for classification and F-Score was used to assess the performance of the model. To test the efficacy of the proposed model, they used the Dow Jones Newswires corpus.

Recently, Nassirtoussi et al. [35] conducted a survey on various methods for market prediction using text mining. They examined various articles based on the data sources used, feature selection methods, applied techniques, and comparison of these techniques. Nassirtoussi et al. [36] predicted Forex market with the news headlines. They proposed a multi-layer dimension reduction technique with the help of semantics and sentiment. In their study, they identified that the previous works on Forex market prediction did not consider high dimensionality and ignored opinion and semantics of textual language. Therefore, they proposed a model with multi-layer architecture. Description of the layers is as follows: first layer acted as a semantic abstraction layer used for determining the co-occurrences of the terms. It enabled the words with same root to be treated as single entity. Similarly, in the second layer they used Sum-Score to integrate the weights of sentiments. It assigned weights to the terms which appeared in sentiments. This layer was helpful for reducing the dimensions of terms in view. Finally, the third layer consisted of dynamic model creation. They updated the models with recent information, which was necessary for prediction. Through this

approach, they reported an accuracy of 83.33% on real-time data. Deng et al. [37] proposed a model called "Lexicon Expansion" for stock market prediction using the text collected from social media. In this approach, they generated the sentiment lexicon which integrates both language and content domain. They analyzed the proposed model on customer reviews, public opinion on politicians, and stock returns. They compared the proposed model with the Opinion Lexicon, SentiWordNet, Multi-Perspective Question Answering Subjectivity Lexicon, General Inquirer models also. They experimented on political tweets (nearly one million), and stock market tweets (7.4 Lakhs) with F-measure as the performance metric. The data appearing on the message boards is also useful for stock prediction. This data is available in the same form as in the other sources considered. Based on this assumption, Galvez and Gravano [38] proposed a model for stock prediction using machine learning methods. They extracted the features from stock price as well as from the message boards also. They employed Random Forest, Ridge Regression models. They deployed the model on Argentinian Stock market with messages extracted from the Argentinian stock message Board. Oliveira et al. [39] conducted a study on stock market forecasting. They predicted the volatility and stock returns using microblogging (Twitter) information. They employed various ML algorithms (Random Forest, Neural Network, SVM, Ensemble Averaging Method, Multiple Regression) for regression and experimented on the stock returns of S&P 500 Index.

Weng et al. [40] proposed a model for stock market prediction using technical indicators and stock time series using online data sources. They constructed the model called "Inference Engine" using SVM, DT, and Neural Network Models. They evaluated the performance of the proposed system on Apple stock value. They predicted the stock movement accurately by 85%. They reported the Precision, F-measure as the performance metric values. Li et al. [41] presented a study called Social Media Date for Sentiment (SMeDA-SA) for stock market prediction using Twitter data. They experimented on the stocks of the 30 companies of New York Stock Exchange and their related tweets (200 million tweets). They predicted the stock movement of the companies with an average accuracy of 70%. Song et al. [42] proposed a model based on relative performance of stocks based on the sentiment of the investors on these stocks. They employed the RankNet,

and ListNet ranking algorithms on ten years of stock data and financial news to rank the articles. They invoked SVM, NN, Boosting algorithms. The effectiveness of the proposed model was evaluated on the S&P 500 stocks returns with the combination of Thomson Reuters News and Bloomberg financial news articles. They concluded that SVM and NN model yielded best results.

2.3.1.3 Regression Task with Respect to Forex and Stock Market

A model for stock price prediction based on news, called NewsCATS (News Categorization and Trading System), classified the new press releases associated with stock prices, based on existing classes (Mittermayer [43]. The SVM algorithm used here for classification examined the stocks of NYSE, NASDAQ, AMEX and five other regional stocks with press releases of Business wire. Antweiler and Frank [44] presented a study on stock message boards. Through Computational Intelligence (CI) techniques, they forecasted the stock returns of the next day using positive messages. Through these message postings, they predicted the volatility of the daily frequencies. Das and Chen [45] developed a model for the extraction of sentiment from stock message boards. They combined different algorithms through a voting scheme. They reported that the performance of the proposed model was better, keeping in view the lower false positive rate as well as accuracy. It was suggested that an aggregation of the cross-sectional message and time series would improve the quality of the sentiment index. They analyzed investor opinions based on news, regulatory changes and announcements made by the company's management.

Mahajan et al. [46] analyzed the impact of news on the stock market. They identified the events by using Latent Dirichlet Allocation (LDA) based on topic extraction method. They analyzed actual market data to understand the impact of news on the market. They generated a model by employing a combination of DT and SVM algorithms. Through this approach, they reported an accuracy of 60%. Tetlock et al. [47] used a quantitative measure of the language to predict the earnings and stock returns of various organizations. They found out the negative words in the firm-specific news and forecasts of the stock returns.

They investigated the S&P 500 companies stock return values based on the articles from the Wall Street Journal (WSJ) and Dow Jones News Service (DJNS). Schumaker and Chen [48] estimated the stock prices, 20 minutes after the corresponding financial news articles were released. The effectiveness of the proposed method which employed SVM, was evaluated on 9,211 financial news articles and 10,259,042 stock quotes covering the S&P 500 stocks during a five-week period. They reported MSE value as 0.04261 for the actual future stock price, direction of price movement i.e. directional accuracy is 57.1% and 2.06% of the highest return. Further, they found that Proper Noun scheme performs better than the rest. Butler and Keselj [49] assessed the performance of stock price, based on the textual financial reports. Initially, they used the Character n-gram (CNG) based method and readability scores method. SVM was then employed for classification. They experimented on Standard & Poor (S&P) 500 index lists.

According to behavioral economics, emotions of humans are very useful in a decision-making process. Bollen et al. [50] extracted feedbacks from Twitter about the Dow Jones Industrial average, over a period. They analyzed the content of Twitter feeds with tracking tools (Opinion Finder and Google-Profile of Mood States). The Opinion Finder measures the positive vs. negative mood and Google-Profile of Mood measures the mood concerning the following dimensions: calm, alert, sure, vital, kind, and happy. To test the hypothesis of this mood prediction they used the self-organizing fuzzy neural network model. They concluded that the Dow Jones predictions significantly improved by the inclusion of specific mood dimensions. They reported a 6% reduction in Mean Average Percentage Error (MAPE) and an overall prediction accuracy of 87.6%. To assist the investors in deciding to buy and sell stocks based on financial news headlines, Huang et al. [51] proposed a model. Text mining and arbitrary association rules were used to estimate the significance of fresh news articles. They demonstrated this approach on Taiwan‘s Stock Exchange Financial Price Index. Li [52] presented the work of analyzing 0-Q and 10-K filings using text mining. For this experimentation, they considered the filings of SEC Edgar (1994-2007). Later, they extracted the Management Discussion and Analysis section (MD&A) from these filings. After preprocessing, they labeled each sentence into four categories

viz., positive, negative, neutral, and uncertain. NB classifier was then employed for the classification task.

Groth and Muntermann [53] explained the implications of news on stock prices. In their work, they identified the risk factors present in the text data. They employed different models viz., NB, k-NN, NN, and SVM for finding the patterns in the textual data. They reported that k-NN performed better as compared to other models. Schumaker et al. [54] proposed a model called Arizona Financial Text system for predicting the price movement using financial news. They developed a system based on financial news articles and combined it with sentiment analysis tools. They employed the Support Vector Regression (SVR) model for this research. In their study, they found that negative words are more useful for the prediction as compared to the positive emotions. One of the key observations in their research was – good news affects stock sale and negative articles leads to buying of the stock. By employing more expressive features to represent text from the market feedbacks, Hagenau et al. [55] proved that robust feature selection helps to improve the accuracy as compared to complex features. They also suggested that selection of semantically relevant features reduces the problem of over-fitting.

2.3.2 Classification Task in CRM Applications

Business intelligence and analytics deals with systems and technologies, practices, and applications to analyze data and mine new knowledge from the markets. These new insights can be useful not only for improving the services but also profits. The subject of Customer Relationship Management (CRM) has become a cornerstone for financial services industry and it encompasses many business problems such as customer acquisition, market basket analysis and customer churn prediction to mention a few and fortunately, all these problems can be formulated as data mining problems. With the advent of call centers, various communication channels and social media, humungous unstructured data is generated which is a treasure trove of useful business insights. In order to extract this business knowledge, one needs text mining. In the current scenario, as products are sold online, the customers' review/feedback is also collected online. This

leads to generation of vast amount of data, which is useful for companies to both analyze and summarize it and therefore identify the customer needs to serve them better. Besides, social media tools like Facebook, Twitter are used by most of the companies to interact with their clients. Opinion mining is helpful to estimate the sentiment associated with the product and also its features. Sentiment analysis is a classification problem where statements are classified into two categories – positive or negative.

Opinion mining impacts the economic growth of the organization also. Numeric ratings alone are not sufficient to analyze the behavior of the customer. Customer feedback (text) analysis plays an important role to explain the behavior of customers. Recommendation systems improve the e-commerce sales in many ways like, viewers to buyers, cross-sell, etc. The impact of recommendation systems on electronic commerce has been studied by various researchers. Reidl et al. [56] presented a work on how the e-commerce websites are benefited based on the recommender systems. They examined its role in sales improvement. They analyzed six popular websites namely Amazon.com, CDNOW, Drugstore.com, eBay, MovieFinder.com and Reel.com. They created a taxonomy for the recommender system (i.e. customers' requirements, techniques for recommendation systems and personalization). They also identified the five commonly used recommender systems applications in e-commerce (raw retrieval, statistical summaries, attribute-based, item-to-item correlation, user-to-user correlation). Pang et al. [57] conducted a study on sentiment classification using unigram, bigram and n-gram models. They applied some of the well-known machine learning methods such as NB, Maximum Entropy and SVM on the movie reviews. They concluded that the results using these techniques were better as compared to human generated values.

Hu and Liu [58] performed text summarization based on customer reviews. Initially, they identified the product features followed by the opinion of the client i.e. positive or negative. Later, they summarized these results. Customer opinions/reviews are a gold mine for market competitors. A huge amount of this information is available on the web. Liu et al. [59] proposed a framework for the examination of the customers' opinions on products. They developed an opinion observer for the manufactures for the comparison of customer opinions. Popesu

and Etzioni [60] developed a tool called OPINE for mining the reviews of the customers. They carried out the experiments on Amazon reviews and they reported best values of precision and recall values. Ghani et al. [61] proposed a model for extraction of attributes and its associated values from the textual description of the products. They extracted data from the retail stores, URLs, etc. from the web. They used wrappers for extraction of information from the sites. As per the domain expert's suggestion, they primarily used eight attributes of each product. NB and Expectation Maximization were used for text classification with data of approximately 600 products, to train the model. They used 5-FCV to evaluate the model which could predict future demand for the products, recommendations about the product and similarities between various providers. Devitt and Ahmad [62] proposed a model based on lexical cohesion for finding the sentiment polarity in the financial news. They examined the relationship between financial news and the stock market. They investigated on the polarity direction (positive/ negative) and its strength on the stock value.

Coussement and Van den Poel [63] proposed a new method for complaint handling strategies through email classification that distinguishes complaints from non-complaints by building a classification model using the linguistic information. Linguistic style features were extracted from 9176 emails out of which 3299 were complaint-based emails and rest of them were general. They proposed three different models using Adaboost (ADA) classifier viz., Adaboost-Singular Valued Decomposition (ADA-SVD), Adaboost-Linguistic Style feature (ADA-LS) and Adaboost-Singular Valued Decomposition with Linguistic Style feature (ADA-SVD-LS). PCC and AUC metrics were used for evaluating the model. They collected and evaluated the call center e-mails of a Belgian newspaper. They reported that ADA-SVD-LS performed better as compared to other two approaches. Pang and Lee [64] presented a survey paper that describes the methods of sentiment analysis and its applications. They compared various traditional (fact-based) methods. They also mentioned different types of publicly available datasets and competitions regarding opinion mining analysis. Sayyadi et al. [65] worked on news event detection. They proposed a model that detects news events with label based clustering approach. Clustering is carried out based on similarity of articles of an event. The model distinguished such events and merged the similar

items. They implemented the proposed model using RSS (Really Simple Syndication) technology. Through this approach they achieved an accuracy of 80%. Bifet and Frank [66] proposed a model for analysis of twitter data streams and extracted the features from twitter data. They used datasets from twittersentiment.appspot.com and Edinburgh corpus. They obtained 10000 unigrams using Weka. They used term presence for vector space model creation and employed various models like Multinomial Naive Bayes, Stochastic Gradient Descent (SGD) and the Hoeffding Tree. They reported that SGD outperformed other models in terms of Kappa statistic value of 62.6% and accuracy of 82.8%. Dey et al. [67] proposed a framework using NLP and Ontology to extract the knowledge from customer opinions.

The success factors of the companies also depend on the information available on their sites. Based on this, Thorleuchter and Van den Poel [68] developed a new model. They extracted the web content of top 500 companies and employed Latent Semantic Indexing (LSI) to identify the semantic patterns in the text. They classified the top 100 (positive class) and the remaining 500 (negative class) e-commerce companies by employing Logistic Regression (LR). The performance of the regression model was evaluated with the measures of lift, ROC, precision and recall. Liu and Zhang [69] presented a survey of opinion mining and sentiment analysis. They defined the objective of opinion mining in terms of sub tasks (extraction, classification, etc.). They also discussed various types of opinion mining like aspect-based, sentiment classification, dictionary-based approach, etc.

Twitter messages are commonly used to determine the sentiments of users. Most of the companies use sentiments to find out the brand/product sentiment. Ghiassi et al. [70] proposed a new model based on supervised feature reduction using n-grams for a similar problem. They developed a new model called Dynamic Architecture for Artificial Neural Networks (DAN2) for sentiment classification. They compared the proposed model with SVM classifier and reported that DAN2 performed better than SVM in terms of accuracy and recall. They carried out the experiments on randomly selected tweets with manually labels assigned. Many people express their opinions in the social media like Twitter, Facebook, etc. and these vary from one demographic field to the other. Ikeda et al. [71] proposed a hybrid model based on text mining and community analysis for demographic

user's analysis. They experimented on the tweets of 1Lakh user profiles. They evaluated the proposed model with the following measures: Recall, Precision and F-measure.

He et al. [72] conducted a study on pizza suppliers. They collected data of Pizza Hut, Domino's pizza and Papa John's pizza from Facebook and Twitter sites. They identified the behavioral patterns related to these suppliers in Facebook and Twitter. They carried out the research using SPSS Clementine and Nvivo 9 tools. SPSS Clementine was used for extracting the key concepts, indexing and grouping the text. They employed NVivo 9 for query search that retrieves patterns and connections. Most of the tweets were about the orders and delivery, pizza quality, purchase decision and marketing. It was reported that the recommendations provided by this study helped improve the business. Verma et al. [73] detected events from political and other news articles. The significant difference between the business events and other activities are that business events are often announcements of future happenings, etc. They proposed a method to identify critical business events from the news articles and then classify them accordingly by invoking the k-NN classifier. Ballings and Van den Poel [74] assessed the feasibility of Facebook usage frequency prediction with six classification algorithms namely Random Forest, Kernel Factory, Logistic Regression, Neural Networks, Support Vector Machines and Stochastic Adaptive Boosting. They studied the deviation from regular patterns which would help in customizing the services like advertisements and recommendations. They reported the highest accuracy of 74% and AUC of 0.66 with Stochastic Adaptive Boosting algorithm. Recently, a survey on the works on sentiment analysis and opinion mining published during 2002-2014 was conducted by Ravi and Ravi [75]. They reviewed the tasks and applications of opinion mining. They presented data sources and methods for building models.

2.3.3 Classification Task in Cyber Security Applications

Cyber security is of paramount importance in financial services industry, as it is increasingly depending on the information technology for delivering products and services. While on one hand, technology provides convenience and comfort to

customers, it also opens up doors to numerous forms of cyber crimes. Cybercrime is closely related to the economic impact of a company/ country. According to the MacAfee [76] report, the economic or financial losses of the firms due to cybercrime varied from 375 to 575 billion, which are greater than some of the countries' annual income. This figure shows the importance of the problem called Cyber Security. In 2015 more than 54 Million people in Turkey, 40 Million people in the USA, 20 Million people in China and Korea, and 16 million citizens in Germany were affected by cybercrime activities. According to Symantec Internet Security Threat report (ISTA [77]), malicious activities are more from China followed by the United States and then, India. They analyzed various malicious activities and their effect on various countries. There is a lot of gap between the actual cost and the recovery cost due to cybercrime. It shows an excellent scope for us to explore the cybercrime activities (prevention, detection, and recovery). In this chapter, we categorized the cyber security applications into five types viz., Phishing, Spam, Malware Intrusion Detection and Fraud detection.

2.3.3.1 Classification Task in Phishing Detection Using Text Mining

Phishing is a widespread problem that is affecting both businesses and consumers. Of late, phishing attacks have increased drastically. The goal of phishing is to first steal the identities and credentials of a genuine user and then siphon off funds from his/her account remotely without his/her knowledge. Attackers adopt numerous strategies to attract or take control of users through counterfeit websites. The common factor among all these phishing websites is that they make the users believe that they are actual websites and mislead them in the process. Although phishing can be detected at both, email and website levels, analyzing phishing detection is a pressing problem, on which researchers across the world have been working.

Pan and Ding [78] proposed an SVM based approach, which was independent of any specific phishing implementation. They examined the anomalies in web pages, discrepancies between website identity, its structural features, and HTTP transactions, which does not require the users' knowledge of the website. Reports as per the data collected from the most frequently attacked websites revealed

that majority of the victims were from financial institutions and e-commerce companies. The values obtained through this approach achieved low miss rate and low false-positive rate. Dhamija et al. [79] provided the first experimental proof that explains about how malicious strategies work. They mentioned various strategies for phishing such as the lack of awareness among users, deceitfulness and lack of attention after analyzing 20 websites.

Chandrasekaran et al. [80] proposed a One Class Support Vector Machine (OCSVM)model based on structural features for phishing detection involving 400 emails. Abu-Nimeh et al. [81] compared the performance of CART, LR, Bayesian Additive Regression Trees (BART), SVM, Random Forest (RF), and Neural Networks for phishing emails detection. They experimented on a dataset consisting of 2889 emails with 43 features. Zhang et al. [82] presented the design and implementation of CANTINA. It is a novel, content-based approach to detect phishing websites. Using the tf-idf term weighting scheme, they identified phishing websites with 95% of accuracy.

Ludl et al. [83] presented a paper on phishing detection. They analyzed 1000 phishing sites using two basic approaches namely, blacklist and page analysis. In the first approach, they compared the URL with the existing blacklisted URLs to find out whether the particular URL is phishing or not. In the second method, they analyzed HTML source code which contained 18 features. C4.5 was employed for classification purpose. Garera et al. [84] identified some measures to find phishing URLs. In this study, they employed Logistic Regression (LR), leading to higher accuracy and found some key features such as page-based, domain-based, type-based and word-based features. Fette et al. [85] proposed a model with 10 most important features for identification of the phishing emails. SpamAssassin and Phishing Corpus were used for evaluating the model. They experimented on the corpus containing 6950 legitimate and 860 phishing emails. Classification task was performed with Random Forest and SVM classifiers. They reported that a detection rate of 96% of was achieved with Random Forest. Miyamoto et al. [86] constructed a model with nine methods viz., SVM, NN, NB, LR, Random Forest, CART, AdaBoost, Bagging, and Bayesian Additive Regression Trees for detection of phishing sites. They analyzed a dataset comprising 1500 phishing sites and 1500 legitimate sites. For performance measure, they used F1 measure,

error rate and Area Under the ROC Curve (AUC). They reported that Adaboost performed the best, with an F1 value of 0.8581, error rate of 14.15%, and AUC value of 0.9342.

Based on structural properties of an email, Basnet et al. [87] identified 16 features and proposed both, supervised as well as unsupervised techniques viz., SVM, NN, SOM and K-Means for phishing detection. In this work, they used SpamAssassin and Phishing Corpus, which consist of 4000 emails in which 3027 are legitimate and 973 are phishing. They reported that among all the techniques, SVM produced the highest accuracy. Xiang and Hong [88] proposed a hybrid model with Information Retrieval (IR) and Information Extraction (IE) methods for phishing detection. Two components were mainly used – one was to discover the identity imitation and another one for keywords retrieval. They achieved a true positive rate of 90.06% as the highest value. Sheng et al. [89] presented a paper on the performance of phishing detection based on blacklist approach. They experimented with eight anti-phishing toolbars viz., Internet Explorer 7 and 8, Firefox 2 and 3, Google Chrome, Netcraft Toolbar, McAfee Site Advisor, and Symantec Norton. Alabama Phishing Team experimented on their legitimate URL's and Phishing URL's. Bergholz et al. [90] proposed a model for phishing detection based on the categorization of different features like structure (body parts), link, element (HTML, Javascript), spam filter and wordlist (account, update, confirm, verify, secure, notify, log). They experimented with a corpus consisting of 16364 non-phishing and 3636 phishing emails. Aburrous et al. [91] presented e-banking phishing detection model developed using fuzzy data mining approach that combines the data mining and fuzzy techniques. They also generated different models with C4.5, RIPPER and CBA.

He et al. [92] extracted 12 features from web pages and trained a model with SVM classifier. They reported high accuracy rate with relatively low-false positive and low-false negative rates. Chen et al. [93] analyzed 1030 phishing sites with a combination of text mining and financial attributes. An accuracy of 89% was achieved through their proposed model. Mohammad et al. [94] developed a tool that automatically extracted features from websites. They collected 2500 URLs from PhishTank and experimented with 17 features. They found that "Request URL" is the most important feature among all. Pandey and Ravi [95] performed

the detection of phishing websites and spam using text mining. In this study, they built various models with GP, LR, PNN, MLP, CART, GP+CART and reported superior results. In both the studies, (i.e. phishing email and websites), they indicated that sensitivity was higher with GP classifier, and included the statistical significance test (t-test) of these results.

Khonji and Iraqi [96] presented a literature survey on phishing detection that included four classes of techniques employed by researchers to approach the problem viz., visual similarity, machine learning, blacklists, and rule-based techniques. Detection rate and low false positive values were used for evaluation purpose. They concluded that their work outperformed other approaches. Abdelhamid et al. [97] presented a review paper on "Phishing detection based on Association Rule Mining technique". They developed an algorithm based on association rule mining named as Multi-label Classifier based Associative Classification (MCAC) for phishing detection. They described various approaches followed by the researchers for phishing detection namely Blacklist based fuzzy rule-based, machine learning techniques, CANTINA, image-based methods. They also presented different sets of features, which are discriminative for identifying the phishing websites. Feature subset selection was performed using chi-square method.

2.3.3.2 Classiifcation Task in Spam Detection using Text Mining Approach

Spam is defined as a junk mail sent by unsought person. It consists of viruses, Trojans etc. The Symantec report [77] outlines the demographics and organizational wise spam activity information. Email spam is one of the major concerns in the internet world as it has potential in creating financial losses. Spam detection can be carried out mainly in two ways. One is text-based analysis, and another is an image-based method. Various feature selection algorithms in literature have been proposed in the literature. Some of them used in spam filtering analysis methods are Document Frequency, Information Gain, Chi-square Statistic, Odds Ratio and Term Frequency. Androutsopoulos et al. [98] proposed Naive Bayes based anti-spam model. They experimented with Ling-Spam corpus. Earlier to this, Androutsopoulos et al. [99] conducted a similar type of work in the same

year, which proposed a model to detect the spam messages built using the NB classifier. They evaluated their model on the PU1 corpus which consisted of 1099 messages (481 spam and 618 legitimate).

Massey et al. [100] reviewed machine learning methods for spam filtering. They applied various feature selection methods and evaluated the model with different classifiers (ID3, NN, etc.) on four corpuses namely Ling-Spam, Spam Assassin, Annexia Spam Archive and their own email collection. Zhang et al. [101] applied SVM, AdaBoost, Maximum Entropy Model, NB, and Memory-Based Learning with various feature subsets for spam filtering. They employed Information Gain, Document Frequency, and Chi-square for Feature Subset selection. They analyzed four datasets viz., PU1 (481 spam and 618 legitimate), Ling-Spam (481 spam and 2412 legitimate), SpamAssassin (1897 spam and 4150 legitimate) and ZH1 Chinese Spam Corpus (1205 spam and 428 legitimate). Klimt and Yang [102] performed spam detection with a new dataset on Enron corpus with SVM classifier. They reported F-Score of 0.7. Metsis et al. [103] worked on the spam filtering with the Naive Bayes (NB) approach. Spam emails sent into the hijacked systems for a short period are called Transient spam-bots. Brodsky and Brodsky [104] presented a framework that fights these types of problems. Chen et al. [105] analyzed spam filtering using three variants of Bayesian methods namely Aggregating One-Dependence Estimators (AODE), Hidden Naive Bayes (HNB), and locally weighted learning with Naive Bayes (LWNB). They selected relevant features with the following feature selection methods: Gain Ratio, Information Gain, Symmetrical Uncertainty and ReliefF. They compared the results of three classifiers with the linear classifiers NB, k-NN, SVM, and C4.5. They reported that Aggregating One-Dependence Estimators (AODE) performed the best with respect to accuracy.

Chen et al. [105] employed Artificial Immune System (AIS) a method for spam filtering. They proposed a classification model that classifies the emails based on AIS gene fragments. The experimental results revealed that the proposed model is effective for spam detection. Similarly, Blanzieri and Bryl [106] conducted a survey on email filtering techniques. They provided an overview of the data mining and machine learning techniques applied to spam filtering. Besides analyzing various methods, they also discussed commercial and non-commercial software

solutions. Guzella and Caminhas [107] presented a comprehensive study of spam filtering with data mining techniques. They covered both textual and image-based approaches and studied the structure of spam filter and representation (i.e. document frequency, information gain, term frequency variance). They listed out the publicly available datasets and various classifiers including SVM, NB, LR, ANN and hybrid methods employed by researchers.

Anomaly detection is one of the important ways to identify the abnormal patterns/behaviors to quickly determine the suspicious transactions which deviate from the normal behavior. Finding deviations from the regular patterns is called Anomaly Detection (Denning [108]). There exist various methods to characterize these anomalies. In this study, we presented some of the cases based on system call traces. Few researchers attempted to solve the intrusion detection problem through machine learning approaches. Zhan et al. [109] applied anomaly detection in the email system to find out whether an email is normal or spam. They proposed the weak estimators such as Stochastic Learning-Based Weak Estimator (SLWE) and Maximum Likelihood Estimator (MLE) for estimating the distributions of events which deviate from the normal pattern. In this study, they used Information Gain to find out the top 200 discriminative features and performed classification with NB classifier. They compared the Precision, Recall and ROC values obtained by the above two approaches. Identification of Spam is an expensive and time-consuming process. Caruna and Li [110] presented a survey on this, which focused on emerging approaches to spam filtering built on recent developments in computing technologies. These include peer-to-peer computing, grid computing, semantic web, and social networks. It also addressed many perspectives related to personalization and privacy in spam filtering. They concluded that despite advanced methodologies being used, attaining high performance and detection rate is still an open problem.

Tan et al. [111] employed statistical methods with Artificial Immune System for Spam Detection. Pandey and Ravi [95] performed spam detection using text and data mining techniques. They analysed the spam corpus consisting of six subgroups. They used different methods such as LR, CART, GP, MLP, and PNN for classification purpose. 10 Fold Cross Validation (10FCV) was performed on

the dataset for evaluating the model and statistical significance of the results was provided. They reported that MLP performed the best.

2.3.3.3 Classification Task in Malware Detection Using Text Mining Approach

Another challenging problem for cyber world is malware detection. It is both important as well as relevant, which is evident from a Gartner's Magic Quadrant report [112]. As per the survey, companies spent approximately 2.8 billion in 2011 for malware protection. For malware protection, organizations must have anti-malware, anti-spyware, vulnerability assessment, personal firewalls and finally host-based intrusion prevention. Despite the high-level security mechanisms, system vulnerabilities became a powerful weapon for malware authors. There are two types of malware detection techniques: anomaly-based detection and signature-based method. In this work, anomaly-based malware detection has been analyzed. Numerous machine learning techniques have been applied in the past few years to analyze malware. Wang and Stolofo [113] developed a model called Payload-based Anomaly Detector (PAYL) and reported an overall 60% of detection rate with 1% false positive rate. Vasudevan and Yerraballi [114] defined variants of Malware as viruses, Trojans, and Spywares. Ye et al. [115] developed a method named Intelligent Malware Detection System (IMDS) using Association Rule Mining, and further employed NB, SVM and J4.8 classifiers which produced accuracies of 83.86%, 90.54% and 91.49% respectively, while their proposed (IMDS) approach yielded 93.07%. Provos et al. (2007) also performed research on web malwares.

Idika and Mathur [116] carried out research on different malware detection techniques and reported their limitations. In their work, they compared 45 techniques about malware detection. Ahmed et al. [117] proposed an amalgam malware detection scheme using run-time Application Program Interface (API) calls of Windows OS. It consists of offline training and online testing phases. Out of thousands of API calls in Windows, 237 calls were chosen for both Benign and Malware programs. Further, these API calls were segregated into seven groups based on their functionality. They categorized different API calls, resulting in

0.97 detection rate. Malwares were grouped based on their common characteristics by Ye et al. [118]. They proposed Automatic Malware Categorization System (AMCS) by combining k-medoids algorithm, Hierarchical clustering and weighted subspace k-medoids algorithm. They evaluated the efficacy of the proposed model on the malware dataset of Kingsoft Anti-Virus Lab. The performance of the model was measured by using F-measure. Hou et al. [119] analyzed the characteristics of a web page to determine whether it is malicious or not. They collected 1141 URLs from StopBadWare site. Information Gain was used for selection of distinctive features, followed by model construction with NB, DT, SVM and Boosted Decision Tree classifiers. They performed 10FCV and reported that Boosted Decision Tree outperformed other models.

Zhuang et al. [120] developed a model for malware and phishing websites categorization using a cluster ensemble method involving hierarchical clustering and k-medoid algorithm. In this work, they represented the malware with static feature extraction methods. They collected the malware dataset from Kingsoft Internet Security Laboratory during the period June 10 – 16, 2012. Feature vectors were generated with tf and tf-idf weighting schemes. Sundarkumar and Ravi [121] proposed a method for Malware analysis using Windows API calls through text mining and they selected the discriminative features using Mutual Information. Classification task was executed by DT/SVM/MLP/PNN/GMDH with 10-FCV after oversampling the dataset. They reported sensitivity values of 100% with SVM and OCSVM. Suarez-Tangil et al. [122] proposed a model to classify smart phone malware using text mining. Recently, Sundarkumar et al. [123] presented a method based on topic modeling and machine learning to detect malware. They used API calls for detection of malware based on features selected by topic modeling. They employed different algorithms viz., DT, SVM, PNN, GMDH, MLP and RF for classification purpose. They experimented with two datasets, and concluded their work with the statistical significance testing.

2.3.3.4 Classification Task in Intrusion Detection Using Text Mining Approach

Cyber fraudsters adopt novel and potent methods to hack computer systems in order to cause maximum damage in just one go. Pharming or DDoS attack are some of the ways of doing it. If successful, it can result in maximum damage. Liao and Vemuri [124], employed text mining techniques to predict abnormal program behavior and evaluated the model with k-NN classifier on DARPA BSM dataset. They reported low false positive values. Helmer et al. [125] proposed a network based multi-agent distributed system for intrusion detection. For a process, they maintained the feature vector representation, and based on the execution of a sequence (at attack) it was labeled as good or bad. Feature selection was carried out with genetic algorithm (GA). They compared the results with and without feature selection. RIPPER was employed for classification and if-then rules were generated for intrusion detection. Finally, they concluded that the combination of GA (for feature selection) and RIPPER (for classification) yielded the best results.

Liu et al. [126] presented a model for intrusion detection based on neural networks. They employed three types of neural networks viz., Back propagation, RBF networks, and Self Organizing Map. They experimented with the Unix System calls of lpr (2703 normal and 1001 abnormal traces) and send mail (172 normal and three abnormal traces) programs, which were obtained from University of Mexico. Chen et al. [127] performed intrusion detection on the DARPA 98 dataset with ANN and SVM classifiers. They generated the tf-idf matrix with the system calls and classified using the above classifiers. They listed out the most commonly used 50 system calls and reported the FP and the detection rate values. By using server logs of a website, intrusion detection was performed by Adeva et al. [128]. They applied the model to telemedicine systems for detecting unauthorized access to the application. Methods like term frequency, document frequency, chi-square, information gain were employed for feature subset selection. A total of 5000 log entries (including normal and abnormal) were used to train the model. For testing purpose, 16995 normal, and 2235 abnormal log entries were used. They evaluated the model with Rocchio, k-NN and NB classifiers.

The study reported that the combination of chi-square feature selection method and NB classifier performed better as compared to other methods.

Rawat et al. [129] proposed a model for intrusion detection. They introduced the binary weighted cosine metric as a similarity measure. The model was employed to classify system calls into normal and abnormal. They evaluated the model with k-NN classifier on DARPA 98 project. They reported low false positive rate with the proposed approach. Sharma et al. [130] presented a model built using k-NN classifier for intrusion detection and evaluated the proposed approach on DARPA 98 data set. Adeva and Ataxa [131] carried out research on intrusion detection in web applications by using the log entries generated by a web server. The proposed model was evaluated with Rocchio, k-NN and Bayes classifiers with F-measure as metric. They reported the highest F-value with NB classifier.

2.3.3.5 Classification Task in Fraud Detection Using Text Mining

Churn prediction and bankruptcy prediction have been the most interesting areas of research in financial domain (Shirata [132]). Most of the existing works are based on numerical data such as financial ratios etc. Recently, some of the works were carried out based on quantitative as well as qualitative information. Few of the works are presented in this section. Appavu et al. [133] proposed a method based on DT called AD Infinitum for identifying the malicious emails. They classified the emails based on the incremental method, and they created two corpora namely, TCEThreatening1 and CEThreatening2. They compared the results with regular classifiers (DT, SVM, NB). They reported that the performance of the proposed method yielded best results in terms of F-measure and Accuracy. Several methods to detect fraud from financial statements have been reported in the literature. Kamaruddin et al. [134] explains the use of outlier patterns in the data to detect fraud. They proposed a framework, which included preprocessing and representation of financial statements on conceptual graphs. Using conceptual similarity and relative similarity measures, the variations are determined.

Cecchini et al. [135] proposed a model to analyze financial events with text-based ontology creation. They created a dictionary called Management Discussion

and Analysis section (MD & A) based on companies' bankruptcy. They carried out a research on Bankruptcy dataset (78 companies) and Fraud dataset (61 fraud and 61 non-fraud companies). Performance of the model was evaluated with accuracy values. They reported the prediction accuracy values as follows: 83.87% for Bankruptcy and 81.97% for Fraud datasets. Shirata et al. [132] proposed a model to predict bankruptcy based on textual data from financial statements in Japan. They experimented on annual reports of 180 companies for the period of 1999-2005 (90 bankruptcy and 90 non-bankruptcy) which are collected from the Tokyo stock exchange. They identified some specific words which are helpful for predicting bankruptcy. They employed CART-based SAF (Simple Analysis of Failure) model for prediction in the OmniFind Analytic Edition (OAE) tool. Sometimes, the data is imbalanced, and is not suitable for prediction tasks.

Glancy and Yadav [136] applied quantitative approaches to text corpus in order to detect fraud in imbalanced financial statements data. Manual auditing is highly inefficient and is not very accurate because of human miscalculation. To avoid such inaccuracies, Saha et al. [137] proposed a model for processing bank loans. Through text mining approaches, they analyzed 100 fraudulent cases of small-scale and medium-scale industries. They analyzed these cases with the help of five reputed auditors to reveal risk level, risk impact, and risk detection. A binary value was assigned based on the domain experts' evaluation, and these values served as output labels for the regression model. They employed logistic regression to classify the fraudulent applications. Hajek and Henriques [138] proposed a model for fraud detection in financial statements. They extracted the features from both financial information, and managerial comments (text) of the companies (Telecom, Finance, Banking, Health, and Software) from the annual financial reports. They employed SVM, NB, Logistic Model Trees, CART, and Bayesian belief network (BBN). They examined the financial statements of 622 firms (311 fraudulent, and 311 non-fraudulent) collected from Accounting and Auditing Enforcement Release (USA). They concluded that NB performed better than other models in terms of misclassification costs.

2.4 Text Summarization Task

In addition to the above applications, some of the text summarization works are also reviewed as follows: Text summarization is one of the primary tasks of text mining. Information extraction and text summarization were done by Rau et al. [139] in 1988. Text summarization is defined as follows: generating a compressed version of the given text which provides useful information for the end users [140]. There exists two types of text summarization techniques - abstractive, and extractive. In the extractive method, a set of representative sentences are picked from the original text. Whereas, abstractive summarization contains novel phrases which are not unseen/ present in the original sources.

Text summarization is a challenging area in text mining. Summarization can contain a single document or multiple documents. Text summarization comprises three phases namely, analysis, transformation, and synthesis. In the first phase (analysis), input text is processed followed by feature selection. In the second phase, the text is transformed into a summary format. In the final phase, an appropriate summary is produced according to the user's requirements. In summarization, the following two things play a significant role: process and compression rate. The compression rate is the ratio of length of the summary to the length of the original text. The quality of summarization depends on the compression rate. If the summary is concise, then we can say compression rate is low, and information loss is high. i.e., compression rate and information loss are inversely proportional. Similarly, if the summary is lengthy, then compression rate is more and information loss is less ([141], [142], [143]). Yeh et al. [143] proposed two models for text summarization. One is using LSA with TRM (Text Relationship Map), and another one is Corpus-Based Approach (MCBA). In the first approach, a semantic measure is obtained using LSA followed by the construction of a Text Relationship Map (TRM). In the second approach, they initially ranked the sentences based on their significance. Later, the feature weights are obtained through genetic algorithm. They evaluated these models on 100 political articles with compression rate and F-measure as performance metrics. Text summarization could be formulated as an optimization problem, with an objective of minimizing the information loss. Ye et al. [144] proposed a model for document summarization

using the lattice concept. As per the proposal, a hierarchy of topics is tied to familiar concepts/ topics and these topics contain corresponding sentences. They experimented on the Document Understanding Conference (DUC) dataset. Most of the sentiment analysis models classify the reviews as either positive or negative. This is not enough to process the customer reviews, since they are ignoring the important information in the reviews. To overcome this, Zhan et al. [145] proposed an approach for customer reviews summarization. In this, they initially extract the topics from the customer reviews. Later these topics are ranked and finally, summary is generated based on these rankings. Fattah and Ren [146] proposed a model for content selection in text summarization by employing the statistical models. They researched on the effect of sentence features. Initially, they trained the Genetic Algorithm (GA) with all features. Then, the combination of feature weights was obtained using Mathematical Regression (MR). Later, they trained the PNN, Feed Forward Neural Network (FFNN), and Gaussian Mixture Model (GMM) with the feature parameters. They experimented with real-time datasets and concluded that GMM is performing better as compared to other models.

Binwahlan et al. [147] proposed a hybrid model for text summarization. PSO was employed on the data to determine optimum weights for the features. In each iteration, new summaries are created and are reviewed manually. Next phase calculates the sentence score for each sentence using the Fuzzy system. Few text summarization works are based on multi-document summarization. Kumar et al. [148] proposed a model called Genetic-Case Base Reasoning (GCBR) for document summarization. They calculated the sentence score using fuzzy approach and then figured out the document relations. They experimented with CST bank and Document Understanding Conference datasets with ROUGE as a performance metric. Mosa et al. [149] proposed a framework for text summarization using Ant Colony Optimization (ACO). They invoked ACO with Jensen-Shannon Divergence (JSD). Through this method, they were able to capture the essential information during summarization. They performed an analysis of comments extracted from Facebook. Category information of text is also helpful for text summarization. Based on this assumption Jeong et al. [150] proposed an integrated framework for document summarization. In this work, text summarization was performed using a feature weighting scheme. This was accomplished by

determining the feature distribution of each class by employing the Maximum Entropy Model (MEM), SVM, and NB algorithms. They conducted the experiments on the AbleNews, KORDIC datasets with F-measure as the performance metric. They mentioned that some of the advantages of this model are language independence and simplicity.

Wu et al. [151] proposed a model for text summarization based on topic modeling. They extracted the sentences associated with topics to achieve the compression ratio and diversity of the topics. They concluded that through this proposed model they obtained higher compression ratio and better summarization quality. They evaluated this approach on a novel dataset. Yousefi-Azar and Amey [152] proposed a model for text summarization using deep learning. They invoked an autoencoder to compute the feature space from the document-term matrix. They evaluated the proposed model on an email corpus with recall and ROUGE as a performance metric.

Recently, Rautray and Balabantaray [153] proposed a model for multi-document summarization using Cuckoo search. They compared the results with Particle Swarm Optimization (PSO) based summarization and Cat Swarm Optimization (CSO) based summarization. They evaluated the proposed model on the benchmark DUC dataset (Document Understanding Conference). They reported that through this approach they obtained the best Recall-Oriented Understudy for Gisting Evaluation (ROUGE) value. Hu et al. [154] proposed a model for text summarization to find the most informative sentences. In this approach, initially they identify the important sentences, followed by similarity (both content and sentiment) calculation. Later, they invoked the k-Medoids algorithm to group those sentences. They evaluated the proposed model on reviews extracted from the TripAdvisor website with usefulness as the performance metric.

2.5 Conclusions

This chapter presents a comprehensive review of text mining applications with respect to the associated tasks. Following are some of the common tasks viz., Classification (Phishing, Spam, Malware, and Intrusion detection), Forecasting

(Stock and Forex prediction), CRM and text summarization. Although there exists several text mining applications, devising more efficient techniques is essential for handling and predicting a significant amount of data. In our study, we found that,

1. Considerable progress has been achieved in the application of text mining to solve problems in various domains. However, there remains a lot to be achieved further. Identifying a suitable feature selection method is still an open problem. Since datasets in this domain have high dimensionality, dimension reduction becomes critical to the success of the data mining techniques.

2. Classification is the most often performed task in text mining, followed by forecasting.

3. SVM, NB, k-NN, and DT are the most often used data mining techniques in text mining applications. Out of them, SVM is the predominant technique applied in various applications because of its high prediction capability in the presence of the large number of features.

4. Performance evaluation metrics are not unique and vary from application to application.

5. Stock market prediction and Forex rate problems are frequently solved using text articles. Intrusion detection and fraud detection are rarely addressed with text mining approaches.

6. Ontologies prove to be helpful in sentiment classification and therefore they need to be employed in future studies in financial domain.

7. Social media analytics is also playing an important role in the financial sector. Hence, this provides scope to explore this domain.

Chapter 3

Binary Classification of Text Documents

Document classification is a significant and a challenging problem. This chapter presents a novel binary classifier for text documents. It begins by stating the significance of the document classification problem and then outlines its applications. Later, this chapter presents the proposed model. Finally, the results of the model evaluations are discussed.

3.1 Introduction

The real world is replete with textual information, and hence text mining is believed to have a commercial potential higher than that of data mining [155]. In fact, a recent study revealed that 80% of the available data is not in the structured form. Text mining is used for converting the text into a structured format which then becomes an input for standard data mining algorithms. However it is a much more complex task (than data mining) as it involves dealing with text data that are inherently unstructured and sometimes fuzzy.

3.2 Motivation and Contributions

Text categorization or document classification is the task of assigning a document to a predefined class. Binary document classification is a type of text categorization in which each document in the corpus is labeled by one of the two predefined categories. This task is accomplished by using the knowledge extracted from pre-classified documents. The challenging part of document classification is the identification of discriminative features. This is because text documents contain a large number of features, which are always huge in number when compared to documents. So, picking up discriminative features is a difficult exercise. Contributions of this chapter are:

1. The efficiency of various Feature Subset Selection methods, namely Gini Index, t-statistic, Chi-square, and Correlation in finding discriminative features for document classification is identified.

2. Various classification algorithms like Decision Trees (DT), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes (NB), Multilayer Perceptron (MLP), Group Method Data Handling (GMDH), Probabilistic Neural Network (PNN), and Repeated Incremental Pruning to Produce Error Reduction (RIPPER) were employed and the best performing classifier is identified.

3.3 Related Work

Classification of text documents involves dealing with a large number of features as each distinct word in a document could be considered as a feature. Various statistical and machine learning techniques have been applied to text categorization. Below are some notable works of document classification in the literature: Willett [156] explored the use of hierarchical agglomerative clustering methods for document retrieval. Apte and Damerau [157] performed extensive experiments using optimized, rule-based induction methods on large document collections. The goal of these methods is to discover the patterns that are used for general document categorization or personalized filtering of text.

Guthrie and Walker [158] presented the theory and practice of determining the best category for a given document. They described a mathematical model using multinomial probability for classification schemes. They also provided the experimental results for evaluation of the method. They experimented on the following two datasets: 1000 samples collected from DARPA TIPSTER and 1100 texts from Message Understanding Conference Texts. Using a combination of classifiers, a model for document classification in the medical domain was proposed by Larkey and Croft [159]. They evaluated the inpatient discharge summaries with three classifiers - k-NN, Relevance feedback, and Bayesian independence. Later, they analyzed the results with the combination of these classifiers. They concluded that the combination of classifiers improved the accuracy of classification as compared to each of the classifiers applied individually.

Yang and Pedersen [160] performed a comparative study of feature selection methods for text categorization. Five methods were evaluated, including term selection based on Document Frequency (DF), Information Gain (IG), Mutual Information(MI), a Chi-square(Chi) test, and Term Strength(TS). They found that IG and Chi are the most effective. Joachims [161] explored the use of SVM for text classification. Experimental results show that SVMs achieve substantial improvements over the best-performing methods and are robust as compared to a variety of different learning methods. In this study, they used the Reuters subset collection. Dumais et al. [162] compared the effectiveness of five different (Find Similar, Decision Trees, Naïve Bayes, Bayes Nets, Support Vector Machines) automatic learning algorithms for text categorization. McCallum and Nigam [163] presented a method for text classification based on Bernoulli multinomial and multivariate models. They experimented with these two models on five data sets. They concluded that multivariate Bernoulli method performed well on a small corpus, whereas another (multinomial) model performed better on large data sets. Dorre et al. [4] explained the differences between text and data mining. They described in detail about the technologies which are essential for text mining. Yang [164] carried out a comparative evaluation of text categorization methods. They experimented with three classifiers viz., k-NN, Linear Least Square Fit (LLSF) and NB on the Reuters collection.

Gabrilovich and Markovitch [165] proposed a model for document classification. They analyzed the effect of redundant features on various classifiers. They applied Information Gain (IG), Chi-Square, Odds Ratio (OR), Document Frequency (DF), Bi-Normal Separation (BNS) methods to remove the irrelevant features. Classification task was performed by SVM, DT and k-NN algorithms. They experimented on TechTC repository. To know the effect of relevant features they carried out the experiments with reduced, as well as full features. They reported that SVM performed well with optimal features (i.e. 85.3% of accuracy) as compared to the other two algorithms. Huang et al. [51] reported research on financial market analysis using financial news headlines. Murthy and Murthy [7] performed text document classification using discriminative feature analysis. In this work, they used feature selection methods like Information Gain, Mutual Information, Fisher Score, Chi-Square and Document Frequency. After selecting the important features, they constructed models using SVM, k-NN, DT, NB and RF. Recommendations help the companies develop their social media competitive analysis strategies. Based on this idea He et al. [72] presented a case study in which text mining is applied to analyze unstructured text content referring to the three large pizza chains: Pizza Hut, Domino's Pizza and Papa John's Pizza, taken from Facebook and Twitter.

Gabrilovich and Markovitch [165] applied SVM, DT and k-NN models only on Tech TC repository. However, they did not apply neural networks which can classify documents very well. This motivated us to apply the neural networks to this problem. The various works are summarized in below Table 3.1.

3.4 Proposed Model

A new model for binary document classification is proposed here. It consists of the following phases: preprocessing, feature subset selection, and modeling. A schematic of the proposed model is depicted in Figure 3.1. Initially, a binary Document-Term Matrix was generated. Each row in the matrix represents a document, and each column represents a distinct word or feature. An entry of '1' in the column denotes the presence of the feature/ word in the document and '0' indicates its absence. Feature subset selection methods were applied next,

Table 3.1: Summarization of Related works on Document Classification

Study	Dataset	Model	Performance Measure
Willet [156]	——-	Survey article	———
Apte and Damerau [157]	Reuters	DT, NB, and Rule induction	Precision, Recall
Guthrie and Walker [158]	DARPA TIPSTER, Message Understanding Conference (MUC) texts	Multinomial Probability	Accuracy
Larkey and Croft [159]	Discharge summaries of hospital	k-NN, Bayesian independence, Relevance feedback, and combination of these models	Accuracy
Yang and Pedersen [160]	Reuters, OHSUMED collection	k-NN, Linear Least Squares FIT (LLSF)	Precision
Joachims [161]	Reuters	SVM, NB, C4.5, k-NN, and Rocchio	Precision, Recall
Dumais et al. [162]	Reuters	DT, NB, SVM, and Bayes Net	Accuracy
Gabrilovich and Markovitch [165]	TechTC	SVM, C4.5, and KNN	Accuracy
McCallum and Nigam [163]	20NG, Market guide, Yahoo web pages	Bernoulli multinomial and multivariate	Precision, Recall
Dorre et al. [4]	Review article	IBM Intelligent Miner	——-
Yang and Liu [164]	Reuters	NB, k-NN, LLSF, and NN	F-Score
Chinta and Murthy [7]	20NG	SVM, k-NN, DT, NB, and RF	Accuracy
He et al. [72]	Twitter and Face book comments	Case study	—-

to remove the irrelevant features and then classifiers are invoked on the reduced document term matrix.

The proposed methodology employing feature selection methods and classifiers, is described in the following Algorithm 3.1

Algorithm 3.1 Binary Classiifcation

Input: Text documents

Output: Classification of the documents

1: Do Text Preprocessing on text documents
2: Form the *Binary Document Term Matrix*
3: Apply t-statistic/ Gini Index/ Chi-square/ Correlation and select top 'n' features, this will decrease dimensions in DTM
4: Divide the data into 10 folds for 10-FCV
5: Repeat the steps 6 through 10 for all folds
6: Perform classification using k-NN, DT, SVM, MLP, NB, and other models document term matrix
7: Compute the average values of Accuracy over 10 folds.

3.5 Data, Techniques and Measures Used

3.5.1 Dataset Used

The dataset was acquired from the TechTC repository [166] which is publicly available. These datasets were created as part of Open Directory Project (ODP). It contains 100 labeled datasets, each having positive and negative class samples. All positive documents were combined into one group and negative documents were combined into another group.

3.5.2 Tools and Techniques Used

The proposed model was implemented on a machine running Windows 8 OS. The system has 8GB RAM and a 500 GB hard disk. *RapidMiner*® [167] was employed for text preprocessing and feature selection. Similarly, other tools ($KNIME$®

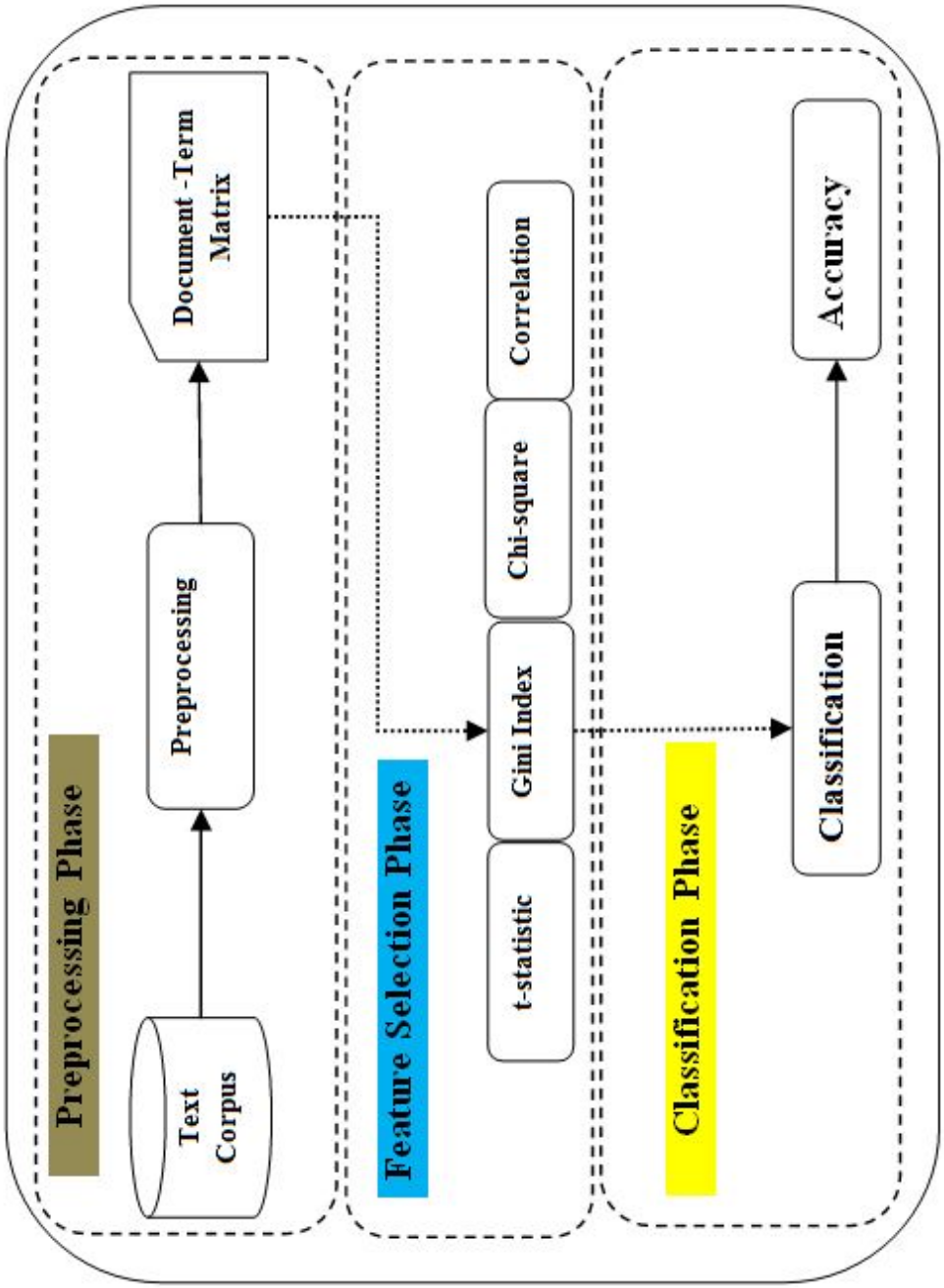

Figure 3.1: Schematic Diagram of Proposed Methodology

Table 3.2: Tools and Techniques Used

Technique Used	**Used for**	**Tools**
Preprocessing	Document Term Matrix	*RapidMiner*®
Gini Index/ Chi-square/ Correlation	Feature Selection	*RapidMiner*®
t-statistic	Feature Selection	*Computed_in_Excel*
DT/ SVM/ KNN	Classification	*KNIME*®
NB/ RIPPER	Classification	*RapidMiner*®
GMDH/ PNN/ MLP	Classification	*Neuroshell*®

[168] and *Neuroshell*® [169] employed for classification are also mentioned in the Table 3.2.

3.5.3 Feature Selection Methods

Feature selection serves two primary purposes. Firstly, it makes training and the process of applying a classifier more efficient by decreasing the size of the vocabulary. Secondly, feature selection often increases classification accuracy by eliminating noisy features. In this study, the following feature selection methods were used:

1. **t-statistic:**

   Features having a higher t-statistic value have more discriminative power[170]. Hence, we calculated the t-static values for each feature and selected the top features.

$$t = \frac{|\mu_1 - \mu_2|}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \tag{3.1}$$

   Where, μ_1 and μ_2 represent the mean values of a feature for two different classes, σ_1 and σ_2 represent the corresponding standard deviations for each class and n_1 & n_2 represent number of samples in each class.

2. **Gini Index:** It measures the impurity of a variable [171]. Features having a higher Gini index value have a higher impact on classification. It is calculated as follows:

$$GiniIndex = 1 - \sum_{i=1}^{n} p_i^2 \tag{3.2}$$

3. **Chi-square:** It estimates the mean of a normally distributed population and of estimating the slope of a regression line. The Chi-squared distribution with k degrees of freedom is the distribution of a sum of the squares of k independent standard normal random variables [172].

$$\chi^2(f_i) = \sum_{i \in \{0,1\}} \sum_{j \in \{0,1\}} \frac{(v_{ij} - e_{ij})^2}{e_{ij}} \tag{3.3}$$

Where,

$$e_{ij} = \frac{\sum_{l \in \{0,1\}} v_{il} \sum_{j \in \{0,1\}} v_{lj}}{n}$$

Here, v_{ij} indicates value of the number of examples from class c_i having feature value f_j, the number of examples from class c_1 is n_1 and that from class c_0 is n_0 and total number of examples is n.

4. **Correlation:** It defines the weight of attributes with respect to the label attribute by using Correlation [173]. The higher the weight of an attribute, the more relevant it is said to be.

$$Correl(x, y) = \frac{1}{n-1} \sum_{i=1}^{n} \frac{(x_i - \bar{x_i})}{\sigma_{x_i}} * \frac{(y_i - \bar{y_i})}{\sigma_{y_i}} \tag{3.4}$$

Where, x and y are two attributes, with $\bar{x_i}$, $\bar{y_i}$ means and σ_{x_i}, σ_{y_i} are standard deviations of respectively.

3.5.4 Performance Measures Used

Accuracy was used as the performance metric for the evaluation of the proposed model.

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{3.5}$$

Where, TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative.

3.6 Results and Discussion

After preprocessing the text, 13725 unique features were obtained; this makes the data very high dimensional. Four feature selection methods were applied: t-statistic, Correlation, Gini index and Chi-square. Then, the top 0.5% (Top-69) features were considered for constructing the Binary Document Term Matrix and building the model. The same procedure was repeated for top 0.1% (Top-15) features also. Throughout this work, 10 Fold Cross Validation (10 FCV) was performed for all the classifiers and average accuracies are reported in Table 3.5. It was fascinating to observe that all the top-15, as well as top-69 features obtained by various feature subset selection methods, were all identical. It shows that all the features chosen by the feature subset selection methods are very reliable in identifying the appropriate class and showcases the reliability of these techniques. The selected features are listed in the Tables 3.3 and 3.4.

It is observed that Neural Network models (GMDH, PNN) and RIPPER performed better as compared to other algorithms. The accuracies of various models developed using k-NN/ DT/ NB/ SVM/ GMDH/ MLP/ PNN/ RIPPER are presented in Table 3.5. It is observed that GMDH produced the highest accuracy of 94.% with top-69 features followed by GMDH and PNN with top-15 features. To validate the trustworthiness of the results obtained by different models, statistical significance test (t-test) was conducted between various models and the corresponding results are presented in Table 3.6. All t-test values among different models are below 2.83, which is the t-test value at 1% level of significance and 18 degrees of freedom (10+10-2=18). Furthermore, t-test was performed across

Table 3.3: Top-15 Features of TechTC Dataset

Top-15 Features				
behavior	freshman	riser	evict	enrol
jordan	digest	saskatchewan	bacon	damp
spike	uplift	spinner	fulton	shame

various models between top 0.1%and top 0.5% feature subset sizes. These values are presented in Table 3.7; all the t-test values are below the table value of 2.83 and hence these values once again statistically prove our models. Based on the t-values, models with 0.1% of features and models with 0.5% features are statistically same at 1% of significance.

GMDH is an abductive network. It selects the optimal predictors from the features set by repeatedly using the Ivakhnenko polynomials, linear regression and a special neural network architecture. Therefore, that it produces accurate results in classification as well as regression tasks. PNN is one of the Neural Network models and its working principle is based on Bayes classifier and has one-pass learning neural algorithm. PNN is extremely fast due to its parallelism of networks, and is very accurate.

3.7 Conclusions

In this chapter, a model for binary document classification is proposed. Four feature subset selection methods were employed viz., Chi-square, Correlation, t-statistic, and Gini Index followed by various classifiers namely, GMDH/ RIPPER/ MLP/ NB/ DT/ SVM/ PNN/ KNN. Models were built using 0.5% and 0.1% features separately. Based on the statistical significance test i.e. t-test values between models corresponding to datasets with 0.1% features and 0.5% features, it is observed that there is not much statistical difference between the results. Therefore, it is concluded that the model built using 0.1% features should be preferable for further analysis, because of performance gains. Further, since the difference in PNN and other NN models is not statistically significant, it is recommended due to its active learning.

Table 3.4: Top-69 Features of TechTC Dataset

Top-69 Features				
Behavior	shame	ditch	fellow	sander
Freshman	matthew	bulgaria	sorbet	abel
Riser	government	oneida	rondo	infract
Evict	toilet	tequila	spar	alhambra
enroll	pill	undertake	punch	coarse
Jordan	veteran	berg	damn	hyphen
Digest	compress	pitt	midwestern	newsprint
saskatchewan	bouncer	sixteen	squeal	tonsil
Bacon	rede	reject	tart	princeton
damp	robertson	gradual	thriller	plow
spike	heirloom	polio	tenderloin	wick
uplift	wheeler	bylaw	laptop	cello
spinner	eucharist	former	warn	detour
fulton	symptom	panther	junk	

Table 3.5: Average Accuracy of Models Using Top 0.5% and Top 0.1% of the Total Features

Method	**Gabrilovich and Markovitch [165] (with 0.5% features)**	**Proposed Model Result (with 0.5% features)**	**Proposed Model Result (with 0.1% features)**
KNN	82.7	80.5	72.56
DT	84.3	81.5	79.2
SVM	85.3	76	82.5
MLP	NA	88*	87*
GMDH	NA	94.5*	93*
PNN	NA	91*	92*
RIPPER	NA	87*	85.5*
NB	NA	80	79

NA=Not Available, *=Best Values

Table 3.6: t-test Values of Various Models

Models	t-test value (0.5% features)	t-test value (0.1% features)
GMDH - MLP	2.276	1.819
MLP - PNN	0.871	1.539
GMDH - PNN	1.291	0.397
RIPPER - MLP	0.212	0.621
RIPPER - GMDH	2.812	2.628
PNN - RIPPER	1.301	2.361

Table 3.7: t-test Values of Models on Feature Subset Selection

Models	t-test value
MLP - MLP	0.31625
GMDH - GMDH	0.61667
PNN - PNN	0.57988
RIPPER - RIPPER	0.49319

Chapter 4

One-Class Classification of Text Documents

This chapter presents a hybrid model for one-class document classification. First, the motivation for the work is stated, followed by the contributions made. Later, the chapter describes the proposed model and then, analysis of the results is presented.

4.1 Introduction

Text mining has several applications, including document classification [157], social network analysis [174], sentiment analysis, outlier detection, customer migration behavior [175], fraud detection in banking and insurance [176], churn prediction [175]. This chapter focuses on the one class classification of text documents.

4.2 Problem Statement

Let there be N documents $D = \{d_1, d_2, ..., d_N\}$. Each document d_i is associated with only one class c. The problem is to classify new documents belong to c or not, using a classifier M. Here, M chosen is OCSVM.

4.3 Motivation and Contributions

In this work, the objective is to develop one-class classification models for text classification. This technique involves training the classifier with samples representing a well-defined class and using this model to detect the outlier class which is the positive class. This approach is very useful in cases where the class to be identified is not well described. Hence, modeling is done using only target class samples. Contributions of this chapter are:

1. A novel hybrid method combining OCSVM and LSI for document classification preceded by t-static based feature subset selection method is proposed.

4.4 Related Work

In this section, previous works related to text mining are presented. We briefly review the work reported in applying text mining to phishing detection and other applications. Phishing detection and protection are challenging areas for text mining. A Multi-label Classifier based Associative Classification (MCAC) was proposed for phishing detection by Abdelhamid et al. [177]. They experimented on 2100 websites with C4.5, JRip classifiers. Apart from these, other techniques viz., Blacklist based, fuzzy rule-based, machine learning techniques, CANTINA were also discussed in their work. Phishing detection using data mining techniques was carried out in the works of Abu-Nimeh et al. [81]. They experimented with a corpus size of 2889 emails (with the proportion of 59.5% legitimate, 40.5% phishing) and used 43 features. They employed different techniques including, Logistic Regression (LR), Naive Bayes (NB), Support Vector Machine (SVM), Random Forest (RF). In another research work carried out by Garera et al. [84] some of the key features (page-based, domain-based, type-based and word-based) of phishing were identified from URLs. They employed LR for model construction.

A method to identify features to detect the phishing web sites was proposed by Ludl et al. [83] who analyzed 1000 sites with two approaches (Blacklist and source analysis) and employed C4.5 classifier with eighteen prominent features. Phishing attacks in relation to risk levels and the loss of market value were assessed by Chen

et al. [93]. They analyzed 1030 phishing alerts of a public database using a hybrid method. They employed the Decision Tree (DT), Support Vector Machine (SVM), and Neural Network (NN) models for classification and reported a prediction accuracy of 89%. The task of classifying web pages into legitimate or phishing was carried out by He et al. [92]. They extracted 12 features from a web page and trained an SVM. They reported a true positive value of 97% and a false negative value of 4%.

Pandey and Ravi [178] employed various classifiers such as Genetic Programming (GP)/ Multilayer Perceptron (MLP)/ DT/ Group Method Data Handling (GMDH)/ SVM/ LR/ Probabilistic Neural Network (PNN) for phishing detection. Recently, Pandey and Ravi [95] performed the detection of phishing websites and spam using text mining. In this study, they employed various classifiers viz., GP, LR, PNN, MLP, Classification And Regression Tree (CART), GP+CART and reported high sensitivity values. In both the studies (i.e. phishing email and websites) they reported high sensitivity value for GP. They also performed the statistical significance test for these experiments. Malware identification was performed by Lee and Stolofo [179] using Data Mining techniques. They extracted some of the features (system calls) for identifying the intrusion. Through the concept of Association Rule Mining, they found frequent system calls and generated the rule set.

Intelligent Malware Detection System (IMDS) using Association Rule Mining was proposed by Ye et al. [115], and they compared it with Naive Bayes (NB), SVM and J4.8 classifiers. The accuracies of these models were reported to be 83.86%, 90.54% and 91.49% respectively and 93.07% with proposed approach (IMDS). Composite malware detection scheme using run-time API (Application Program Interface) calls of Windows was introduced in the works of Ahmed et al. [117]. They selected 237 API calls of both benign and malware programs and categorized them into seven groups based on their functionality. They reported a detection rate of 0.97. Malware detection using API calls through text mining was performed using Mutual Information as a feature selection method by Sundarkumar and Ravi [121]. They oversampled the minority class samples and performed classification using DT/ SVM/ MLP/ PNN/ GMDH/ OCSVM classi-

fiers under 10-Fold Cross-Validation (10-FCV). Finally, they reported sensitivity values of 100% with OCSVM.

Latent Semantic Indexing (LSI) has been applied in various works in the past. MLP and Singular Valued Decomposition (SVD) method [180] were used for document classification on a subset of Reuter-21578 dataset. Clustering of text documents by combining LSI and Genetic Algorithm (GA) was reported in the works of Song and Park [181]. The proposed model was evaluated on Reuter-21578 and they concluded that an optimal number of clusters are formed with this approach. Recently, classification of research projects based on technologies of research funding organizations was done by Thorleuchter and Van den Poel [182]. Technologies that occur together are grouped into classes by LSI. Textual patterns are representatives for each class, and projects are assigned to these classes. This enables the assignment of each project to all technologies grouped by LSI.

In literature, most of the works approached the classification task from a binary perspective; very few approached it from a one-class perspective. Hence, an attempt has been made through this work, to address this gap. In this work, row-dimension reduction rather than column-dimension reduction is performed with the help of support vectors of one-class SVM and a classifier is constructed using LSI. Few of the existingworks experimented with oversampling method (adopted when minority (positive) class samples are less as compared to majority (negative) class samples). Sundarkumar and Ravi [121] achieved a sensitivity value of 100% using oversampling technique. Pandey and Ravi [[178], [95]] experimented with 17 features and achieved the sensitivity of 98%. However, achieving the same or near sensitivity without using the oversampling techniques and with fewer features, is more desirable. With this motivation, a novel hybrid one-class classification model is proposed in this study. The various works are summarized in below Table 4.1.

4.5 Proposed Hybrid Model

In the proposed method, it is assumed that historical data of other classes is not available. The proposed methodology mainly consists of three phases. The first phase is preprocessing, the second phase is the training phase, and the third phase

Table 4.1: Summarization of Related works on Document Classification

Study	Dataset	Model	Performance Measure
Abdelhamid et al. [177]	Millersmiles/ PhishTank	Associative Rule	Accuracy
Abu-Nimeh et al. [81]	Monkey.org/ Spambase	LR/ CART/ SVM/ NN/ BART/ RF	Error Rate, False Positive
Garera et al. [84]	Google safe browsing toolbar	LR	Accuracy
Ludl et al. [83]	PhishTank	C 4.5	Accuracy
Chen et al. [93]	Millersmiles	DT/ NN/ SVM	Accuracy
He et al. [92]	Millersmiles\PhishTank\3Sharp	SVM	True Positive, False Positive
Pandey and Ravi [178]	PhishTank/ SpamAssasin	GP/ MLP/ GMDH/ SVM/ LR/ PNN	Sensitivity, Accuracy
Pandey and Ravi [95]	PhishTank/ SpamAssasin	GP/ LR/ PNN/ PNN/ CART/ GP+CART	Sensitivity
Lee and Stolofo [179]	tcpdump data	Association Rule	Accuracy
Ye et al. [115]	API Calls	ARM, NB, SVM and J4.8	Accuracy
Ahmed et al. [117]	API Calls	DT	Accuracy
Sundarkumar and Ravi [121]	Windows API Calls	DT/ SVM/ MLP/ PNN/ GMDH/ OCSVM	Sensitivity
Song and Park [181]	Reuter	LSI + GA	F-Score
Thorleuchter and Van den Poel [182]	Techology projects	LSI	Precision, Recall

is the test phase. The proposed methodology is depicted in Figure 4.1. Firstly, all the samples from various types of datasets were collected(i.e. Phishing/ Malware.) and text preprocessing was performed. After the preprocessing (i.e. tokenization, stop words removal, stemming) step was completed, the processed data was mapped into an intermediate form, popularly called document-term matrix. Later, feature selection was applied to remove the unnecessary features. In the second phase, OCSVM was used for the extraction of support vectors from the majority class. As it was assumed that the history (characteristics) of any Phishing Email/ Website is not known, the most important documents i.e. support vectors of negative class (i.e. genuine/ benign/ legitimate) were extracted from the document-term matrix using OCSVM. After extraction of these influential support vectors, the LSI technique was applied in the third and final phase which is the test phase. In this test phase, first the support vector matrix was decomposed into three matrices (U, S, V) using Singular Valued Decomposition (SVD). Later, the new document vector co-ordinates (which consist of Eigenvalues) were calculated in the new dimensions . These co-ordinates are the individual support vectors/ document vectors. After this step, the query vector for each upcoming query/ document was found in the positive class. After that, the cosine similarity between query documents and training documents was calculated. After computation of similarity score (average), we set the threshold to even chance i.e. greater than 0.5 value. Then, classification task was performed based on this value. That is, if the similarity score is greater than or equal to 0.5 then, it was assigned to the positive class, otherwise it was assigned to the negative class.

The proposed methodology employing LSI and OCSVM, is described in the Algorithm 4.1.

4.6 Data, Techniques and Measures Used

4.6.1 Datasets Description

Various datasets were used for evaluating the effectiveness of our proposed method; a description of these datasets is given below: Phishing e-mail dataset consists of a total of 2500 emails out of which, 1240 are legitimate (Spam Assassin

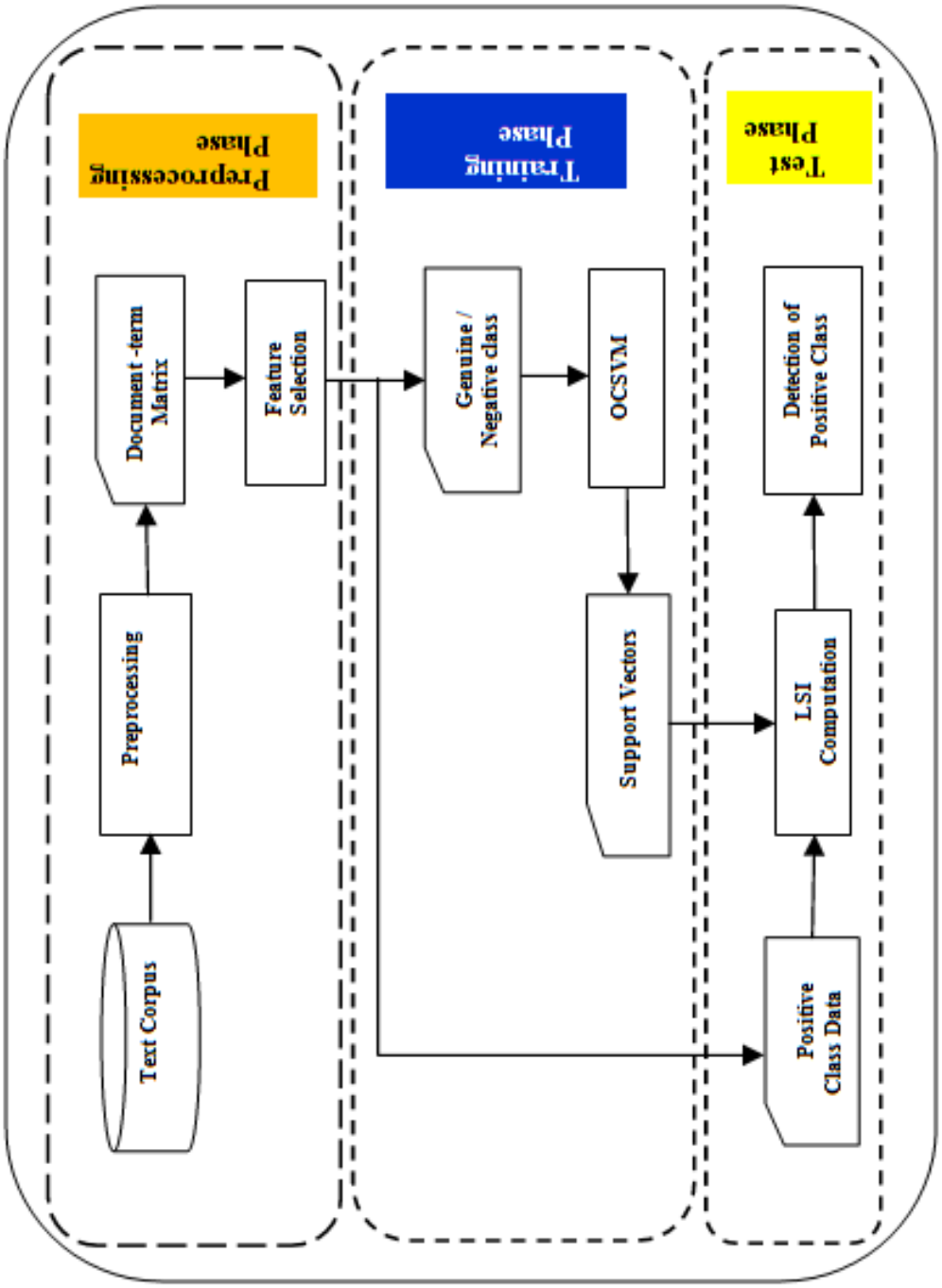

Figure 4.1: Schematic Diagram of Proposed Methodology

Algorithm 4.1 OCSVM_LSI

Input: Text documents

Output: classification

1: Do Text Preprocessing on text documents
2: Form the *Binary Document Term Matrix*
3: Apply t-statistic method and select top 'i' features, this will decrease dimensions in DTM
4: Extract the negative and positive class data
5: Train the OCSVM with negative class data and extract the support vectors
6: Compute the LSI using support vectors and positive class data
7: Compute the average values of sensitivity.

[183]) and 1260 are phishing examples [184]. The body of the emails was used for extracting the features as mentioned in Pandey and Ravi [178]. Phishing website dataset consists of 200 URLs from Phish Tank [185] among which 100 URLs are phishing websites, and the remaining are legitimate sites. This was accomplished by extracting source code from the URLs as in Pandey and Ravi [95]). Imperial Bank dataset comprises the feedback of 786 customers consisting of 132 satisfied customers, 148 very satisfied, 166 unsatisfied, 172 very unsatisfied, and 168 neutral customers. The satisfied and very satisfied customers were combinedinto one class and similarly unsatisfied and very unsatisfied were combinedinto another class. The neutral comments were ignored. After combining, class-1 consisted of 280 documents (feedbacks) and class-2 consisted of 338 documents. Then binary classification task was performed on these two classes. The dataset was obtained from IBM [186]. Malware corpus [187] consists of 388 logs (API calls) out of which, 320 Malware traces were labeled as '1' and 68 benign software traces were labeled as '0'

4.6.2 Tools and Techniques Employed

A machine with Intel i5 processor, 2.6GHZ, 8GB RAM, 500 GB HDD and 64-bit Windows 8 OS was used for conducting the experiments. Tools employed for this work are described in the Table 4.2. For text preprocessing, *RapidMiner*® [167]

Table 4.2: Tools and Techniques Used

Technique Used	Used for	Tools
Preprocessing	Document Term Matrix	*RapidMiner*®
OCSVM	Support Vectors	*LIBSVM*®
LSI	Classification	*MATLAB*®

was used, while LSI computation was done in *MATLAB*® [188].

4.6.3 Fetaure Subset Selection Method

Feature selection serves two primary purposes. First, it makes training and testing more efficient by decreasing the size of the vocabulary. Second, feature selection often increases classification accuracy by eliminating noisy features. In this study, the t-statistic based feature selection method was used.

t-statistic: Features with higher t-statistic value have more discriminative power [170]. Hence, the t-static values for each feature were calculated and the top features were selected according to the values of the t-statistic.

$$t = \frac{|\mu_1 - \mu_2|}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \tag{4.1}$$

4.6.4 Performance Measures

Sensitivity/ True Positive rate was used as the performance measure for the evaluation of the proposed model. The goal is to train the classifier with the samples of one class and test it with those of the other class.

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

Where, TP = True Positive and FN = False Negative.

4.7 Results and Discussion

Table number 4.3 summarizes the number of features extracted from the datasets. Here, the goal is to detect Phishing emails or Phishing websites or Malware, which are defined as the positive class in the proposed approach. Therefore, a higher sensitivity value implies a greater detection of fraudulent events/ mails.

Table 4.3: Features Extracted from Various Datasets

Dataset	Features extracted
Malware	313
Imperial Bank	216
Phishing websites	17
Phishing Email	42

The features from the phishing website dataset which are mentioned in the work of He et al. [92] were extracted. These features are reported in Table 4.4. Similarly, we extracted 23 influential features from the phishing email corpus. Then, we performed the classification task with these features. In Table 4.5 the feature subset of the phishing email corpus is listed. The total number of features extracted from Malware dataset is 313. From these features, a subset was selected based on the t-statistic based feature selection technique. There were 11 features with a t-statistic value greater than 5 and these were chosen. These are presented in Table 4.6. For all datasets, after construction of the document term matrix and performing feature selection, the support vectors were extracted from negative class examples using OCSVM [189]. The number of support vectors obtained for various datasets is as follows: 621 for phishing emails, 54 in phishing websites (with feature selection) and 56 (without feature selection). After extracting the support vectors for each dataset, LSI was invoked. Results obtained by our method are reported in Table 4.7. Support vectors were extracted with all kernels (Linear/ Polynomial/ Sigmoid/ RBF). Experiments were conducted with these kernels, and sensitivity values obtained in comparison to those reported by the previous works, are reported. First the results of the Imperial Bank dataset are discussed; sensitivity values are reported separately for each kernel. The

Table 4.4: Features Extracted from Phishing Website Dataset

Features		
search engine	cookie	whois lookup
foreign requesturl in id	slash in page add	nil anchor
foreign anchor in id set	slash in url	server form handler
ssl certificate	foreign request	ip address
blacklist	using @ symbol	dots in url
foreign anchor	dots in page add	

Table 4.5: Features Extracted from Phishing Email Dataset

Features			
account	price	response	system
member	market	offer	process
access	online	transaction	service
email	information	agreement	request
address	work	registration	message
update	credit	person	

sensitivity values obtained are as follows: 66.44% (linear)/ 81.3% (polynomial)/ 99.11% (RBF) and 66.5% (sigmoid). For phishing emails dataset the sensitivity values obtained are 100% (linear)/ 93.19% (polynomial)/ 99.83%(RBF)/ 99.67% (sigmoid). Similarly for Phishing websites, without feature selection the sensitivity values obtained are: 92.73% (linear)/ 98.21% (polynomial)/ 96.11% (RBF)/ 92.73% (sigmoid). The sensitivity for the same dataset with feature selection (i.e. with 9 features) turned out to be 88% (linear)/ 89% (polynomial)/ 100% (RBF)/ 88 (sigmoid). Similarly, for Malware dataset we obtained the sensitivity values of 94.06% (linear)/ 88.12% (polynomial)/ 91.25% (RBF)/ 96.87%(sigmoid).

The above table shows the results obtained by the proposed method. For Imperial Bank dataset, an accuracy of 99.11% was achieved, for Phishing Email 100%, for phishing websites 98.07% was achieved without feature selection and

Table 4.6: Feature Subset of Malware Dataset

Features		
GetFileAttributesW	GetDriveTypeW	ExitProcess
GetFullPathNameW	NtOpenKey	GetLongPathNameW
GetEnvironmentVariableW	RegSetValueExW	NtQueryValueKey
SearchPathW	RegOpenKeyW	

Table 4.7: Sensitivity Values of the Proposed Model with Various Datasets

Dataset	Sensitivity	
	Other Approach	Proposed Approach
Imperail Bank	NA	99.11*
Phishing Email	97.29 [178]	100*
Phishing Websites	98 [95]	100*
Malware (11 features)	100 [121]	96.87

NA=Not Available, *=Best Values

100% with feature selection, and similarly with Malware dataset an accuracy of 96.61% was achieved. These results are good with respect to accuracy/ sensitivity.

4.8 Conclusions

Different approaches exist for classifying the text documents. In this chapter, a new methodology for text document classification using OCSVM and Latent Semantic Indexing in tandem, is proposed. Primarily, support vectors from negative classwere extracted and LSI based classification was computed. This method was applied to various datasets like Phishing Email and Website corpus, Malware and Imperial Bank dataset. Through this, more than 99% of sensitivity was achieved on most of the datasets. Results are superior to the previous approaches reported in the literature.

Chapter 5

Text Document Classification with PCA and One-Class SVM

This chapter presents a hybrid model for document classification which involves documents belonging to one-class. In what follows, the motivation behind this work and contributions made are explained. Later, the proposed model is described and the results are analyzed.

5.1 Introduction

Text classification task involves several challenges, including high dimensionality of the feature space, where each unique word represents a feature [190]. It is essential to reduce the dimensions of the documents as most of the textual data will have sparseness when it is mapped into a structured format. Here, it is intended to reduce the dimensions of the feature space, without compromising the performance of a classifier.

5.2 Motivation and Contributions

In general, binary classifiers train on samples from both classes (positive and negative). In reality, some of the datasets consist of samples from only one class. It indicates that we do not have prior knowledge of other class patterns. Hence, it is

challenging to build a classification model with the samples of the available class. This motivated us to build a text one-class classification model that is built based on the samples of one-class, usually negative class and test it on positive class, which is the minority class. Therefore, we chose to employ OCSVM for one-class classification In this chapter, we wanted to study the influence of dimensionality reduction on the classifier. For this purpose we preferred to employ PCA, a popular dimension reduction technique. Accordingly, this chapter proposes a hybrid of PCA (for performing dimensionality reduction) and OCSVM (for performing one-class classification) in tandem.

Contributions of this chapter are:

1. A novel hybrid model using PCA and OCSVM is proposed for document classification.

2. Dimensionality reduction using Principal Component Analysis (PCA).

5.3 Related Work

In this section, the past works performed in text mining, one-class SVM and finally Principal Component Analysis are discussed. Text document classification is the primary task for retrieval and summarization of text documents. Text categorization was initially performed by Maron [8] in 1961. In his research, he developed a mechanism for automatically categorizing the documents, based on their subjects. On the basis of occurrences of selected terms, the classification was performed. Masand et al. [191] proposed a method to classify the news stories with the help of memory based reasoning. They trained the model with approximately 50K stories from Dow Jones Press release News and reported a recall of about 80% and precision of about 70%. Later, Lewis and William [192] developed a sequential sampling algorithm for text classification task. Through this approach, they reduced the training time.

A comparative examination of feature subset selection methods for text document categorization was made by Yang and Pedersen [160]. They evaluated the model with five different feature selection methods: document frequency, mutual

information, information gain, Chi-square test and term strength, and concluded that features selected by information gain is performing well for classification among others. Dumais et al. defined that Text Categorization [162] is the assignment of natural language texts to one or more predefined categories based on their content. They explored the task of text classification with Naive Bayes, Decision Trees (DT), Support Vector machines (SVMs), and with two other approaches. Joachims [190] first experimented with SVMs for text categorization and identified the benefits of SVMs for text classification. Feature selection is done through Information Gain. They invoked the Bayes, Rocchio, C 4.5 (Decision Tree) and k-Nearest Neighbor (k-NN) classifiers for classification and they also provided the F-Score values of the experiments. McCallum and Nigam [163] approached the classification task with Naive Bayes (NB) classifier. They experimented with Multivariate Bernoulli model and multinomial model and concluded that the multivariate Bernoulli model performed well with few dimensions, whereas multinomial model performed better at huge dimensional data.

Taira and Haruno [193] investigated the effect of feature selection preceding SVMs for text classification. In their study, they found that SVM could handle large scale of dimensions. They also experimented with Decision Tree (C4.5) classifier for classification of text documents. Yang and Liu [164] performed text classification and provided the results of statistical significance tests on five methods, namely SVMs, k-NN, Neural Network, Linear Least squares fit and Naive Bayes. They experimented with a dataset containing over 300 instances. Similarly, Forman [194] carried out an experimental study on feature subset selection methods. He evaluated the model with twelve feature subset selection methods, including Information Gain, etc. on standard datasets. Results were analyzed based on the following measures: Accuracy, F-measure, precision, recall. Vert and Vert [195] discussed about the convergence criterion of One-Class SVMs.

Metsis et al. [103] explored the problem of Spam e-mail classification. They tested five different versions of Naive Bayes classifiers, like Multivariate Bernoulli, Multinomial NB with term frequency, Multinomial NB with binary attributes, Multivariate Gauss NB and Flexible Bayes. They experimented with datasets of 500, 1000, and 3000 features. The overall performance of the classifier was the highest with the 3000 features dataset. Phishing websites detection and the

effectiveness of the machine learning techniques in text mining was evaluated by Miyamoto et al. [86]. They employed nine techniques namely, NB, SVM, AdaBoost, Neural Networks, CART, Bayesian Additive Regression Trees, Random Forests, LR and Bagging in the study. The performance evaluation of these models was done using F1-Score and Area Under the Curve (AUC) plots. Lan et al. [196] investigated several supervised and unsupervised term weighting techniques on standard datasets with SVM and k-NN classifiers. Also, they proposed one new term weighting approach in their paper. Phishing email detection was carried out by Pandey and Ravi [178]. In this study, they constructed models with different classifiers such as, Multilayer Perceptron (MLP), DT, SVM, Group Method Data Handling (GMDH), Genetic Programming (GP), Probabilistic Neural Network (PNN) and LR. They collected 2500 emails and from these mails they extracted 23 important features. They also provided the results of statistical significance tests of these experiments.

Chinta and Murthy [7] analyzed various feature subset selection methods for categorizing the text documents. They found the minimal subset of features which were discriminative for document classification. Feature subsets were selected by information gain, Fisher Score, Chi-square statistic, mutual information and document frequency. Pandey and Ravi [95] performed phishing and spam detection using text and data mining. They extracted 17 features from source code of the URL. Sensitivity and accuracy values reported were higher than the previous experiments. In this study, they built various models with GP, LR, PNN, MLP, Classification And Regression Tree (CART), GP+CART and were able to report higher sensitivity values. Also, they provided the Decision Tree rules for classifying legitimate as well as phishing classes. Jun et al. [197] proposed a model to overcome the sparsity problem of document clustering. They combined dimension reduction techniques with the k-Means clustering algorithm, and experimented with patent documents which were retrieved from the United States patent office. Sundarkumar and Ravi [198] recently worked with OCSVM for data imbalance problem by employing k-Reverse Nearest Neighborhood algorithm. They tested their approach on customer credit card dataset (used for churn prediction) and Automobile Insurance dataset (used for fraud detection).

Principal Component Analysis (PCA) is one powerful technique to condense the number of dimensions. Detailed explanation on this method is provided in the next section. Performance of the model will depend on the chosen principal components; sometimes it may be improved by dropping a few components. Jolliffe [199] explained about the redundant components in PCA. Extracting significant components from the results of PCA was emphasized in Ferre [200]. Based on the objectives or constraints of the user, selection of principal components varies from one user to another since different criteria are adopted.

In this paragraph, we discuss the various works conducted on one-class classification problems. An essential task of web mining is web pages classification. This task can be accomplished by training a classifier with web pages from both, positive and negative class. However, collecting the data and preprocessing it would require a lot of time and effort. Collection of web pages in which the user is not interested is more difficult as compared to the positive samples (web pages in which the user is interested). To solve this problem Yu et al. [201] introduced a new framework called Positive Example Based Learning (PEBL) for negative pages collection. They presented amapping convergence criterion which enabled them to achieve a higher accuracy compared to the regular binary SVM classifier. They evaluated their model using the DMOZ and WebKB datasets.

Denis et al. [202] conducted research on one-class classification problems. They trained a Naive Bayes classifier with positive documents and evaluated it on the WebKB dataset. Use of linear functions to train on positive and unlabeled data was experimented with, by Lee and Liu [203]. They proposed a model using logistic regression with weighted samples and performance index. To evaluate the efficacy of the model, they tested it on the 20NG dataset. Manevitz and Yousef [204] worked on the one-class classification task for text classification. Initially, they trained a neural network with positive class data and they observed that this approach performs better than other standard methods. They generated a vector space model with tf-idf weighted scheme on the Reuters 21578 collection. Generally, in a binary classification task, classifiers are trained with two classes of examples (i.e. positive and negative). Unlike this traditional method Elkan and Noto [205] proposed a model which trains using only the positive samples. They applied this concept on a biological database. Some more noteworthy works

on one-class classification problems are present in the literature: Bostrom [206], Muggleton [207], Liu et al. [208], Manevitz and Yousef [209], Denis et al. [202].

The various works are summarized in below Table 5.1.

5.4 Proposed Hybrid Model

In the proposed method it is assumed that historical data of positive class is not available. The proposed methodology mainly consists of three phases. The first phase is preprocessing, second phase is dimensionality reduction phase, and the third phase is modeling. The proposed methodology is depicted in Figure 5.1. First, all the samples from various types of datasets (i.e. phishing/ Malware) were collected and text preprocessing was performed. After the preprocessing (i.e. tokenization, stop words removal, stemming) step was completed, the processed data was mapped into an intermediate form called document-term matrix. Later, PCA was applied for dimensionality reduction in the second phase. After this, the top principal components were selected such that the cumulative sum of the eigen values is greater than or equal to 0.5 ($\sum \lambda_i \geq 0.5$). After this, the negative and positive class samples corresponding to the selected principal components were extracted, thereby forming the row-dimension reduced matrix. From this matrix the rows corresponding to the negative samples were fed to the OCSVM for training in the third phase. Thereafter, the support vectors corresponding to the negative class were extracted. Later, the model was tested with positive class samples, i.e. classification was performed. Finally, the sensitivity values are reported.

The proposed methodology employing PCA and OCSVM, is described in the following Algorithm 5.1.

5.5 Data, Techniques and Measures Used

5.5.1 Datasets Description

Various datasets were used for evaluating the effectiveness of the proposed method; a description of these datasets is given in Table 5.2.

Table 5.1: Summarization of Related works on Document Classification

Study	Dataset	Model	Performance Measure
Maron [8]	Text documents	Automatic Probability indexing	Accuracy
Masand et al. [191]	News stories	Memory based reasoning	Precision, Recall
Lewis and William [192]	AP newswire	NB	Precision, Recall
Yang and Pedersen [160]	Reuters, OHSUMED collection	k-NN, Linear Least Squares Fit (LLSF)	Precision
Dumais et al. [162]	Reuters	DT, NB, SVM, and Bayes Net	Accuracy
Joachims [190]	OHSUMED collection	SVM	F-Score
McCallum and Nigam [163]	20NG, Market guide, Yahoo web pages	Bernoulli multinomial and multivariate	Precision, Recall
Taira and Haruno [193]	RWCP articles (news)	DT, SVM	Precision, Recall
Yang and Liu [164]	Reuters	NB, k-NN, LLSF, NN	F-Score
Manevitz and Yousef [204]	Reuters	OCSVM	Precision, Recall
Yu et al. [201]	DMOZ, WebKB	PEBL SVM	Precision, Recall
Forman [194]	Reuters, TREC, OHSUMED	SVM	Accuracy, Precision, Recall, F-Score
Vert and Vert [195]	———	Theoritical study	——
Metsis et al. [103]	SpamAssassin	NB	Recall, ROC
Miyomoto et al. [86]	Phishtank	NB, SVM, NN, CART, RF, LR, and AdaBoost	F-Score, AUC
Lan et al. [196]	Reuters, 20NG, OHSUMED	SVM, k-NN	F-Score
Pandey and Ravi [178]	PhishTank/ SpamAssasin	GP/ MLP/ GMDH/ SVM/ LR/ PNN	Sensitivity, Accuracy
Chinta and Murthy [7]	20NG, OHSUMED	SVM, k-NN, DT, NB, and RF	Accuracy
Pandey and Ravi [95]	PhishTank/ SpamAssasin	GP/ LR/ PNN/ PNN/ CART/ GP+CART	Sensitivity
Jun et al. [197]	United patent documents	PCA + K-Means, SVD	Silhouette value
Sundarkumar and Ravi [198]	Credit card datset, Automobile insurance data	k-RNN+OCSVM with Binary classifiers(DT/ SVM/ LR/ MLP/ GMDH)	Sensitivity, Specificity, Accuracy, AUC
Joliffe [199]	———-	PCA explaination	——
Ferre [200]	———	PCA explaination	——
Denis et al. [202]	WebKB	Postive Examples with NB	Accuracy
Lee and Liu [203]	20NG	LR	F-Score
Elkon and Noto [205]	SwissPort database	SVM	TP, FP

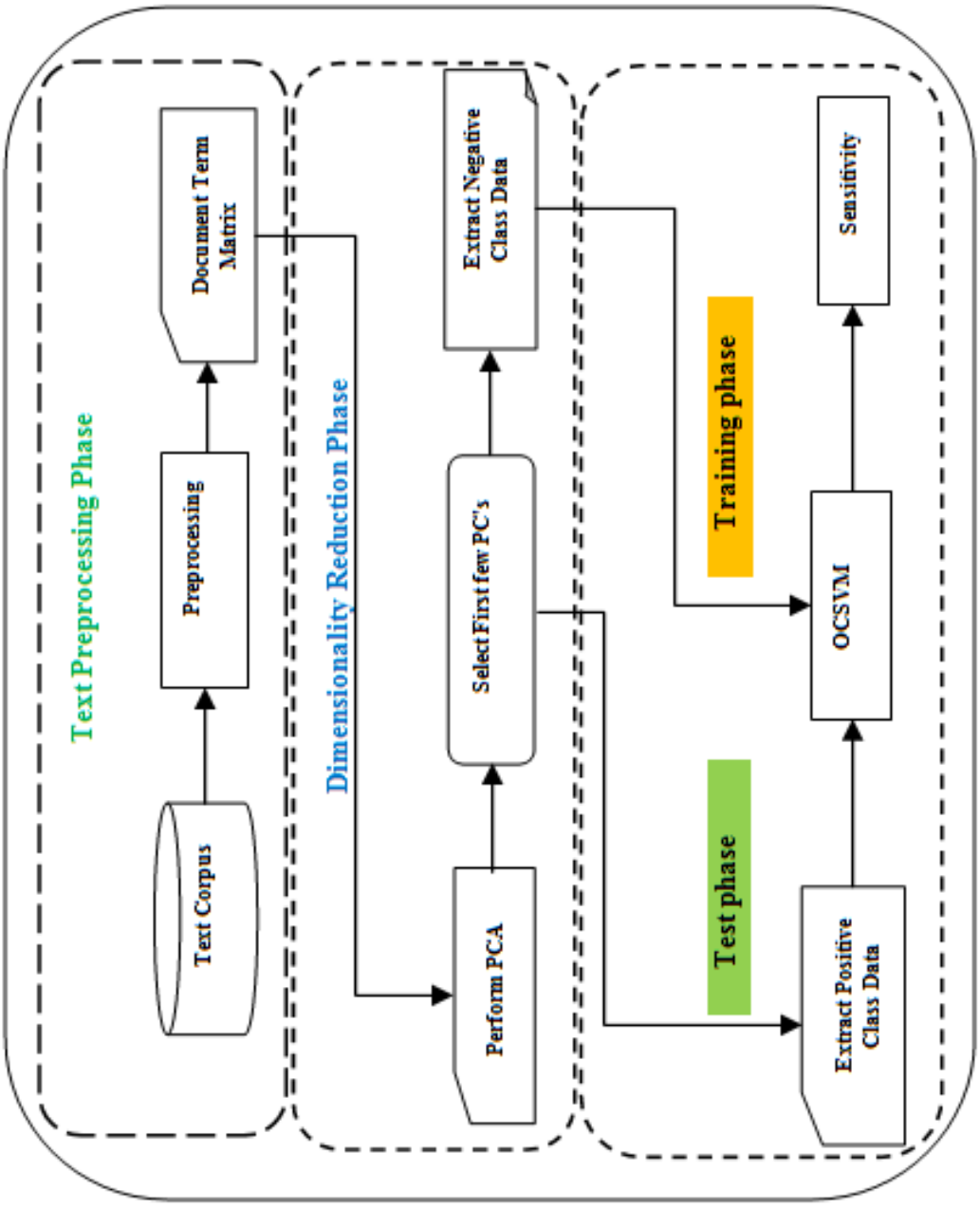

Figure 5.1: Schematic Diagram of Proposed Methodology

Algorithm 5.1 PCA_OCSVM

Input: Text documents

Output: classification

1: Do Text Preprocessing on text documents
2: Form the *Binary Document Term Matrix*
3: Apply PCA and select top 'i' PC's such that $\sum \lambda_i \geq 0.5$
4: Extract the negative and positive class data
5: Train the OCSVM with negative class data
6: Test the OCSVM with positive class data
7: Compute the Sensitivity value

Table 5.2: Datasets Description

Dataset	Samples	Features extracted
Malware [187]	388	313
Imperial Bank [186]	786	216
SW [210]	280	2038
20NG [210]	2000	16237

5.5.2 Tools and Techniques Employed

A PC with Intel i5 processor, 2.6GHZ, 8GB RAM, 500GB HDD and 64-bitWindows 8 OS was used for simulations. Tools employed for this work are mentioned in the Table 5.3.

Table 5.3: Tools and Techniques Used

Technique Used	Used for	Tools
Preprocessing	Document Term Matrix	*RapidMiner*®
PCA	Dimensionality Reduction	*MATLAB*®
OCSVM	Classification	*LIBSVM*®

5.5.3 Dimensionality Reduction

PCA (Anderson [211]; Jolliffe [199]; Ferre [200]; Burges [212]; and Lian [213];) is one of the useful statistical techniques, which is applied in various fields like image processing, signal processing, pattern recognition and many branches of engineering and sciences for reducing the feature space dimensionality of datasets and also removing the multicollinearity in datasets. It is carried out by a linear transformation and eigen analysis, resulting in as many principal components as the number of original variables. The important hallmark of PCA is that each principal component is a linear combination of the original features and that the first principal component (PC) consists of maximum variance (information), then the second principal component explains second highest variance (information) in the original data and so on. Consequently, if one selects first few PCs then, he/she is assured of maximum variance accounted for, in the data. By eliminating the remaining PCs, feature space dimensionality reduction is achieved.

5.5.4 Performance Measures

Sensitivity (True Positive rate) was used as the performance metric for the evaluation of the proposed model. The goal is to train the classifier with negative class samples and test it with positive class labels.

$$Sensitifity = \frac{TP}{TP + FN} \tag{5.1}$$

Where, TP = True Positive and FN = False Negative

5.6 Results and Discussion

After extraction of features [167], PCA was performed for dimensionality reduction. [188]. Out of the principal components returned, the principal components which contribute to more than 50% of the information from the original data, were chosen. The number of principal components $\sum \lambda_i \geq 0.5$ computed from various datasets is as follows: 33 from 20NG, 1 from Malware, 2 from SW, 10 from Imperial Bank. From the above result, it can be asserted that the first

principal component of Malware explains more variance than the other principal components. Similarly, for SW dataset, first two principal components have a higher contribution than the remaining, whereas 10 PCs were chosen for Imperial Bank dataset. For 20NG dataset, significant amount of information is explained by the top 33 principal components.

The distribution of information in principal components generated for each dataset is as follows: For Malware dataset the first principal component was found to have explained more than 70% variance and the second principal component had a 20% variance, together covering more than 91% of the information. In SW dataset, the first principal component accounts for 43%, second principal component accounts for 17% and 6% variance is explained by the third principal component. For 20NG dataset, the first principal component was found to have explained 16% variance, second principal component accounts for 8%, and 3% variance is explained by the third principal component. Similarly, first principal component of the Imperial Bank dataset accounts for 9%, the second principal component accounts for 7%, third and fourth principal components had explained 6% variance each. Several parameter settings are available for tuning in OCSVMs. The best parameter combination always gives the lowest error rate. Experiments were conducted with four kernels applicable to OCSVM on all datasets and their performance is compared. They are: Linear, Polynomial, RBF, and Sigmoid kernels. The performance of these kernels on each of the datasets is discussed as follows. However, best values of accuracy obtained for each dataset are reported, irrespective of the choice of kernels, in the Table 5.4.

We start our discussion with 20NG dataset results; sensitivity values are reported separately for each kernel for each dataset. The values obtained are as follows: 99.9% (linear), 99.9% (polynomial), 100% (RBF) and 99.9% (sigmoid). Similarly, for Malware dataset the sensitivity values obtained with respect to each kernel are 95.94% (linear), 98.12% (polynomial), 100% (RBF) and 100% (sigmoid). For Imperial Bank dataset the values are 95.27% (linear), 99.81% (polynomial), 94.86% (RBF) and 98.53% (sigmoid). For SW dataset, the sensitivity values are 57.36% (linear), 58.24% (polynomial), 99.62% (RBF), 98.53% (sigmoid). All kernels performed almost similarly on 20NG dataset, with very slight variation. The highest accuracy was obtained with RBF kernel. On the

Table 5.4: Classification Accuracy of the Proposed Model on Various Datasets

Dataset	Proposed model
20NG	100*
Malware	100*
Imperial Bank	99.81*
SW	99.62*

Table 5.5: Classification Sensitivity of the various Datasets with Regular/ Binary Classifiers

Dataset	PCA-DT	PCA-NB	PCA-KNN	PCA-SVM
20NG	99.95	98.25	99.95	99.85
Imperial Bank	84.46	59.73	80.85	60.51
Malware	92.25	86.85	93.29	83.54
SW	84.64	69.28	81.78	85.35

Imperial Bank data, the polynomial kernel performed the best among all other kernels, followed by sigmoid. Whereas on Malware dataset, RBF and Sigmoid kernels performed well and the values obtained from these kernels were identical. For SW dataset, RBF kernel performed well, followed by sigmoid.

The results of the proposed model are tabulated in 5.4. Apart from the proposed novel method, the results were computed with regular binary classification models like DT, k-NN, NB and SVM for the comparative study purpose. For this purpose, the open source tool KNIME [168] was used. These sensitivity values are listed out in the Table 5.5. Here 10FCV method was performed.

The classification accuracy of binary classifiers in combination with PCA for 20NG, Imperial Bank, Malware, and SW datasets are listed in the Table 5.5. It can be observed that, most of the classifiers, except NB, performed equally well by achieving an accuracy of at least 99%. For Imperial Bank dataset, the highest accuracy of 84.46% was obtained with DT. Performance of rest of the models was not as good. Similarly, on Malware dataset, k-NN yielded the best results. However, on the SW dataset SVM produced the best accuracy of 85.35%, as compared to other models. Similarly, experiments were also conducted without dimension-

ality reduction (i.e. with all features). The results are reported in this subsection in the same format as seen previously. The results obtained with respect to 20NG dataset are 99.9% (linear), 99.9% (polynomial), 99.9% (RBF) and 99.9% (sigmoid). For this dataset, all kernels performed equally well. With respect to the Malware dataset, the sensitivity values obtained are 96.87% (linear), 96.87% (polynomial), 100% (RBF) and 66.87% (sigmoid). For Imperial Bank dataset, the values obtained are 77.52% (linear), 98% (polynomial), 53.56% (RBF) and 77.52% (sigmoid). Here, polynomial kernel performed the best, followed by linear and sigmoid. For SW dataset, the values obtained are 16.91% (linear), 23.53% (poly), 89.7% (RBF) and 16.91% (sigmoid). For this dataset, except RBF none of them performed well.

5.7 Conclusions

In this chapter, a model for document classification, which performs dimensionality reduction using PCA and one-class classification with OCSVM, is proposed. The experimental results show that the proposed approach has classified text documents with higher sensitivity. To assess the fruitfulness of the dimensionality reduction technique, experiments were conducted on 20NG, Imperial Bank, SW, and Malware. For the purpose of analysis, after running PCA, the models were trained with binary classifiers. Comparative evaluation of the proposed model is performed with methods available in the literature to gauge the efficacy of the former for document classification. The following conclusions are obtained from the study: PCA performs well in removing the multicollinearity in the data. PCA and OCSVM in tandem performed one-class classification very well on all datasets in terms of high sensitivity values of documents.

Chapter 6

Clustering of Text Documents assisted by Topic Modeling

This chapter presents a novel hybrid model for document clustering involving topic modeling and clustering. In what follows, the motivation behind the work and contributions made towards it are stated. In the later sections, a comprehensive description of the model is presented and the results are analyzed.

6.1 Introduction

Clustering is one of the fundamental tasks of data mining. The main difference between supervised (classification) and unsupervised learning (clustering) is the presence of labeled patterns in the former and absence of the same in latter. Thus, clustering [214] groups unlabeled data into clusters based on the underlying features and similarities between patterns in the data. The following challenges exist in clustering documents: (i) Identifying an appropriate feature subset, (ii) Selection of clustering method, and (iii) Selection of an appropriate validity measure of clustering. The selection of an optimal feature subset can make clustering efficient by decreasing the intra-cluster distance to minimum. Therefore, before clustering the data, various feature subset selection methods including Term Variance (TV), Term Significance (TS), and Document Frequency (DF) were adopted. More powerful and more informative technique namely, Latent Dirichlet Allocation (LDA)

was also employed, which essentially performs topic modeling thereby enabling feature selection. Several clustering algorithms are then explored to build robust clusters. Clustering has various applications, which includes customer segmentation, visualization, and indexing.

6.2 Motivation and Contributions

Text clustering is used to group similar documents. As each document results in many features, clustering becomes complex and time consuming. So, there is a need to increase the efficiency of the clustering method. In this context, it is proposed to apply topic modeling to identify the discriminative features and thereby construct a feature subset. A clustering algorithm can then be invoked on this subset to obtain better performance. In this work, we primarily draw the attention to feature reduction through feature selection, and its influence on the clustering performance.

Contributions of this chapter are:

1. Effectiveness of feature selection through various unsupervised methods including Term Variance (TV), Term Significance (TS), Document Frequency (DF), and Latent Dirichlet Allocation (LDA) is analyzed.

2. The performance of various clustering algorithms namely, k-Means, k-Medoids, SOM, and Fuzzy C-means algorithms in combination with the feature selection methods, is studied.

6.3 Related Work

In this section, the significant contributions made towards text clustering are briefly reviewed. Initially, Jain et al. [214] tossed the word '*text clustering*' in 1999. Later, Steinbach et al. [215] presented their work on the comparison of document clustering techniques and analysis of k-Means, bisecting k-Means and hierarchical clustering algorithms. The experimental study was carried out on TREC and Reuters' datasets with F-Measure and entropy as the performance

metrics. They concluded that bisecting k-Means performed better than the other clustering algorithms.

Liu et al. [216] introduced a new feature selection method called Term Contribution (TC) for document clustering and studied different feature selection methods like document frequency, information gain, chi-square, term strength. They experimented with Reuters, Web and 20NG datasets with entropy and precision as performance metrics. Zhao and Karypis [217] evaluated the performance of partitional clustering algorithms on various functions, and 15 different datasets (available in `http:\cs.umn.edu`) have been used for experimental purpose to characterize these functions. Liu et al. [218] also proposed a feature selection method called Term Variance (TV) for document clustering. Document Frequency (DF), Term Variance Quality (TVQ), and Term Contribution (TC) were compared using TREC dataset. They concluded that feature subset selection methods could improve the clustering performance. Andrews and Fox [219] presented an overview of recent developments in document clustering. They provided a comparative analysis of the algorithms and discussed the future directions along with open problems. In the same year, Gelgi et al. [220] presented a model for clustering of web search results with TermRank as a feature selection method. The proposed approach was tested with two clustering algorithms on open directory project data and results were compared. They concluded that TermRank performed better than other (term frequency and tf-idf) feature ranking methods with respect to the F-Measure values.

Jadhao and Murthy [221] proposed a hybrid algorithm for document clustering which works by invoking online Non-negative Matrix Factorization (NMF) for dimensionality reduction followed by the K-means algorithm for clustering. They concluded that the proposed model yields better results as compared to the existing models. Aggarwal and Zhai [222] conducted a survey on text clustering algorithms and discussed various types of clustering algorithms (partitioning, agglomerative, etc.) and feature selection methods. In a text corpus, not all terms have discriminative ability. Few terms may have little impact individually, but when they are grouped their impact is more. Through this assumption, Shamsinejadbabki and Saraee [223] proposed a model based on Modified Term Variance (MTV) and Genetic Algorithm (GA) for finding discriminative features combined

with the k-Means algorithm. They experimented on the Reuters collection with accuracy and F-Measures as the performance metrics.

Bharti et al. [224] proposed a hybrid feature selection method for document clustering by employing the Document Frequency and Term Variance for selecting important features. Later, a feature set was generated by combining the features obtained from these two approaches (for generating combined feature subsets) and then principal component analysis was applied to it. Then k-Means clustering was employed for clustering the documents using the generated feature set. Experimental evaluation was made on the Reuters collection and Classic datasets with F-measure as a performance metric.

Basu and Murthy [225] proposed a model for document clustering using a hybrid method. A two-stage approach based on a new distance metric and clustering was introduced. The first stage performs hierarchical clustering using a new distance metric and second stage performs k-Means document clustering. Finally, the performance of the proposed model was assessed over TREC, Reuters, and Ohsumed datasets using F-Measure values. Bharti and Singh [226] proposed a model based on modified Artificial Bee Colony (ABC) algorithm for document clustering. Through this approach, they selected the appropriate cluster centers. They experimented on Reuters and Classic4 datasets and compared the performance of the proposed model with a traditional clustering algorithm (k-Means) as well as with Artificial Bee Colony algorithms. Recently, Abualigah et al. [227] proposed a model for document clustering. The suggested approach employs evolutionary algorithms for dimensionality reduction followed by K-Means algorithm for clustering. They introduced a new feature weighting scheme called Length Feature Weight (LFW) which assigns weights for each term. The optimal number of features is then selected through the Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Harmony Search algorithms. They concluded that through the proposed method they achieved the best F-Measure values.

The various works are summarized in below Table 6.1.

Table 6.1: Summarization of Related works on Document Clustering

Study	Dataset	Model	Performance
Jain et al. [214]	———	Review	———
Steinbach et al. [215]	TREC, Reuters	k-Means, Bisecting k-Means, Hierarchical clustering	F-Score
Liu et al. [216]	Reuters, 20NG	FS comparative study with k-Means	Precision, Entropy
Zhao and Karypis [217]	UMN datasets	Partitional clustering	Entropy
Liu et al. [218]	TREC, Reuters	DF/ TC/ TV with k-Means	Accuracy
Andrews and Fox [219]	———	Comparative analysis	———
Gelgi et al. [220]	ODP dataset	k-Mean	F-Score
Jadhao and Murthy [221]	Reuters, 20NG	NMF + k-Means	Mutual Information, Purity
Aggarwal and Zhai [222]	———	Survey article	———
Shamsinejadbabki and Saree [223]	Reuters	Modified TV/GA + k-Mean	F-Score
Bharti et al. [224]	Reuters, Classic	Hybrid Feature Section/ PCA + kMean	F-Score
Basu and Murthy [225]	Reuters, TREC, OHSUMED	Hybrid (Hirarchical + k-Mean)	F-Score
Bharti and Singh [226]	Reuters, Classic	Artificial Bee Colony, k-Mean	Precision, Recall, F-Score
Abualigah et al. [227]	LABIC dataset	PSO/ GA/ HS (Feature Selection) + k-Mean	F-Score

6.4 Proposed Methodology

The proposed methodology consists of three phases as depicted in Figure 6.1. The first phase consists of the preprocessing phase where the text is processed by following the steps of tokenization, removal of stop words and stemming. Finally, the document-term matrix is formed. In the second phase, feature subset selection is performed using four methods DF, TS, TV and LDA. In the third phase, the selected feature subsets are clustered separately using four clustering algorithms i.e. k-Means, k-Medoids, SOM, and Fuzzy C-means algorithms. Finally, the evaluation metrics namely, Precision, Recall, and F-Measure values are computed.

The proposed methodology employing Feature selection methods and clustering algorithms, is described in the following Algorithm 6.1.

Algorithm 6.1 FS_CLUST

Input: Text documents

Output: Clustering the documents

1: Do Text Preprocessing on text documents
2: Form the *Binary Document Term Matrix*
3: Apply DF/ LDA/ TS/ TV and select top 'n' features
4: Perform clustering using k-Means/ k-Medoids/ SOM/ Fuzzy C-Means algorithms on reduced feature space document term matrix
5: Repeat the step 4 for various combinations i.e. features selected in step 3
6: Compute the Precision, Recall and F-Measure values

6.5 Dataset and Techniques and Performance Measure

6.5.1 Datasets Description

To test the efficacy of the proposed model, we conducted experiments on the following datasets: (i) 20NG, which is a popular dataset consisting of 20 subgroups and each group comprises 1000 documents. We consider a subset (Graphics, Baseball, Electronics, and Politics) of 20NG which contains 4000 documents . (ii)

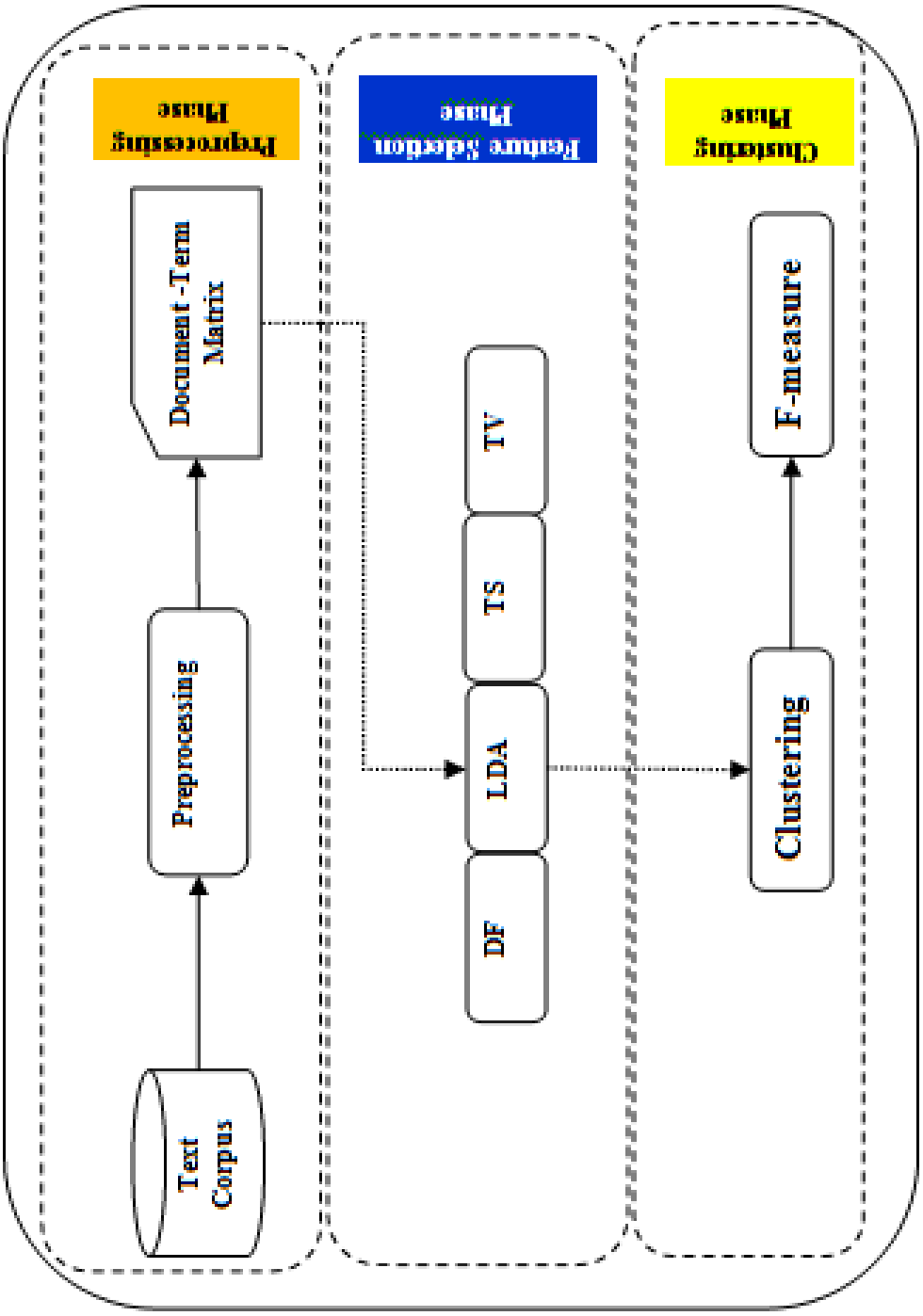

Figure 6.1: Schematic Diagram of Proposed Methodology

Table 6.2: Tools and Techniques Used

Technique Used	Used for	Tools
Preprocessing	Document Term Matrix	*RapidMiner*®
TV/ TS/ DF/ LDA	Feature Selection	*R*®
k-Means/ k-Medoids/ SOM/ Fuzzy C-means	Clustering	*R*®

WebKB, consisting of four groups namely course, faculty, project and student. It contains a total of 4159 documents. We collected the aforementioned datasets from the available online sources.

6.5.2 Tools and Techniques Used

A PC with Intel i5 processor, 2.6GHZ, 8GB RAM, 500GB HDD and 64-bit Windows 8 OS was used for modeling and experiments. Tools employed for this work are described in the Table 6.2.

6.5.3 Clustering Algorithms

6.5.3.1 k-Means

k-Means algorithm was proposed by Lloyd [228] in 1957 (published in 1982). Due to its simplicity and ease of implementation [228], it is widely using in clustering tasks even though it was proposed long ago. However, it suffers from the following disadvantages: the clustering solution produced by the algorithm is highly sensitive to the choice of initial cluster centers. Also, at its core it works as an optimization algorithm which tries to minimize the sum of squared differences between the data points and their corresponding cluster centers. As such, it is susceptible to local minima. For further details of the algorithm, readers are referred to the works of Jain and Dubels [229].

6.5.3.2 k-Medoids

It is also known as Partitioning Around Medoids (PAM), proposed by Kaufman and Rouseeuw [230]. It is similar to the k-Means algorithm except that, instead

of centroids it considers the data points/ objects as medoids which are centrally located objects in the cluster. Its working principle is the basis of minimization of the sum of dissimilarities in the data points. When the data set is large, it may not work in an efficient manner due to its time complexity it is the drawback of this algorithm.

6.5.3.3 Self Organizing Maps (SOM)

The Self-Organizing Map was introduced by Kohonen [231]. It is one of the traditional neural network models. It has the following advantages: (i) It identifies the features inherent to the data. So, we can also call it as a Self-Organizing Feature Map. (ii) It can recognize the patterns that have have not been presented to it before, i.e. generalization. For further details refer Kohonen [232], [231].

6.5.3.4 Fuzzy C-means

It is a clustering algorithm proposed by Dunn [233] which is commonly used for fuzzy clustering applications. In this approach, data points are bounded in the cluster by its membership function mean value, such that each data point can belong to more than one clusters, thereby allowing the formation of fuzzy clusters. For details, please refer to Bezdek [234] article.

6.5.4 Unsupervised Feature Selection Methods

To extract the discriminative features, we employed the following feature selection methods. These methods are described below.

6.5.4.1 Term Variance (TV)

Term Variance [218] assigns a value to each feature based on the deviation from the mean value. In this case, non-uniformly distributed features are more important as compared to uniformly distributed features. It is calculated as follows:

$$Term\_Variance = \frac{1}{n}\sum_{j=1}^{n}(x_{ij} - \bar{x_i})^2 \tag{6.1}$$

$where, n = total\_no.of\_documents, D_j = j^{th}document, x_{ij} = i^{th}term\_of\_document\_D_j, \bar{x_i} = mean\_value\_of\_i^{th}term$

6.5.4.2 Document Frequency (DF)

Document frequency was proposed by Yang and Pedersen [160]. It is another method for finding the important features based on the frequency distribution. It assigns a value based on the number of documents containing that feature. A feature is allotted a higher weight if it is frequently occurring and is otherwise allotted less weight (rare feature).

6.5.4.3 Term Significance

Rare words are not useful for the performance of the clustering task. Salton and McGill [235] introduced the Term Significance criterion for feature selection. It is calculated as follows:

$$Term\_Significance = \sum_{i=1}^{n} tf_i^2 - \frac{1}{n}\Big(\sum_{i=1}^{n} tf_i\Big)^2 \tag{6.2}$$

where, n=no.of_documents, tf=term_frequency

6.5.4.4 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA), proposed by Blei et al. [236] is also called as Topic modeling. It is a probabilistic Bayesian model. Each item is a collection of a mixture of topics, and these are sets of combinations of topic probabilities. Topic modeling provides an explicit representation of a document and is widely used in various applications to discover the topics present in the documents. It preserves the essential characteristics of text which are useful for regular data mining tasks. For further details on LDA, please refer to Blei [237].

6.5.5 Performance Measures Used

The proposed model was evaluated with the following metrics:

$$Precision = \frac{TP}{TP + FP} \tag{6.3}$$

$$Recall = \frac{TP}{TP + FN} \tag{6.4}$$

$$FMeasure = \frac{2 * (Precision * Recall)}{Precision + Recall} \tag{6.5}$$

Where, TP=True Positive, TN=True Negative, FP=False Positive, FN=False Negative.

6.6 Results and Discussion

After preprocessing the text, 7195 features were extracted from the 20NG dataset and 4159 features were extracted from WebKB dataset. Various feature subset selection methods (DF, TS, TV, and LDA) were applied to remove the redundant features. Through topic modeling, top 'k' topics were obtained for feature selection. Thus, for each topic, top-ranked features were selected based on probability. The list of selected combinations are given in the Table 6.3.

Table 6.3: Combinations of Features Selected

Feature selection method	Number of features selected
LDA_k_2	4, 6, 8, 10, 12, 14, 16, 18, 20, 24, 30, 40, 50
LDA_k_3	6, 9, 12, 15, 18, 21, 24, 27, 30, 39, 45, 51
LDA_k_4	4, 8, 12, 16, 20, 24, 32, 40, 52
LDA_k_5	5, 10, 15, 20, 25, 30, 35, 40, 50
LDA_k_6	6, 12, 18, 24, 30, 42, 48, 60
TS	4, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 100, 200, 300, 400, 500
TV	4, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 100, 200, 300, 400, 500
DF	4, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 100, 200, 300, 400, 500

Experiments were conducted with all combinations of the features seen in the above Table 6.3 and the best values obtained by these combinations are reported in Table 6.4 and Table 6.5. Table 6.4 presents the results of the 20NG dataset. Similarly, Table 6.5summarizes the results of the WebKB dataset.

In this work, the results of the proposed model were evaluated based on F-Measure value. First, the 20NG dataset is discussed. The values concerning the

Table 6.4: Clustering Results with 20NG Dataset

			K-Means			K-Medoids			SOM			Fuzzy C-means		
Method	**Features**		**Pr**	**Re**	**F**	**Pr**	**Re**	**F**	**Pr**	**Re**	**F**	**Pr**	**Re**	**F**
DF	5		0.43	0.25	0.32	0.18	0.23	0.20	0.18	0.24	0.21	0.32	0.26	0.29
	15		0.15	0.24	0.19	0.25	0.27	0.26	0.06	0.24	0.09	0.25	0.23	0.24
	300		0.31	0.24	0.27	0.19	0.25	0.21	0.06	0.24	0.09	0.32	0.6	**0.41***
LDA	**k=2**	4	0.55	0.38	0.45	0.88	0.86	0.87	0.06	0.24	0.09	0.35	0.44	0.39
		14	0.82	0.58	**0.68***	0.98	0.98	**0.98***	0.06	0.24	0.09	0.24	0.20	0.22
		40	0.57	0.46	0.51	0.98	0.98	**0.98***	0.56	0.25	**0.34***	0.01	0.02	0.02
	k=3	51	0.82	0.49	0.61	0.57	0.57	0.57	0.06	0.24	0.1	0.09	0.18	0.09
	k=4	32	0.58	0.58	0.55	0.98	0.98	**0.98***	0.56	0.25	0.34	0.5	0.29	0.36
	k=5	40	0.26	0.04	0.07	0.98	0.98	**0.98***	0.06	0.24	0.1	0.27	0.30	0.28
	k=6	60	0.56	0.37	0.44	0.98	0.98	**0.98***	0.06	0.24	0.1	0.28	0.03	0.06
TS		4	0.19	0.25	0.21	0.40	0.26	0.31	0.56	0.24	**0.34***	0.16	0.25	0.20
		200	0.53	0.26	0.34	0.22	0.24	0.23	0.06	0.24	0.09	0.28	0.34	0.31
TV		4	0.31	0.25	0.27	0.06	0.24	0.09	0.06	0.24	0.09	0.31	0.25	0.27
		10	0.31	0.25	0.27	0.06	0.24	0.09	0.06	0.24	0.09	0.41	0.25	0.31

k=No.of topics, *=Best Values, Pr=Precision, Re=Recall, F=F-Measure

Table 6.5: Clustering Results with WebKB Dataset

			K-Means			K-Medoids			SOM			Fuzzy C-means		
Method		**Features**	**Pr**	**Re**	**F**	**Pr**	**Re**	**F**	**Pr**	**Re**	**F**	**Pr**	**Re**	**F**
DF		15	0.47	0.28	0.35	0.45	0.30	0.36	0.59	0.25	0.35	0.24	0.23	0.24
		25	0.13	0.22	0.16	0.43	0.31	0.36	0.59	0.25	0.35	0.43	0.32	0.37
		100	0.13	0.22	0.16	0.41	0.27	0.33	0.84	0.25	**0.38***	0.25	0.24	0.25
LDA	**k=2**	14	0.32	0.22	0.26	0.54	0.34	0.42	0.70	0.25	0.37	0.40	0.29	0.35
		30	0.32	0.25	0.35	0.52	0.44	**0.47***	0.30	0.25	0.27	0.51	0.42	0.46
	k=3	27	0.49	0.28	0.35	0.54	0.35	0.42	0.54	0.25	0.34	0.13	0.16	0.14
	k=4	52	0.15	0.25	0.19	0.58	0.35	0.43	0.59	0.25	0.35	0.34	0.33	0.34
	k=5	50	0.58	0.29	**0.38***	0.48	0.32	0.38	0.34	0.25	0.29	0.13	0.22	0.16
	k=6	30	0.09	0.21	0.13	0.54	0.37	0.44	0.84	0.25	**0.38***	0.24	0.29	0.27
TS		10	0.50	0.25	0.34	0.42	0.25	0.31	0.59	0.25	0.35	0.12	0.21	0.15
		20	0.43	0.25	0.32	0.42	0.25	0.31	0.34	0.25	0.29	0.26	0.28	0.27
		100	0.36	0.25	0.29	0.53	0.30	0.38	0.34	0.25	0.29	0.35	0.25	0.29
TV		35	0.09	0.24	0.13	0.22	0.24	0.23	0.09	0.24	0.13	0.52	0.58	**0.55***
		40	0.40	0.25	0.31	0.22	0.24	0.23	0.09	0.24	0.13	0.32	0.26	0.29
		300	0.08	0.24	0.12	0.59	0.25	0.35	0.09	0.24	0.13	0.29	0.24	0.26

k=No.of topics, *=Best values, Pr=Precision, Re=Recall, F=F-Measure

feature selection methods, the number of features selected, and the results of clustering algorithms are reported in Table 6.4. Here 'k' is the number of topics chosen. Highest F-Measure of 0.98 was obtained with both, 14 and 40 feature (k=2) subsets, with LDA. We obtained an identical value of F-Measure also with 32 feature (k=4), 40 feature (k=5), and 60 feature (k=6) subsets. All these results were obtained through the k-Medoids algorithm. The next best value of 0.68 was obtained with 14 features (k=2) subset using the K-means algorithm, followed by SOM with 0.34 value for 40 features (k=2) subset. For DF feature selection method a value of 0.41 was obtained with 300 features subset clustered with the fuzzy C-means algorithm. Similarly, with TS, the best value obtained was 0.34 with SOM algorithm on a subset of four features. For TV with a feature subset of ten features clustered with the fuzzy C-means algorithm an F-Measure of 0.31 was observed. For this dataset, the performance was inferior with TV, TS and DF feature selection methods.

Similarly, for the WebKB dataset the following results are reported in Table 6.5. Highest value of 0.55 was obtained through TV feature selection method on a subset of 45 features with fuzzy C-means algorithm. The next best value of 0.47 was obtained with a combination of LDA, 30 features (k=2) subset and the K-medoids algorithm, followed by 0.38 with 50 features (k=5) with K-means algorithm and 30 features (k=6) with SOM algorithm. Similarly, identical values were obtained with DF, for a 100 features subset with SOM algorithm. A value of 0.35 was obtained for a combination of DF, K-means algorithm, and a feature subset of 15 features. For this dataset, TV performed the best with fuzzy C-means algorithm for only one case compared to other feature subset selection methods. However, LDA performed better than DF, TS, and TV in most of the cases. Through this analysis, it is concluded that LDA identifies the discriminative features which lead to the most precise grouping of the documents.

In k-Medoid clustering method set of points from the original data is considered as medoids. The key idea behind this method is to find the optimal set of points around the clusters which are formed from the original corpus. Its working principle is based on the minimization of sum of dissimilarities. It outperformed other clustering algorithms. LDA performs better on both datasets with all clustering algorithms; an exception is highlighted in the above paragraph. Hence,

it is suggested that LDA is preferable when compared to other feature subset selection methods.

6.7 Conclusions

In this chapter, a model for text document clustering in which, feature subset selection acts as a precursor to clustering, is proposed. Initially, the discriminative features were selected from the document term matrix using LDA, TV, TS, and DF methods. Subsets of various sizes were generated from the results of feature subset selection methods, and then various clustering algorithms namely, k-Means, k-Medoids, SOM, and Fuzzy C-means algorithms were run on each of the subsets. The performance of the clustering models was evaluated with three metrics (precision, recall, F-measure). Through the proposed model the best F-measure values were achieved on the 20NG dataset with a combination of LDA and the k-Medoids algorithm. It is concluded that LDA is performing better as compared to other feature selection methods.

Chapter 7

Class Association Rule Mining of Text Documents with Feature Selection by Topic Modeling

This chapter presents a hybrid model for document classification. It states the motivation behind the work and contributions made towards it. Later, the chapter describes the proposed model and then an analysis of the results is presented.

7.1 Introduction

Of late, data, both in structured and unstructured formats, exploded exponentially. In particular, huge amount of text data is generated from social media and online shopping websites. As these texts change dynamically, sophisticated techniques to analyze them are a need of the hour.

In this chapter, a novel hybrid model that performs feature selection through topic modeling, followed by classification through class association rule mining based on Apriori [238] and Frequent Pattern (FP) Growth algorithms, is proposed. The difference between the earlier works and the proposed method is that the former applied the regular feature selection methods followed by classification task with regular classifiers.

7.2 Motivation and Contributions

Text classification basically aims at predicting the class to which a document belongs to. Prediction of the accurate class is always a challenging problem. In Market Basket Analysis, association rule mining (ARM) can help in establishing the relationship among products very well. As text classification task also focuses on establishing the relationship of documents with classes, the adoption of ARM in text classification can be beneficial. So, we are motivated to adopt ARM in text classification. LDA is eminently suitable for feature subset selection for text corpora. LDA accomplishes it through topic modeling, whereas other feature selection methods are very generic in nature. Similarly, CARM combines strengths of classifiers as well as ARM (Association Rule Mining). Therefore, we combined them for developing a novel hybrid classifier.

Contributions of this chapter are:

1. A hybrid model for text classification is proposed.
2. The efficiency of the LDA algorithm in performing feature selection, is studied.
3. Classification is performed using the rules generated by the Apriori and FP Growth algorithms instead of the traditional classifiers.

7.3 Related Work

In this section, existing works in the fields of Association Rule Mining (ARM), topic modeling, Malware analysis and various other applications of ARM such as medical diseases prediction and phishing detection are reviewed. The related research conducted in the field of ARM is as follows: Associative Classification is a data mining technique which uses association rule mining for the classification task. To discover associations and co-occurrences among the words, FACT (Finding Associations in the Collections of Text) system was developed by Feldman and Hirsh [239]. A review of Associative Classification methods was presented

by Abdelhamid and Thabtah [97].Works related to Adaptive Associative Classification [240] and Multiclass Associative Classification (MAC) [241] are also found in literature. Lent et al. [242] proposed a method to detect trends in text databases. They used sequential pattern mining to identify phrases in text, and shape queries to identify trends. This approach was evaluated using the US patent systems dataset.

The primary Class Association Rules (CARs) proposed by Ma and Liu [243] uses a combination of classification and ARM. The proposed model was evaluated using 26 datasets available in the UCI repository. They concluded that this model performed better than the regular C4.5 classifier. Later, Yin and Han [244] proposed a model based on association rule mining called Classification based on Predictive Association Rules (CPAR). The Multiclass and Multilabel Classification model based on Association Rule Mining (MMAC) was proposed by Thabtah et al. [245]. This method produces the rules for multi-class classification problems. The runtime behaviour of the system is helpful to identify whether any malicious processes are being executed. The most familiar methods to find the intrusion are based on Windows API calls analyzed dynamically. In the research work of Ahmed et al. [117], Malware detection was based on spatial (arguments) and temporal (sequences) features using Windows API calls. They extracted the features and applied Machine learning techniques for detection of malicious activity. They concluded that the spatio-temporal feature subset helps in improving the performance. Similarly, Abdelhamid et al. [241] proposed a new model called Multiclass Associative Classification (MAC). They evaluated it on 19 datasets taken from UCI repository. They compared its efficacy with that of classifiers like RIPPER and C4.5. Another work related to classification based on Multiple Association Rules (CMAR) was proposed by Li et al. [246].

Various works relevant to topic modeling are as follows: Wang and Blei [247] developed a model for finding relevant and appropriate articles in online search. They combined the advantages of collaborative filtering with topic modeling to identify the latent structure of users and items. They demonstrated the effectiveness of this approach on CiteULike dataset. They concluded that this method was performing better than the traditional methods. Gujraniya and Murthy [248] proposed a bag of phrases model to identify the discriminative power of phrases.

They extracted phrases from the text using the topic model and represented them using the vector space model. They reported that it performed better than the previous classifiers.

The works relevant to Malware analysis found in the literature are as follows: Anomaly detection through API calls was pursued primarily in Forrest et al. [249]. They used the n-gram approach for detection of anomalies. Reddy and Pujari [250] proposed a model based on n-gram and another based on byte sequence model for malware detection. They used these methods for feature extraction and feature selection by information gain. Classification was then performed by combining SVM, DT, and Instance-based Learner classifiers. They reported that this approach yielded the best results. Shanakarapani et al. [251] presented a research work based on kernel methods to identify the malwares. In this work, API calls were extracted from PE files, weighted with term frequency approach and then a classifier was trained with SVM. Sami et al. [252] proposed a model based on portable executables for malware detection. Using executable files, they extracted API calls. Later, they invoked various classifiers like Naive Bayes, Random Forest, J48 (DT). They reported sensitivity, accuracy values of 98.31%, and 99.7% respectively.

N-gram analysis for malware detection has the following disadvantage: features are directly proportional to the value of n i.e., when n increases, the number of features also increase. To overcome this O'Kane et al. [253] presented a framework using eigenvector analysis for malware detection. Sundarkumar and Ravi [121] proposed a Malware detection model using Windows API calls through text mining with the informative features selected by Mutual Information criterion. After over-sampling the positive class samples to balance the data, classification task was performed using DT/ Support Vector Machines (SVM)/ Multilayer Percepton (MLP)/ Probabilistic Neural Network (PNN)/ Group Method Data Handling (GMDH) validated by 10-Fold Cross Validation. They reported a sensitivity value of 100% with SVM and OCSVM. Malware analysis using text mining approach with topic modeling was carried out by Sundarkumar et al. [123]. They combined topic modeling with data mining techniques. They employed various classifiers to detect Malware. They tested their method on two datasets which are publicly available. They concluded that DT and SVM out-performed other

models and preferred DT to SVM since, DT yields easily interpretable if-then rules.

Fan et al. [254] proposed a model using sequential pattern mining and All Nearest-Neighbor for malware detection. They experimented with 10,307 windows PE samples; distribution of these samples is as follows: 8847 of malicious and 1460 of benign instances. They compared the corresponding results with other classifiers such as SVM, k-NN, J4.8 and NB. They concluded that the proposed model performed better than other models with respect to accuracy (95.25%) as well as detection rate (96.17%). Hashemi et al. [255] presented a framework for malware detection using executable files operational codes (OpCode). They generated a graph using this OpCode and later they converted it into eigenvectors which are the linear combination of the executable files. Then, they employed Adaboost, DT, SVM and k-NN classifiers. They evaluated the performance of the model with F-Measure and False Positive Rate (FPR) on the vx_heaven dataset. They concluded that Adaboost and SVM performed better as compared to other models.

Huda et al. [256] proposed a semi-supervised approach for malware detection. Initially, they extracted the malware patterns using k-Means clustering. Then geometric information of the patterns was obtained. They employed the SVM, J48, NB, RF and Instance-Based (IB) classifiers for classification purpose. They experimented with two datasets, namely, CA Technologies dataset (485 samples) and Win32-based executable calls (967 samples). The results were also compared with other supervised models. They concluded that the proposed semi-supervised approach yielded a detection rate as well as AUC of 100% with a combination of SVM and RF. Recently, Salehi et al. [257] proposed a model for malware detection using arguments of API calls and its return values. They selected important features through Fisher score method and then trained a classifier with Support Vector Machine based on Recursive Feature Elimination (SVM-RFE). They reported the 10FCV accuracy of 99.4%. The work that finds the association between structured records and text documents was proposed by Agrawal et al. [258]. Phishing detection based on Association Rule Mining was carried out by Abdelhamid et al. [177]. They proposed the Multi-label Classifier based Associative Classification (MCAC) method to identify the prominent features

of legitimate, and phishing websites.. They deployed the proposed model on adataset consisting of 1350 websites.

In literature most of the works exist on regular feature selection methods followed by any binary classifiers. Few works exist in Class Association Rule Mining, where, they addressed the classification task as binary as well as multi-class problem. Very few works exist, which augment the classification task with topic modeling. This fact motivated us to propose topic modeling as a feature selector followed by classification with CARM. This will enable us to determine not only the class rules but also the rule strength. The various works are summarized in below Table 7.1.

7.4 Proposed Method

In this chapter, an integrated framework using the two methods mentioned above in tandem i.e., topic modeling followed by ARM is proposed. Through the experiments conducted, superior performance of the proposed approach for binary classification problems is demonstrated. The proposed methodology consists of three phases: preprocessing phase, training phase and test phase as depicted in Fig. 7.1. In the first phase, all samples of the two datasets viz., Malware dataset$_1$ and dataset$_2$ were collected, and the API calls were extracted from the datasets. Then, tokenization was performed on the datasets and a *Document Term Matrix (DTM)* was constructed, which is a structured data format. The terms were weighted using the "*Binary*" approach. The DTM was fed as an input to LDA to perform dimensionality reduction. LDA yielded a list of topics from which the top features having higher probabilities from each topic were extracted. This yielded us a dataset with reduced dimensionality, which was divided into 10 equal parts to perform 10-fold cross validation.

In the second phase, ARM techniques were employed and the association rules were generated from these features (API calls) with pre-specified minimum support and confidence to filter out the uninteresting rules. Next, the rules which do not contain the class label in the consequent were removed. Then, we were left with the Class Association rules. The strength of the rules was then computed by multiplying the obtained support and confidence for each rule. Later, we

Table 7.1: Summarization of Related works on Association Rule Mining

Study	Dataset	Model	Performance Measure
Feldman and Hirsh [239]	Reuters	FACT	CPU Time
Abdelhamid and Thabtah [97]	———	Review of associative classification	———
Abdelhamid [241]	UCI dataset	Multiclass Aussociative Classification (MAC)/ DT/ RIPPER	Lift
Lent et al. [242]	IBM Patents	Patent Miner System	Support
Ma and Liu [243]	UCI datasets	Class Association Rule/ C4.5	Accuracy
Yin and Han [244]	UCI datasets	Classification based on Predictive Association Rules (CPAR)	Accuracy
Thabtah et al. [245]	WEKA data collection	Multiclass Multilabe Association Rule Mining (MMAC)	Accuracy
Ahmed et al. [117]	API Calls	DT	Accuracy
Li et al. [246]	UCI datasets	Classification based on Multiple-Class Association Rules (CMAR)	Accuracy
Gujraniya and Murthy [248]	20NG, LingSpam	LDA +NB, LDA + SVM	Accuracy
Reddy and Pujari [250]	Vx Heavens	n-gram with SVM, DT	True Positive, False Positive
Shankarapani et al. [251]	Windows Portable Executable (PE) files	SVM	Sensitivity, Specificity
Sami et al. [252]	Windows PE files	NB/ RF/ DT	Sensitivity, Accuracy
O'Kane [253]	OpCodes	Eigen Analyis followed by SVM	Accuracy
Sundarkumar and Ravi [121]	Windows API Calls	DT/ SVM/ MLP/ PNN/ GMDH/ OCSVM	Sensitivity
Sundarkumar et al. [123]	Windows API Calls	SVM/ DT/ GMDH/ MLP/ RF	AUC
Fan et al. [254]	Windows PE files	Sequential Pattern Mining with k-NN/ SVM/ J4.8/ NB	Accuracy
Hashemi et al. [255]	OpCodes	Eigen Analysis with DT/ Adaboost/ SVM/ k-NN	F-Score, FP Rate
Huda et al. [256]	Win32 executable files, CA Technologies dataset	k-Mean + SVM/ J48/ NB/ RF/ Instance-Based (IB) Classifier	AUC
Salehhi et al. [257]	Windows API Calls	SVM-RFE	Accuracy
Abdelhamid et al. [177]	Millersmiles/ PhishTank	Associative Rule	Accuracy

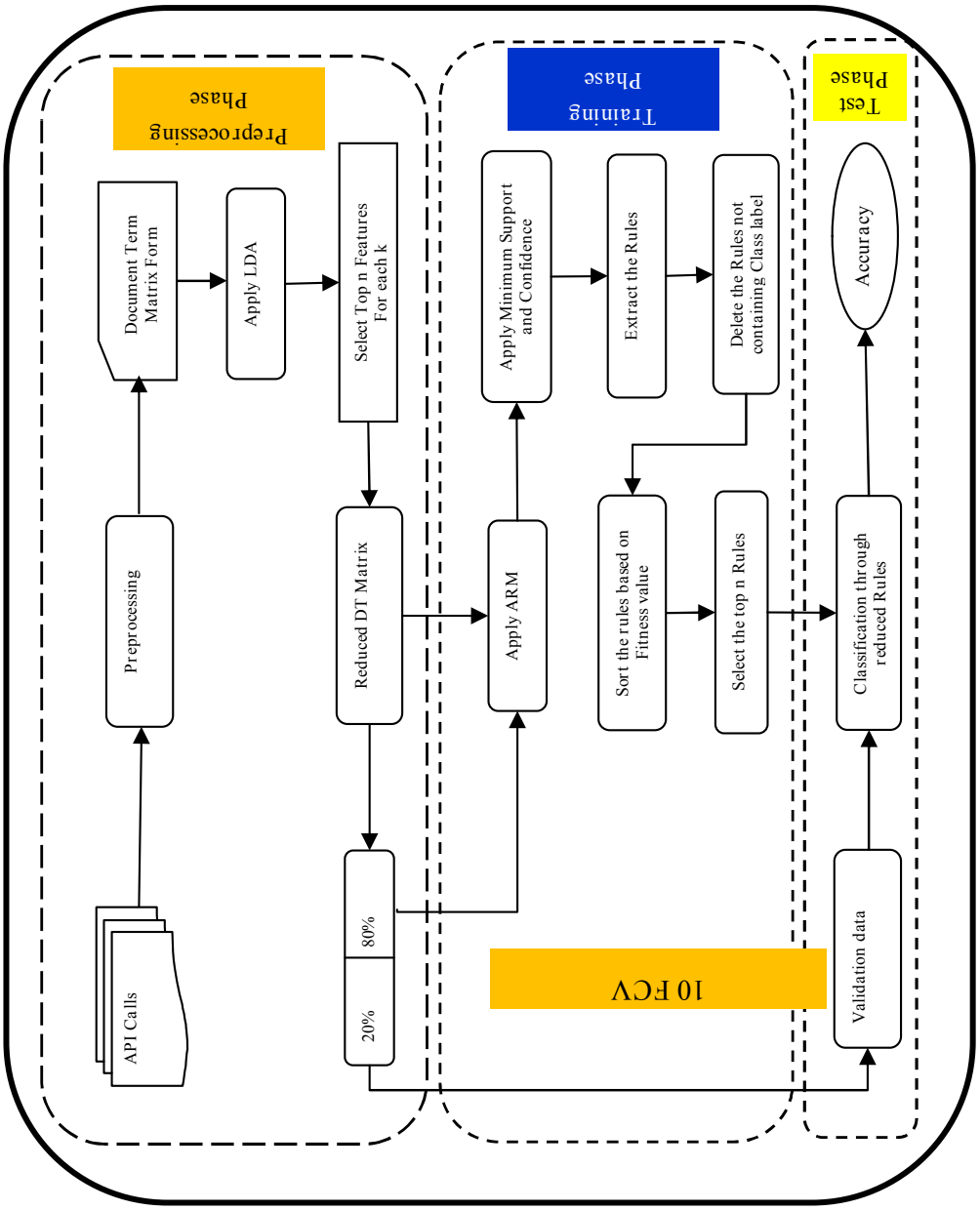

Figure 7.1: Schematic Diagram of Proposed Methodology

sorted all rules on the basis of their strength and selected a few top rules for classification purpose. Finally, in the test phase, validation was performed with the rules generated in the second phase. For each fold, a different set of rules was obtained; these were aggregated by removing the repetitive rules. The final set of rules obtained can perform associative classification in the domain under consideration. The proposed methodology employing topic modeling and class association rule mining, is described in the following Algorithm 7.1.

Algorithm 7.1 LDA_CARM

Input: Text documents consisting API call sequences

Output: Malicious software classes

1: Do Text Preprocessing on text documents
2: Form the *Binary Document Term Matrix*
3: Apply LDA and select top 'n' features for each 'k', this will decrease dimensions in DTM
4: Divide the data into 10 folds for 10-FCV
5: Repeat the steps 6 through 10 for all folds
6: Apply ARM Algorithms [238](Apriori, FP-Growth) with pre-specified minimum *support* and $confidence$
7: Extract association rules
8: Delete those rules not containing the Class label in the consequent.
9: Sort the rules based on fitness value, which is defined as *support * confidence*
10: Select the top n rules and build the classifier, where n is a pre-specified number.
11: Apply reduced rule set on validation set for Classification and compute the average values of sensitivity, specificity and accuracy over 10 folds.

7.5 Data, Techniques and Performance Measures Used

7.5.1 Datasets Used

To validate the proposed method, experiments were conducted on the datasets which consist of Windows API calls. It should be noted that program execution is a flow of API calls [259]. Windows operating system provides various types of API calls and they are categorized based on their functionalities. Each API call is identified by its unique name and its return value. Two datasets were analyzed. Malware dataset$_1$ is one of the datasets for the data mining contest in ICONIP 2010 and Csmining group [187]. It consists of a selection of Windows API/ System Call trace files in two parts: 388 logs, out of which there are 68 Benign software traces labeled as $'0'$ and 320 Malware traces labeled as $'1'$. Similarly, Malware dataset$_2$ consists of Windows API calls of 416 malware samples with a breakup of 117 Trojan, 165 Virus, 134 Worm and 98 Benign Samples [260]. API calls description was given by Alazab et al. [3] and these API calls are listed in Table 7.2.

7.5.2 Performance Measures Used

The following metrics were used to evaluate the performance of the proposed model.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{7.1}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{7.2}$$

$$Specificity = \frac{TN}{TN + FP} \tag{7.3}$$

$$AUC = \frac{(Sensitivity + Specificity)}{2} \tag{7.4}$$

Where, TP=True Positive, TN=True Negative, FP=False Positive, FN=False Negative, AUC=Area Under the ROC Curve.

Table 7.2: Description of the API Calls by Alazab et al. [3]

API calls	Description of the calls
FindFirstFile, FindFirstFileEx, FindFirstFileName,TransactedW, FindFirstFileNameW, FindClose, FindFirstStream, TransactedW, FindFirstStreamW,FindNextStreamW, SearchPath, FindNextFile, FindFirstFileTransacted, FindNextFileNameW.	Search files to infect
CloseHandle, CopyFile, CopyFileEx, CopyFileTransacted, CreateFileTransacted, CreateHardLink, CreateHardLink, Transacted, CreateSymbolicLink, CreateSymbolic, LinkTransacted, DeleteFile, DeleteFileTransacted, CreateFile.	Copy or delete files
GetBinaryType, GetCompressed, FileSize, GetCompressedFile, SizeTransacted, GetFileAttributes, GetFileAttributesEx, GetFileAttributes, Transacted, GetFileBandwidth, Reservation, GetFileInformation, ByHandle, GetFileInformation, ByHandleEx, GetFileSize, GetFileSizeEx, GetFileType, GetFinalPathName, ByHandle, GetFullPathName, GetFullPathName, Transacted, GetLongPathName, GetLongPathName, Transacted, GetShortPathName,GetTempFileName,GetTempPath.	Get file information
MoveFile, MoveFileEx, MoveFileTransacted, MoveFileWithProgress	Move files
OpenFile„OpenFileById, ReOpenFile, ReplaceFile, WriteFile, CreateFile, loseHandle.	Read or write files
SetFileApisToANSI, SetFileApisToOEM, SetFileAttributes„SetFileAttributesTransacted, SetFileBandwidthReservation„SetFileInformationByHandle, SetFileShortName, SetFileValidData	Change file attributes

7.5.3 Tools and Techniques

A machine with Intel i5 processor, 2.6GHZ, 8GB RAM, 500GB HDD and 64-bit Windows 8 OS was used for modeling and experiments. *RapidMiner*® [167] was used for text preprocessing and *R*® [261] was used for executing the Apriori algorithm. Tools employed for this work are described in the Table 7.3.

7.5.4 Feature Selection through Topic Modeling

Topic modeling, proposed by Blei et al. [236], is also called Latent Dirichlet Allocation (LDA). It is a probabilistic Bayesian model. In this technique, each item is considered as a collection of mixture of topics, and these are sets of combinations of topic probabilities. Topic modeling provides an explicit representation of a document and is widely used in various applications to discover the topics in the documents. It preserves the essential characteristics of text that are useful for regular data mining tasks. For further details on LDA, the reader is referred to

Table 7.3: Tools and Techniques Used

Technique Used	Used for	Tools
Preprocessing	Document Term Matrix	*RapidMiner*®
LDA	Fetaure Selection	*R*®
Apriori	Association Rules	*R*®
FP Growth	Association Rules	*RapidMiner*®
CARM	Classification	*R*®

Blei et al. [237]. Numerous applications developed using topic modeling include, recommender systems, software traceability, document summarization, etc.

7.5.5 Association Rule Mining

Association rule mining (Agrawal et al. [238]; Agrawal and Srikant [262]; Agrawal and Srikant [263]) detects a set of association rules that exist in the database, with a pre-specified minimum support and confidence. The main difference between association rule and classification rule mining (Class Association Rule Mining) is that a target variable is conspicuously absent in an association rule, whereas it is predefined (a class label) in the Class Association Rule Mining. Both techniques have diverse applications. Association rule mining works as follows: first, it finds antecedent and consequent associations; second, the support and the confidence of an association rule are estimated in order to determine interesting rules. Association rules are in the form of "*if X then Y*", where X and Y are two set of items. Here, X and Y are also known as antecedent or premise and consequent or conclusion respectively. Association rule mining found numerous applications in fields like health, marketing, financial (credit card transactions) etc. Let 'M' be a binary document-term matrix of size n by m where 'n' is the number of documents and 'm' is the number of features. For every document, there exists some features, which are a subset of 'm'. If $f_1, f_2, f_3, \ldots, f_m$ are features (API Calls) and C_1, C_2 are class labels of Malware and Benign respectively, the rule will be in the form of $(f_1, f_2, f_3, ..., f_m)$ implies $(C_1 \text{or } C_2)$. Here, $f_1, f_2, f_3, ..., f_m$ is called as an item set (association rule point of view). For a given item set

$f_1, f_2, f_3, ..., f_m$ the *Support* of a rule is calculated as in (1) and the $Confidence$ of a rule is calculated as in (2).

$$Support((f_1, f_2, f_3, \ldots) \Rightarrow (C_1 \, or \, C_2)) = \frac{No.of\_documents\_containing\_f_1, f_2, f_3, \ldots}{n} \quad (7.5)$$

$$Confidence((f_1, f_2, f_3, \ldots) \Rightarrow (C_1 \, or \, C_2)) = \frac{Support((f_1, f_2, f_3, \ldots) \Rightarrow (C_1 \, or \, C_2}{Support((f_1, f_2, f_3, \ldots)} \quad (7.6)$$

7.6 Results and Discussion

The number of features extracted from each dataset is as follows: 313 from Malware dataset$_1$ and 216 from Trojan, 216 from Virus and finally, 214 features from Worm of the dataset$_2$. To gauge the efficacy of the proposed model, experiments were first conducted with the complete set of features. The rules obtained are as follows: 353 million rules for Trojan, 355 million rules for Worm, 341 million rules for Virus with dataset$_2$ and 551 million rules for Malware with dataset$_1$. Since classification is not feasible with such a huge set of rules, we applied feature selection with an intention to obtain fewer rules. Later, top k topics were extracted for feature selection, where *k=2, 4, 8*. For each topic, $k * l$ number of top ranked features were selected based on the probability, where *l=1, 2 and 4*. In other words, the top two ranked features were selected based on probability thereby ending up with $4, 8$ and 16 features respectively, when *l=2*. Thus, sensitivity analysis was performed by considering 3 different values of l. Results for the top features selected through this process for *k=2* and *l=4* are reported . After extracting the rules, a threshold was imposed on the strength value (fitness value) to condense the number of rules. The minimum support was fixed as 0.3 and confidence as 0.9. Then the top most rules were chosen based on the fitness value. The selected features are listed in Table 7.4 and Table 7.5.

Ahmed et al. [117] employed Naive Bayes (NB), RIPPER, J4.8, SMO, and IBk algorithms for classification. The reported values are as follows: Trojan (96.6%), Virus (99%), and Worm (98.4%); all are produced by NB classifier. Recent works of Sundarkumar et al. [123] also experimented with this dataset. They performed the experiments with DT, SVM, MLP, GMDH, PNN, and Random Forest (RF)

Table 7.4: Features Selected from Malware dataset$_2$

Trojan Features	**Value**
GlobalMemoryStatus	0.79
CreateThread	0.76
VirtualAlloc	0.75
RegOpenKeyW	0.73
GetFullPathNameW	0.837
GetFileAttributesW	0.832
GetLongPathNameW	0.832
CreateFileW	0.797

Virus Features	**Value**
VirtualAlloc	0.785
RegNotifyChangeKeyValue	0.780
CreateThread	0.774
GlobalMemoryStatus	0.766
GetFileAttributesW	0.832
GetLongPathNameW	0.832
GetFullPathNameW	0.817
SearchPathW	0.789

Worm Features	**Value**
GetFullPathNameW	0.857
GetLongPathNameW	0.853
GetFileAttributesW	0.847
SearchPathW	0.768
CreateThread	0.807
VirtualAlloc	0.790
RegOpenKeyW	0.787
GlobalMemoryStatus	0.773

Table 7.5: Features Selected from Malware dataset$_1$

Malware Features	**Value**
WSAGetLastError	0.33
FreeLibrary	0.70
RegQueryValueA	0.45
GlobalMemoryStatus	0.48
GetLongPathNameW	0.37
GetShortPathNameW	0.28
NtOpenKey	0.61
SearchPathW	0.61

classifiers. It was reported that DT and SVM performed equally with respect to accuracy values, which were 98.46% and 98.63% respectively.

It is very interesting to note that with the proposed approach, an AUC value

Table 7.6: Comparison of Results of Proposed Model with Existing Models

Dataset		Approach	Accuracy	Sensitivity	Specificity	AUC
Dataset$_1$	**Malware**	Sundarkumar and Ravi [121]	73.43	80.54	59.09	0.698
		Sundarkumar et al. [123]	88.67	92.95	54.99	0.739
		Proposed approach (Apriori)	94.74*	100*	66.67*	0.833*
		Proposed approach (FP-Growth)	86.44	75.00	100*	0.875*
Dataset$_2$	**Trojan**	Ahmed et al. [117]	97	NA	NA	NA
		Sundarkumar et al. [123]	98.46	NA	NA	NA
		Proposed approach (Apriori)	100*	100*	100*	1*
		Proposed approach (FP-Growth)	100*	100*	100*	1*
	Virus	Ahmed et al. [117]	99	NA	NA	NA
		Sundarkumar et al. [123]	98.75	NA	NA	NA
		Proposed approach (Apriori)	100*	100*	100*	1*
		Proposed approach (FP-Growth)	100*	100*	100*	1*
	Worm	Ahmed et al. [117]	98.4	NA	NA	NA
		Sundarkumar et al. [123]	98.7	NA	NA	NA
		Proposed approach (Apriori)	100*	100*	100*	1*
		Proposed approach (FP-Growth)	100*	100*	100*	1*

NA=Not Available, *=Best Values

of 1 was obtained in all the cases of dataset$_2$, while the highest AUC of 0.875 was yielded by FP growth in the case of dataset$_1$. It is also remarkable that across all the 10 folds, identical top 20 rules were obtained in the case of both datasets. It shows that this model is not only robust and stable but also outperforms other approaches proposed for this problem.

The top-20 rules with respect to Malware, Trojan, Virus and Worm are presented in Table 7.7, Table 7.8, Table 7.9 and Table 7.10 respectively for the corresponding results of the Table 7.6.

Experiments were conducted separately with top-10, top-15, top-20, top-25 and top-30 rules. The accuracy values are reported in Table 7.6 for the experiment with top-20 rules. For Malware dataset$_1$, since it is unbalanced, we obtained a specificity of zero. To circumvent the situation, top rules from benign class also were included. It was found that addition of benign class rules resulted in high specificity. In dataset$_1$, when data was not balanced, the total number of rules generated by Apriori for Malware *vs*. Benign was 1540 with a breakup of 188 for Malware and 44 for Benign class. FP-Growth produced a total of 116 rules in which fifteen pertain to Malware. These were the observations for the unbalanced dataset. For Malware dataset$_2$, around 3261 rules were obtained for Benign *vs*. Trojan for very fold, with Apriori algorithm. 227 rules were obtained for Trojan

Table 7.7: Classification Rules for Malware

Rules	Antecedent	Consequent
1	if(GetLongPathNameW=1)	Malware
2	if(FreeLibrary=0 and GetLongPathNameW=1)	Malware
3	if(GlobalMemoryStatus=0 and GetLongPathNameW=1)	Malware
4	if(RegQueryValueA=0 and GetLongPathNameW=1)	Malware
5	if(FreeLibrary=0 and GlobalMemoryStatus=0 and GetLongPathNameW=1)	Malware
6	if(FreeLibrary=0 and RegQueryValueA=0 and GetLongPathNameW=1)	Malware
7	if(RegQueryValueA=0 and GlobalMemoryStatus=0 and GetLongPathNameW=1)	Malware
8	if(FreeLibrary=0 and RegQueryValueA=0 and GlobalMemoryStatus=0 and GetLongPathNameW=1)	Malware
9	if(WSAGetLastError=0 and GetLongPathNameW=1)	Malware
10	if(WSAGetLastError=0 and FreeLibrary=0 and GetLongPathNameW=1)	Malware
11	if(WSAGetLastError=0 and GlobalMemoryStatus=0 and GetLongPathNameW=1)	Malware
12	if(WSAGetLastError=0 and RegQueryValueA=0 and GetLongPathNameW=1)	Malware
13	if(WSAGetLastError=0 and FreeLibrary=0 and GlobalMemoryStatus=0 and GetLongPathNameW=1)	Malware
14	if(WSAGetLastError=0 and FreeLibrary=0 and RegQueryValueA=0 and GetLongPathNameW=1)	Malware
15	if(WSAGetLastError=0 and RegQueryValueA=0 and GlobalMemoryStatus=0 and GetLongPathNameW=1)	Malware
16	if(WSAGetLastError=0 and FreeLibrary=0 and RegQueryValueA=0 and GlobalMemoryStatus=0 and GetLongPathNameW=1)	Malware
17	if(GetLongPathNameW=1 and NtOpenKey=0)	Malware
18	if(GetLongPathNameW=1 and GetShortPathNameW=0)	Malware
19	if(FreeLibrary=0 and GetLongPathNameW=1 and NtOpenKey=0)	Malware
20	if(FreeLibrary=0 and GetLongPathNameW=1 and GetShortPathNameW=0)	Malware

Rules and 235 rules for Benign. Similarly, FP-Growth algorithm generated 202 rules. 15 rules were obtained for Trojan and eleven for Benign. For Benign *vs.* Virus classification, a total of 1540 rules (Apriori) were generated with a break up of 188 rules for Malware, and 44 rules for Benign class. Similarly, with FP-Growth, the number of rules generated was 115, including 15 rules for Virus. Finally, for Benign *vs.* Worm identification, a total of 2691 rules (Apriori) were produced. Distribution of these rules is as follows: 228 for Worm and 133 for Benign. FP-Growth generated 72 rules, where seven rules describe Malware and eight describe Benign. For the sake of comparison, the results of earlier works are included here. First, the recent work of Sundarkumar et al. [123] on Malware dataset$_1$ is discussed. They achieved the highest accuracy with GMDH (89.46%) followed by DT (88.67%), SVM (88.15%) and MLP (87.64%), Sensitivity value was high for MLP (98.61%) followed by SVM (96.87%), GMDH (94.12%) and DT (92.95%). They reported that with respect to AUC, DT and GMDH outper-

Table 7.8: Classification Rules for Trojan

Rules	Antecedent	Consequent
1	if(GetLongPathNameW=1)	Trojan
2	if(GetFileAttributesW=1)	Trojan
3	if(GetFileAttributesW=1 and GetLongPathNameW=1)	Trojan
4	if(GetFullPathNameW=1)	Trojan
5	if(GetFullPathNameW=1 and GetLongPathNameW=1)	Trojan
6	if(GetFullPathNameW=1 and GetFileAttributesW=1)	Trojan
7	if(GlobalMemoryStatus=0 and GetLongPathNameW=1)	Trojan
8	if(GlobalMemoryStatus=0 and GetFileAttributesW=1)	Trojan
9	if(GetFullPathNameW=1 andGetFileAttributesW=1 and GetLongPathNameW=1)	Trojan
10	if(GlobalMemoryStatus=0 and GetFileAttributesW=1 and GetLongPathNameW=1)	Trojan
11	if(GlobalMemoryStatus=0 and GetFullPathNameW=1)	Trojan
12	if(GlobalMemoryStatus=0 and GetFullPathNameW=1 and GetLongPathNameW=1)	Trojan
13	if(GlobalMemoryStatus=0 and GetFullPathNameW=1 and GetFileAttributesW=1)	Trojan
14	if(GlobalMemoryStatus=0 and GetFullPathNameW=1 and GetFileAttributesW=1 and GetLongPathNameW=1)	Trojan
15	if(VirtualAlloc=0 and GetLongPathNameW=1)	Trojan
16	if(VirtualAlloc=0 and GetFileAttributesW=1)	Trojan
17	if(VirtualAlloc=0 and GetFileAttributesW=1 and GetLongPathNameW=1)	Trojan
18	if(GlobalMemoryStatus=0 and VirtualAlloc=0 and GetLongPathNameW=1)	Trojan
19	if(GlobalMemoryStatus=0 and VirtualAlloc=0 and GetFileAttributesW=1)	Trojan
20	if(GetLongPathNameW=1)	Trojan

Table 7.9: Classification Rules for Virus

Rules	Antecedent	Consequent
1	if(GetFileAttributesW=1)	Virus
2	if(GetLongPathNameW=1)	Virus
3	if(GetFullPathNameW=1)	Virus
4	if(GetFileAttributesW=1 and GetLongPathNameW=1)	Virus
5	if(GetFileAttributesW=1 and GetFullPathNameW=1)	Virus
6	if(GetLongPathNameW=1 and GetFullPathNameW=1)	Virus
7	if(GetFileAttributesW=1 and GetLongPathNameW=1 and GetFullPathNameW=1)	Virus
8	if(GetFileAttributesW=1 and GetLongPathNameW=1 and GetFullPathNameW=1)	Virus
9	if(GlobalMemoryStatus=0 and GetFileAttributesW=1)	Virus
10	if(GlobalMemoryStatus=0 and GetLongPathNameW=1)	Virus
11	if(GlobalMemoryStatus=0 and GetFullPathNameW=1)	Virus
12	if(GlobalMemoryStatus=0 and GetFileAttributesW=1 and GetLongPathNameW=1)	Virus
13	if(GlobalMemoryStatus=0 and GetFileAttributesW=1 and GetFullPathNameW=1)	Virus
14	if(GlobalMemoryStatus=0 and GetLongPathNameW=1 and GetFullPathNameW=1)	Virus
15	if(GlobalMemoryStatus=0 and GetFileAttributesW=1 and GetLongPathNameW=1 and GetFullPathNameW=1)	Virus
16	if(VirtualAlloc=0 and GetFileAttributesW=1)	Virus
17	if(VirtualAlloc=0 and GetLongPathNameW=1)	Virus
18	if(VirtualAlloc=0 and GetFullPathNameW=1)	Virus
19	if(VirtualAlloc=0 and GetFileAttributesW=1 and GetLongPathNameW=1)	Virus
20	if(VirtualAlloc=0 and GetFileAttributesW=1 and GetFullPathNameW=1)	Virus

Table 7.10: Classification Rules for Worm

Rules	Antecedent	Consequent
1	if(GetFileAttributesW=1)	Worm
2	if(GetLongPathNameW=1)	Worm
3	if(GetFullPathNameW=1)	Worm
4	if(GetLongPathNameW=1 and GetFileAttributesW=1)	Worm
5	if(GetFullPathNameW=1 and GetFileAttributesW=1)	Worm
6	if(GetFileAttributesW=1 and GlobalMemoryStatus=0)	Worm
7	if(GetFullPathNameW=1 and GetLongPathNameW=1)	Worm
8	if(GetLongPathNameW=1 and GlobalMemoryStatus=0)	Worm
9	if(GetFullPathNameW=1 and GlobalMemoryStatus=0)	Worm
10	if(GetFullPathNameW=1 and GetLongPathNameW=1 and GetFileAttributesW=1)	Worm
11	if(GetLongPathNameW=1 and GetFileAttributesW=1 and GlobalMemoryStatus=0)	Worm
12	if(GetFullPathNameW=1 and GetFileAttributesW=1 and GlobalMemoryStatus=0)	Worm
13	if(GetFullPathNameW=1 and GetLongPathNameW=1 and GlobalMemoryStatus=0)	Worm
14	if(GetFullPathNameW=1 and GetLongPathNameW=1 and GetFileAttributesW=1 and GlobalMemoryStatus=0)	Worm
15	if(GetFileAttributesW=1 and CreateThread=0)	Worm
16	if(GetLongPathNameW=1 and CreateThread=0)	Worm
17	if(GetFullPathNameW=1 and CreateThread=0)	Worm
18	if(GetLongPathNameW=1 and GetFileAttributesW=1 and CreateThread=0)	Worm
19	if(GetFullPathNameW=1 and GetFileAttributesW=1 and CreateThread=0)	Worm
20	if(GetFileAttributesW=1 and CreateThread=0 and GlobalMemoryStatus=0)	Worm

formed other models. Before this work, Sundarkumar and Ravi [121] reported an analysis on Malware dataset$_1$. They employed various algorithms like DT, SVM, GMDH, and MLP, etc., along with oversampling. In the Table 7.6, it can be observed that the proposed model outperformed other models. Similarly, for dataset$_2$ too we compared the accuracy values of earlier works. Ahmed et al. [117] employed Naive Bayes (NB), RIPPER, J4.8, SMO, and IBk algorithms for classification. The reported values using the NB classifier are as follows: for Trojan (96.6%), Virus (99%), and Worm (98.4%). Recent works of Sundarkumar et al. [123] also experimented with this dataset. They experimented with DT, SVM, MLP, GMDH, PNN, and Random Forest (RF) classifiers. It was reported that DT and SVM performed equally with respect to accuracy values, which were 98.46% and 98.63% respectively. It is very interesting to note that with the model proposed in this chapter an AUC value of 1 was obtained in all the cases of dataset$_2$, while the highest AUC of 0.875 was yielded by FP growth in the case of dataset$_1$. It is also remarkable that across all the 10 folds, identical top-20 rules were obtained in the case of both datasets. It shows that this model is not

only robust and stable but also outperforms other approaches proposed for this problem.

To further validate our results conclusively, statistical significance test viz., ($Wilcoxontest$) was proposed on the two proposed models i.e. LDA+Apriori and LDA+FP-Growth algorithms on the 10 fold results of two datasets. The statistic values indicate that that there is no statistically significant difference between them. Of particular significance is the fact that FP-Growth produced rules with lower fitness values compared to that generated by Apriori. Therefore, by taking this as a tie breaker, it is suggested that Apriori is preferable.

7.7 Conclusions

In this chapter, a novel hybrid model for text classification, i.e., topic modeling followed by Class Association Rule Mining algorithm, is proposed. The discriminative features were selected through the LDA weighting scheme. Here, LDA identified appropriate API calls efficiently, and these calls were used to discover the antecedents using class association rules. Then, the rules were ranked based on the fitness values and the top-20, top-30 rules were extracted. The efficacy of the proposed model was evaluated on malware_1 and malware_2 datasets. The proposed method demonstrated the best performance compared to traditional classifiers such as DT, SVM and other neural network architectures. Thus, it is concluded that the proposed hybrid model can handle malware detection efficiently.

Chapter 8

Predicting Indian Stock Market using the Psycho-linguistic Features of Financial News

This chapter presents a model for financial forecasting using news articles as a proxy for explanatory variables. It briefly describes the problem, the motivation behind the work and contributions made thereof. Later, it elaborates the proposed approach, and then, the results are analyzed.

8.1 Introduction

Stock Market prediction is an interesting research problem as stock value always varies significantly with respect to time. Time series data is noisy and chaotic. Any forecasting model that finds the intricate relationship between the financial news about a company and its stock price is useful. Future stock values are predicted using the financial news about that company. Especially, the outcome of the prediction [264] will have a direct bearing on future decision making such as fresh investment on sale or status-quo of the stocks.

Despite proliferating research in the field, forecasting future stock prices is a complicated process since stock market exhibits a dynamic trend coupled with chaotic and/or random behavior. Forecasting stock price based on relevant news

articles is even more difficult. It is well-known that the raw text data is not useful for any data mining task. Therefore, we convert the text into structured intermediate form called Document-Term Matrix using which, one can perform syntax-based document classification based on some tokens or features. But, in sentiment or opinion mining tasks these syntactical features do not play a significant role. Semantic features are helpful for understanding the customer behavior/ opinion analysis/ sentiment analysis. Extraction of semantic features, where each feature maps to an opinion word poses a significant challenge. There are various methods available for extraction of semantic features. Few of them are, OpinionFinder, Linguistic Inquiry and Word Count (LIWC), Google Profile of Mood States (GPOMS), SentiWordNet, R sentiment analysis and Python NLP package, etc. The features extracted by these tools are based on opinions, writing style and mood in which a particular article was written. This approach alleviates the problem faced by syntactic features in not being able to present the hidden semantic meaning of the text that can represent a real pattern. However, LIWC & TAALES software extract psycholinguistic features unlike other methods mentioned above.

8.2 Motivation and Contributions

In literature, there is a huge number of stock prediction models that deal with only numeric data under time series analysis framework and comparatively not much work is reported in stock prediction using text mining of financial news articles. So far, models were built using only news headlines that have limited text with no details of the entire information. It is evident from the news that the news article contains more details instead of news headlines. Therefore, the sentiment of a news article can be a useful predictor for forecasting a stock. Therefore, in this chapter, an attempt is made to predict the stock price of a company using the information contained in news articles related to the particular company in question. We conjecture that a correlation exists between news and the stock values. The sentiment present in news articles contains useful information about stock price forecasting. The contributions of this chapter are:

1. Extraction of psycholinguistic features from the financial news articles concerning Indian companies. These features collectively convey the sentiment hidden in the article.

2. Extraction of lexical sophistication features from financial news articles.

3. Imputing missing linguistic/ lexical feature values for the cases where the stock price is available for a company but the corresponding news is not.

4. Developing stock prediction models with these features as predictor variables using a host of intelligent techniques.

8.3 Related Work

Financial markets drive a lot of investment decisions all over the world. Stock markets witness dramatic changes over time in response to the geo-political, social and fiscal changes globally. These in turn trigger financial risks in investments, with the investors and the financial institutions being the stakeholders. Consequently, researchers started studying the cause and effect relationship between various market factors and the corresponding movements in stock prices. Most of the works focused on quantitative data like historical prices as predictor variables to predict the stock price. Less attention was paid to the use of the enormous amount of unstructured textual data generated from the web in the form of published news articles, public opinions in social media and blogs by experts in the field of financial investments. In this section, the past works of investment risk modeling and market predictions using this unstructured data is briefly reviewed. Engle and Ng [265] predicted volatility in the stocks using news. Autoregressive Conditional Heteroskedasticity (ARCH), Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models were fitted on stock returns of Japan from 1980 - 1988. They concluded that the impact of volatility due to negative news is higher than positive news for stock returns. Lavrenko et al. [266] presented a model to identify the news stories which affect the trend of financial markets. They identified patterns in time series with the help of piecewise linear fit followed

by label assignment with an automated binning process. They concluded that specific stock related news is useful for analysis as compared to the global news.

Thomas and Sycara [267] worked on discerning the behavior of financial markets. They hypothesized that textual information available on the website of a company impacts its business. They proposed two models based on maximum entropy and genetic algorithm to predict financial markets. They concluded that the combination of these two models outperformed the stand-alone models. Then, Peramunetilleke and Wong [268] proposed a new model for forecasting exchange rates based on the current status of world financial markets. The study investigated on how news headlines of the financial market could be helpful for forecasting the currency exchange rates. They concluded that the proposed approach was better than random guessing and suggested that hybrid models perform better prediction. Koppel and Shtrimberg [25] proposed a model based on news articles for stock price prediction. They extracted the features from the Multex Significant Development corpus and predicted the Standard & Poor 500 (S&P 500) stock index. Before the process of modeling, they labeled the news as positive or negative according to their impact on the price. Later, they employed SVM to train the news articles and reported an accuracy of 70%.

Rachlin et al. [269] proposed a model called ADMIRAL, based on textual information of web documents and time series data. They employed automatic extraction of text instead of predefined expert list. They explained the functionality of ADMIRAL in six steps: Data collection, feature extraction, term weighting, combined data construction, classification using decision tree (DT) and market recommendation. They acquired the data from online sources, like Forbes and Reuters. They reported an accuracy of 83.3% with DT on both the datasets. Zhai et al. [270] presented a model for stock price prediction using news and technical indicators as explanatory variables and SVM for classification. They considered the daily share prices of BHP Bilton Ltd. from Australian Stock Exchange as output. The proposed method yielded higher directional prediction accuracy of 70% compared to specific models considering news alone or technical indicators alone. Mahajan et al. [46] analyzed the impact of news on the stock market. They analyzed the actual market data with news to understand the impact on the SENSEX market. The events were identified by employing Latent Dirichlet

Allocation (LDA) based topic extraction method. Then, by using a hybrid model developed by combining the DT and SVM, they were able to report a prediction accuracy rise or fall of 60%.

Evans and Lyons [16] also experimented with macro news for studying the currency flow. In this work, they observed that the arrival of macro news could account for more than 30% of daily price variance. They experimented with US News and FX rates with standard error as a performance metric. They concluded that macro news impacted two-thirds of the directional movement and exchange rates. Butler and Keselj [49] presented a model based on N-gram analysis for financial forecasting. They constructed various models using character n-gram, word n-gram, a hybrid of readability model with SVM and a hybrid of readability and n-gram. They predicted closing values of the S&P 500 companies with the help of the textual information present in their annual reports. They concluded that hybrid model of character n-gram yielded the best performance compared to other models with respect to the percentage of returns. Bollenet al. [50] proposed a model for stock prediction using Twitter tweets. The opinions are extracted from the tweets using OpinionFinder and GPOMS tools, where OpinionFinder consists of positive or negative opinion and GPOMS consists of 6 mood states namely alert, calm, happy, kind, sure, and vital. They employed a self-organizing fuzzy neural network for prediction of the Dow Jones Industrial Average values. They predicted the Up and Down values of the stock (closing values) with an accuracy of 87.6%. They concluded that through this approach the MAPE value was reduced by more than 6%.

Groth and Muntermann [53] published work in the field of intra-day market risk management by using textual data analysis to discover patterns that can explain risk exposure. Different learners used include Naive Bayes, k-Nearest Neighbor, Neural Network, and Support Vector Machine trained on processed feature datasets, followed by traditional measures of evaluation namely accuracy, recall, precision and F-measure as well as domain specific simulation-based model evaluation. The results clearly support the influence of textual information in financial risk management. Chan and Franklin [32] proposed a novel text-based decision support system which extracts event sequences from text patterns and

predicts the likelihood of the occurrence of events using a Hidden Markov Model-based inference engine. They investigated more than 2000 financial reports with 28,000 sentences. Experiments showed that the prediction accuracy of the model outperformed similar statistical models by 7% for the seen data while significantly improving the prediction accuracy for the unseen data. Further comparisons substantiate the experimental findings. Li et al. [271] proposed a model for stock market prediction by integrating quantitative and qualitative information. They collected the news articles during the Hong Kong Stock Exchange trading time. After pre-processing of text, they generated tf-idf matrix and applied Chi-square feature selection method to find out prominent features. Then, they employed NaiveBayes (NB), Multi-Kernel Learning (MKL) and SVM algorithms on these features. They concluded that MKL outperformed other models.

Vu et al. [17] proposed a model for predicting stock price up and down movements based on Twitter messages. Initially, they labeled the sentiment into two categories – positive and negative. Based on this, they predicted the stock price of four companies viz., Amazon, Apple, Microsoft, and Google with 41 days' data using Decision Tree;reported accuracies values are 75%, 82.93%, 75.61% and 80.49% respectively. Hagenau et al. [55] proposed the use of robust feature selection approaches for stock price prediction. Chi-square and bi-normal separation methods were used to select semantically relevant features, and hence improve classification accuracy for financial stock prediction. Initially, they classified the news articles and then they constructed the prediction model. They experimented on German, UK announcements with stock values available in Datastream. They built various models incorporating various feature subset selection methods viz., single words, 2-Gram, 2-word combination using SVM. They concluded that with 2-word combinations they achieved an accuracy of 76%. Jin et al. [18] proposed a model called Forex-foreteller which mines news articles and forecasts the movement of foreign currency markets. A combination of language models, topic clustering, and sentiment analysis was used to identify the relevant news articles. These were combined with historical stock index and currency exchange values for prediction. They employed linear regression for currency forecasting. They experimented with the exchange rates of Argentina, Brazil, Chile and Columbia currencies versus US Dollar value. They concluded that, with this proposed model

they obtained higher recall values of 0.6, 0.63, 1 and 1 respectively compared to precision values.

The effect of macro news on upward and downward movements of Forex is studied by Chatrath et al. [20]. They employed multivariate regression in this approach. They investigated the effect on currencies of UK, Japan, Swiss and Euro on the arrival of news. They observed that US announcements are directly linked to nearly 15% of the currency jumps. They concluded that 56% of currency change happens within the 5 min of news arrival. Li et al. [272] presented a work based on Extreme Learning Machine (ELM) for stock market prediction. They carried out the experiments on 23 stocks of the H-share (Chinese) market and corresponding news. By considering the news articles and stock prices, they employed SVM, Neural Network, etc., for comparison with the proposed approach. They concluded that the ELM approach outperformed other techniques. Forex market prediction using news headlines as predictors was experimented by Nassirtoussi et al. [36]. They proposed a multilayer architecture consisting of semantic abstraction, sentiments aggregation, and dynamic model creation. They concluded that their approach yielded an accuracy of 83.33%.

Shynkevich et al. [273] presented a framework to predict the stock movements using Multiple Kernel Learning (MKL). They extracted the news from LexisNexis source and categorized the news based on their relevance to the stock, industry, and sub-industries, etc. After preprocessing, they applied chi-square method for feature selection on tf-idf matrix. For experimental purpose, they considered S&P 500 index stocks in Health sector. They employed SVM, k-NN, and MKL techniques. They concluded that (i) the predictive performance of all the models is better due to the various types of news sources. (ii) Proposed MKL method performed better than other two models.

According to the behavioral economics, moods, sentiments and emotions play a significant role in investors' decision-making process. Ho and Wang [274] developed a model for predicting stock market movement using Artificial Neural Networks (ANN). They experimented on stock prices of Google (NASDAQ:Google) and news articles of Dow Jones for predicting upward and downward movement of the stocks. They evaluated the model with prediction rate, sensitivity, and specificity. They concluded that the proposed model is better than Random walk

forecast method. It is evident from the overview of past works in stock market prediction that various information sources are combined to produce a joint feature set. However, they may not provide valid information for assessing the effect of each source on the stock. To overcome this difficulty, Li et al. [275] proposed a framework using Tensor methods for stock market prediction. Through this approach, they could capture the essential information among multiple sources. They experimented with the data sources like CSI 100 stocks, financial discussion boards, and news reports. The performance evaluation was carried out with Directional Accuracy (DA) and Root Mean Square Error (RMSE) values.

Literature abounds with several approaches for solving this problem. While our proposed method resembles some of them, it differs from all of them in that we extracted psycho-linguistic and lexical features from financial news and fed them as input to the prediction models. The various works are summarized in below Table 8.1.

8.4 Proposed Method

In this work, a hybrid model which performs text mining on the financial news articles and forecasting of the stock price in tandem, is proposed. The proposed methodology consists of three phases namely preprocessing, imputation and forecasting as depicted in Figure 8.1. Initially, all news articles and the corresponding stock prices of a set of companies were collected. Datasets' description can be found in the Section 8.5.1. Later, the news articles were preprocessed by employing LIWC [276] and TAALES [277] software. The output, i.e., the documents and their corresponding set of linguistic feature values, was captured in a structured format. Stock value, the target variable, was appended to the matrix obtained by using LIWC tool, where the columns of the matrix were used as predictor variables. Details about the linguistic features and LIWC are presented in Section 8.4.1. Same process was repeated with the features extracted using TAALES. LIWC and TAALES were recently employed by Ravi and Ravi [278] for irony and satire detection in news and textual corpora. In imputation phase, initially the missing records i.e., examples where the stock value is present, but corresponding news articles with respect to the particular stock is not available were found.

Table 8.1: Summarization of Related works on Stock Market Prediction

Study	Dataset	Model	Performance Measure
Engle et al. [265]	News	ARCH/ GARCH	Stock return
Lavrenko et al. [266]	Yahoo News, Yahoo Stock	Linear Regression	Gain/ Loss
Peramunetileke and Wong [268]	Financial News	Rule Classifier	Up/ Down
Koppel and Shtrimberg [25]	S & P News	SVM	Accuracy
Rachlin et al. [269]	Financial News	DT	Up/ Down
Zhai et al. [270]	Financial Reviews	SVM Kernels	Up/ Down
Mahajan et al. [46]	Financial News	Hybrid Classifiers	Up/ Down
Evans and Lyons [16]	Macro News	Heteroskedasticity	Up/ Down
Butler and Keselji [49]	Annual Reports	SVM	Good/ Bad
Bollen et al. [50]	Tweets	SOFNN (Hybrid)	Error Rate
Groth and Muntermann [53]	Announcements	ANN,k-NN, NB, SVM	Up/ Down
Chan and Franklin [32]	Bloomberg.com, Quamnet.com	HMM, DT	Sensitivity
Li et al. [271]	Stock Exchange News	NB, SVM, Multi-Kernel Learning (MKL)	Buy/ Sell
Vu et al. [17]	Tweets	DT	Up/ Down, Accuracy
Hagenau et al. [55]	Corporate News	SVM, SVR	Rise/ Down
Jin et al. [18]	Financial News	LR	Precision/ Recall
Chatrath et al. [20]	Financial News	MV Regression	Mean, SD
Li et al. [272]	News Articles	SVM, NN, ELM	Accuracy
Nassirtoussi et al. [36]	News Headlines	Multilayer model	Accuracy
Shynkevichet et al. [273]	LexisNexis source	SVM, k-NN, MKL	Accuracy, Return value
Ho and Wang [274]	Dow Jones News	ANN, Random walk forecast	Prediction rate
Li et al. [275]	CSI 100 stocks, news	Tensor Methods	Directional Accuracy, RMSE

Then, the neighbors of a stock value within a range of 10% were selected, and imputation was performed with respect to stock value using mean. In the final phase, i.e., modeling, initially the regression models were built with all features. Later, two feature selection methods namely Chi-square and MRMR were applied for identifying the discriminative features. Finally, experiments were conducted with top-10 as well as top-25 feature subsets. Finally, MAPE and NRMSE values were reported for performance evaluation.

The formal definition of the proposed model is as follows: Let $Y = \{y_1, y_2, ..., y_k, y_{k+1}, ..., y_N\}$ be a stock market time series and $F = \{d_1, d_2, ..., d_k, d_{k+1}, ..., d_N\}$ be the financial articles at time $t = \{1, 2, ..., k, k+1, ..., N$ respectively. Let M be a prediction model, that is fit using $y_1, y_2, ..., y_k$ and $d_1, d_2, ..., d_k$. The problem is to predict $y_{k+1}, ..., y_N$ using the fitted model M.

8.4.1 Linguistic Features

The Linguistic Inquiry and Word Count (LIWC) software was employed to extract the linguistic features from the news articles. LIWC [279] includes the text analysis module along with a group of built-in dictionaries that is used to count the percentage of words reflecting different emotions, thinking styles, social concerns, and even parts of speech. LIWC counts the words which are psychologically meaningful. It contains a dictionary of 6400 words [280]. These dictionaries again contain sub-dictionaries. The output of LIWC contains 93 variables. These variables belong to one of the following groups: General description, the summary of language, linguistic, personal concern, physiological and personal.

8.4.2 Lexical Sophistication Features

TAALES (Tool for the Automatic Analysis of Lexical Sophistication) was employed to extract lexical features from the news articles. TAALES [281] calculates various indices, and these are related to the frequency of the words and its ranges, n-gram frequency, word neighbors, strength association between the words, psycholinguistic properties of the words, word recognition norms (standard deviation), polysemy, Mutual information, etc. There are two versions of

this software viz., TAALES 1.0 and TAALES 2.0 [277]. TAALES 1.0 consists of 103 indices, whereas TAALES 2.0 consists of 424 indices. The output of TAALES 2.0 contains 241 variables. These indices are created from the British National Corpus.

The proposed methodology employing LIWC, TAALES and various regression models are described in the following Algorithm 8.1.

Algorithm 8.1 Psycholinguistic_Stock

Input: Text documents consisting of news articles

Output: Predicted stock values

1: Do Text Preprocessing on text documents using LIWC and TAALES
2: Form the *Document Term Matrix*
3: Append the stock value (target variable) to DTM
4: Find the missing records where the stock value is present and corresponding news are not
5: Select the neighbhoring records with in $\pm 10\%$ range of the stock value
6: Impute the missing predictor variables by mean value of neighbhoring records
7: Apply the feature selection methods (Ch-square, MRMR) on imputed data
8: Apply the various regression models (SVR, MLP, GMDH, RF etc.)
9: Compute the MAPE, NRMSE values

8.5 Data, Techniques and Performance Measures Used

8.5.1 Datasets Used

The data used in the proposed experiments was collected through "Business Standard" online news resource. This involved collecting news articles for 12 major Indian companies including, Bharti Airtel Limited, Mahindra & Mahindra Limited, Tata Consultancy Services Limited (TCS), Tata Motors Limited, Reliance Industries Limited, Tata Steel Limited, State Bank of India (SBI) and Oil and Natural Gas Corporation (ONGC). The corresponding historical stock prices were extracted from the "Yahoo Finance India" online web resource [282]. The tools

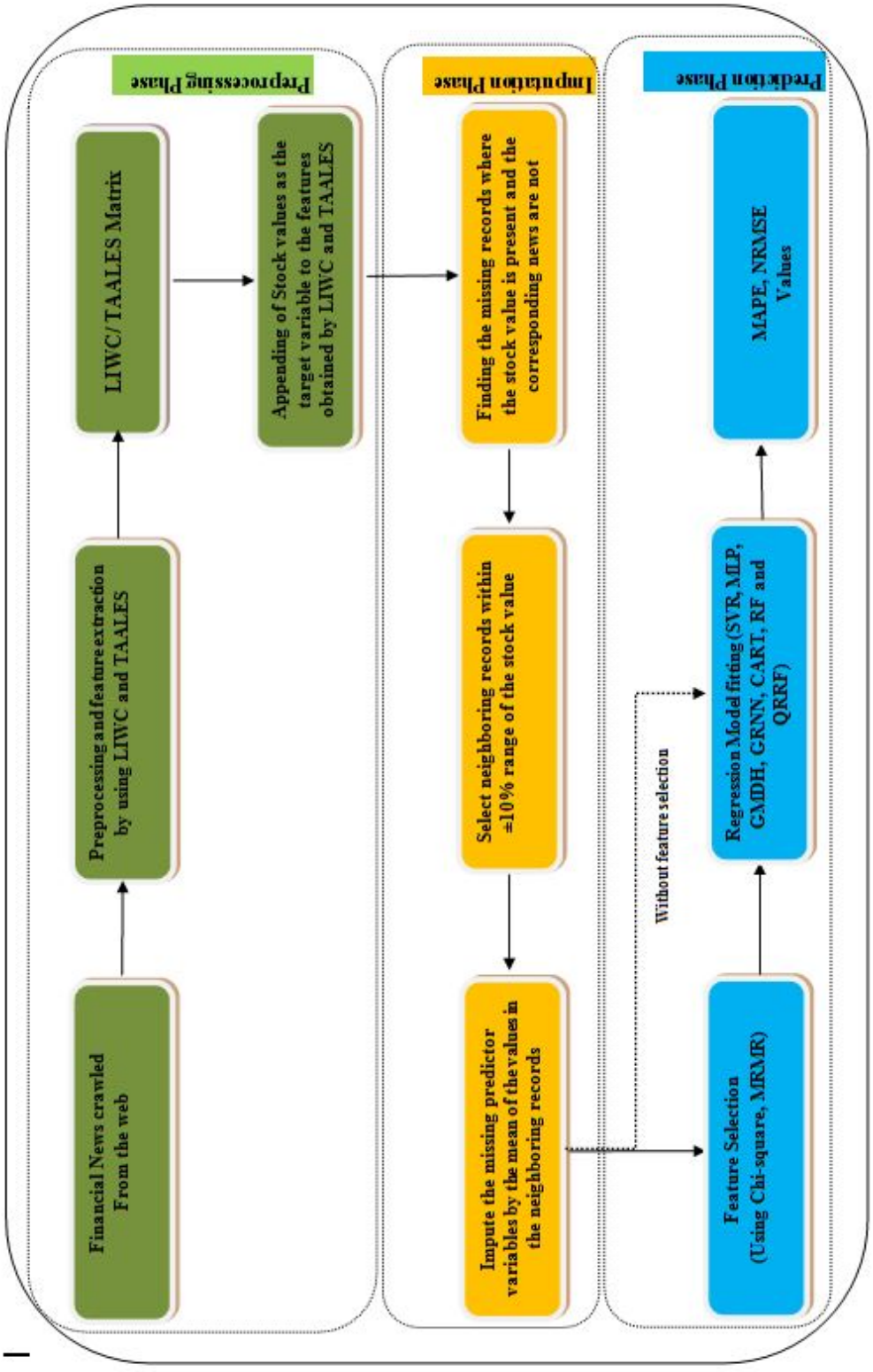

Figure 8.1: Schematic Diagram of Proposed Methodology

used for web crawling were "WebScraper" [283]. The description of datasets is outlined in Table 8.2.

Table 8.2: Distribution of News Articles with Respect to the Company

S.No.	Dataset	Number of articles	Period of data availability
1	Bharti Airtel	85	28^{th} January 2016 to 25^{th} May 2016
2	Mahindra	125	1^{st} December 2015 to 31^{st} May 2016
3	Tata Motors	108	26^{th} November 2015 to 22^{nd} April 2016
4	Reliance Industries	116	29^{th} November 2015 to 9^{th} May 2016
5	Tata Steel	131	2^{nd} December 2015 to 30^{th} May 2016
6	TCS	126	10^{th} December 2015 to 27^{th} May 2016
7	SBI	126	7^{th} December 2015 to 30^{th} May 2016
8	ONGC	125	22^{nd} December 2015 to 31^{st} May2016
9	Infosys	130	22^{nd} December 2015 to 15^{th} June2016
10	Sun Pharma	129	11^{th} December 2015 to 5^{th} June2016
11	Spice Jet	131	21^{st} December 2015 to 24^{th} May 2016
12	Jet Airways	125	14^{th} December 2015 to 3^{rd} June2016

8.5.2 Data Imputation Process

In today's world, handling incomplete data is a very common difficulty in most of the datasets. There are various causes for missing data: weak data acquisition process, data privacy issues, non-availability of data and many other reasons. It leads to uncertainty in the dataset and causes inaccurate prediction. So, here imputation plays a significant role. Imputation is defined as the process of replacing missing values with substituted values. Imputation plays a significant role in datasets in various fields, including financial, speech processing and medical diagnostics, etc. In a dataset, it is necessary to know the reason for missing data. There are various types of missing data. Little and Rubin [284] categorized the missing data into three categories namely

1. MCAR (Missing Completely at Random): According to MCAR, the missing data mechanism is unrelated to values of any other variable in the dataset.

2. MAR (Missing at random): MAR mechanism is involved when the probability of missing values corresponding to a particular variable is related to some other variable in the dataset but not with the variable itself.

3. MNAR (Missing Not at Random): According to MNAR, the missing values on a variable are related to the variable itself and not on other controlled variables in the dataset.

Conventional imputation methods include mean imputation and regression imputation. Multiple imputation methods involve replacing missing value with a set of plausible values. These imputed datasets are analyzed by using standard procedures. Nishanth and Ravi [285] proposed mean imputation followed by running probabilistic neural network for imputation and tested its effectiveness on a set of benchmark problems. Earlier, Gautam and Ravi [286] proposed two models for data imputation based on Counter Propagation Auto-Associative Neural Network (CPAANN) and Grey System Theory with CPAANN. Then, Ravi and Krishna [287] proposed a hybrid model for data imputation using mean imputation followed by General Regression Auto Associative Neural Network (GRANN) or Particle Swarm Optimization based Auto Associative Neural Network (PSOANN). They concluded that GRANN outperformed other models on four benchmark datasets. However, we employed an imputation method different from the above.

News articles corresponding to a particular company are not published every day. It creates gaps in the time series values of the LIWC/ TAALES feature scores. Hence, in this scenario data imputation plays a significant role. Before we present the imputation procedure, some data preprocessing steps employed in this work are noteworthy. In some cases, it was found that multiple news articles were available for a particular company on the same day. Consolidated LIWC/ TAALES feature scores for that date could be calculated by averaging the individual LIWC/ TAALES scores of all the news articles published on that day. Further, it is a known fact that stock markets remain closed on weekends i.e. Saturday and Sunday, and on public holidays. This leads to a situation where news articles are available on a particular day when there is a stock market holiday. Losing this data will lead to information loss. To retain such feature values, all these data instances with no available stock prices on the corresponding

date are merged into next instance with available stock price. This is done by averaging out values of all these instances till the date where next stock price information is available.

In this approach, initially the records where the stock value is available, and the corresponding news is not available or missing is found. For every record with a given stock price and missing financial news, the missing feature values are imputed as follows:

(i) Pick all those records whose stock price is within ±10% of the current stock price.

(ii) Compute the mean of all the feature values of the records so chosen, in order to form a new feature vector.

(iii) Finally, this feature vector acts as a proxy for the missing financial news.

(iv) For imputation, the method of Ling and Mei [288] was adopted.

Similar works are also found in the literature on imputation (Patilet al. [289], Garcia-Laencianaet al. [290]. The missing values are finally imputed using the following formula.

$$x_i^{'} = \sum_{i=1}^{n} \frac{x_i}{d_i} \tag{8.1}$$

Where, x_i' is the imputed value for the missing i^{th} attribute, x_i is the attribute value obtained in the step (ii) above and d_i represents the absolute difference between the corresponding stock value of missing record and that obtained in step (ii) above.

8.5.3 Performance Measures Used

In this work, the performance of the proposed system was evaluated with the following metrics: Mean Absolute Percentage Error (MAPE) [291] and Normalized Mean Square Error (NRMSE). The stock value varies from one company to another company. Hence Mean Squared Error (MSE) value was not considered as a

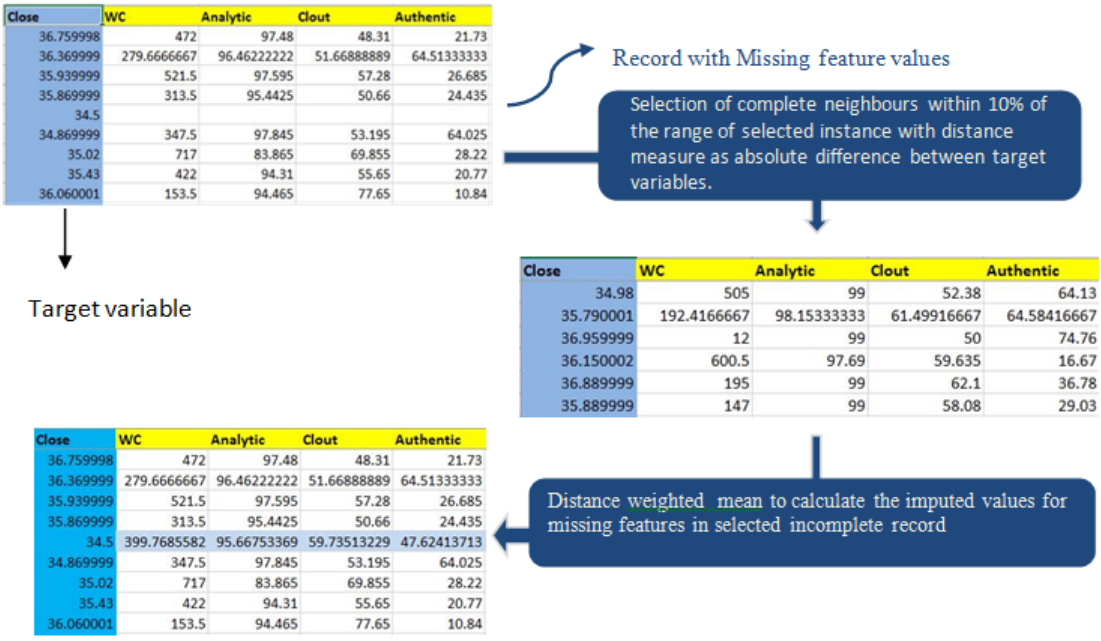

Figure 8.2: Data Imputation Process

performance metric for this study. For uniform scaling, only the aforementioned metrics were employed.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} * 100 \tag{8.2}$$

$$NRMSE = \frac{1}{Range}\sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{8.3}$$

Here, Range = Max - Min.

8.5.4 Tools and Techniques

A machine with Intel i5 processor, 2.6GHZ, 8GB RAM, 500GB HDD and 64-bit Windows 8 OS was used for experiments and modeling. $R^{®}$ language packages [261] was used for fitting RF, QRRF, RPART and SVR models. GMDH, GRNN models were implemented using *Neuroshell*®. Similarly, for MLP, Statistica trial

version [292] was used. Various parameter settings configured for each model are listed in Table 8.4. Tools employed for this work are listed in the Table 8.3.

Table 8.3: Tools and Techniques Used

Technique Used	Used for	Tools
Crawling	NEWS extraction	*WebScraper*®
RF/ QRRF/ RPART/ SVR	Regression	*R*®
GMDH/ GRNN	Regression	*Neuroshell*®
MLP	Regression	*Statistica*®

Table 8.4: Parameter Settings for Various Models

RPART	
Parameter	**Value**
Minsplit	20
minbucket i.e. round(minsplit/3)	7
complexity parameter (cp)	0.01
Maxcompete	4
maxsurrogate	5
Usesurrogate	2
Maxdepth	30
SVM	
Type	eps-regression
Kernel	radial
Cost	1
Epsilon	0.1
no. of support vectors	78
Random Forest	
ntree	1000
node size	5
maximum nodes	83
QRRF	
No.of trees	200
No.of variables used for split	5
MLP	
Hidden units	4

Continued on next page

Table 8.4 – continued from previous page

Parameter	Value
Max hidden units	13
Networks to train	20
Networks to retain	5
Error function	Sum of squares
Activation function	Tanh
Cycles	200
Learning rate	0.1
Momentum	0.1
GMDH	
Scale function	Linear(-1,1)
type	advanced
Maximum variables in connection	$x_1x_2x_3$
Maximum product terms in connection	$x_1x_2x_3$
Max variable degree connection	x_3
Selection criterion	GCV
Type of schedule	asymptotic
Optimization of the model	full
missing values treated as	error
GRNN	
Smoothing factor	0.3
Scaling function	Linear[0,1]
distance	Vanila(euclidean)
caliberation	Genetic, adaptive
Genetic breeding pool size	300
Auto termination of the generations with no improvement of 1%	20
Missing values treated as	error

8.5.5 Feature Subset Selection Methods

Feature subset selection is an important task in any text mining task. In this work, the following two feature subset selection methods were used.

(i) **Chi-square:** Chi-squared test helps us decide whether a categorical predictor variable and the target class variable are independent or not. High Chi-squared values indicate the dependence of the target variable on the

predictor variable. It is employed in many text mining applications (Zheng et al. [293]).

(ii) **Minimal Redundancy and Maximal Relevance (MRMR):** Minimum Redundancy Maximum Relevance (MRMR) [294] feature selection method uses a heuristic to minimize redundancy while maximizing relevance to select promising features for both continuous and discrete datasets. The maximum relevance condition is obtained through features' F-statistic values. For further details, the reader is referred to Peng et al. [294], Ding and Peng [295].

8.6 Results and Discussion

The Linguistic Inquiry and Word Count (LIWC) software was employed to extract the linguistic features from the news articles. LIWC includes (Tausczik and Pennebaker [279]) a text analysis module along with a group of built-in dictionaries which is used to count the percentage of words reflecting different emotions, thinking styles, social concerns, and even parts of speech. LIWC counts the words which are psychologically meaningful. It contains a dictionary of 6400 words (Pennebakeret al. [280]). These words again contain sub-dictionaries. The output of LIWC contains 93 variables; these variables belong to one of the following groups: General description, the summary of language, linguistic, personal concern, physiological, personal.

Similarly we employed TAALES for extracting lexical sophistication features. The output contains 241 variables. The results of all models with LIWC and TAALES features are presented in the following cases,

(i) Full features

(ii) Chi-square top-25 features (Ch-25)

(iii) Chi-square top-10 features (Ch-10)

(iv) MRMR top-25 features (MRMR-25)

(v) MRMR top-10 features (MRMR-10)

For all datasets, the results with LIWC features are presented in Table 8.5 through 8.9 . In these tables, cells highlighted in * indicate the best performance of the model under consideration vis-à-vis other models.

Table 8.5 presents the results in terms of MAPE and NRMSE corresponding to various prediction models without feature selection using LIWC features. GMDH outperformed all other techniques in terms of both MAPE and NRMSE, in all but two datasets (ONGC, Spice Jet). For these datasets, GRNN performed well. Table 8.6 presents the results of various models fed by the top-25 features obtained by Chi-square feature selection method using LIWC features. It can be observed from the table that GMDH yielded the best predictions in all datasets except SBI, ONGC, SpiceJet and Jet Airways. For these four datasets, GRNN outperformed all other techniques. It is to be noted that in case of Mahindra, Tata Motors and Reliance Industries datasets, Chi-square value returned is 0 for most of the features. It means that they have no impact on prediction. Therefore, the results of these datasets are not presented in Table 8.6. Table 8.7 presents the results of the models with top-25 features selected by MRMR method using LIWC features. In this combination, GMDH performed the best in terms of MAPE and NRMSE on all datasets except SBI, ONGC, Infosys, Sun Pharma, Spice Jet and Jet Airways datasets; for these datasets, GRNN performed the best. Table 8.8 presents the results of the models trained with top-10 features selected by Chi-square method using LIWC features. From this table, it can be observed that GMDH outperformed all other techniques on the datasets of Airtel, Mahindra, Tata Motors (8 features), TCS, Infosys, Sun Pharma. Whereas, GRNN could yield the best predictions on Reliance Industries, Tata Steel, SBI, ONGC, Spice Jet, and Jet Airways in terms of both MAPE and NRMSE values. Interestingly, GMDH and GRNN performed almost identically on Tata Steel. Table 8.9 presents the results with top-10 features selected by MRMR method using LIWC features. The table shows that GRNN outperformed other models in terms of both MAPE and NRMSE on seven companies' stocks (Tata Motors, Reliance Industries, SBI, ONGC, Infosys, SpiceJet, and Jet Airways) and GMDH performed the best on the remaining five datasets. Further, the features (LIWC)

Table 8.5: Stock Prediction Results with All Features from LIWC Features

Dataset	GMDH		GRNN		RF		QRRF		RPART		SVR		MLP	
	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE
Airtel	**0.067***	**0.014***	1.150	0.48	4.89	0.806	3.11	0.510	5.56	0.925	9.69	0.89	2.77	0.513
Mahindra	**0.61***	**0.025***	9.49	0.36	16.99	0.617	13.31	0.469	17.99	0.641	15.66	0.565	7.709	0.375
Tata Motors	**0.367***	**0.045***	0.485	0.051	7.111	0.593	5.153	0.455	9.108	0.809	6.66	0.576	6.155	0.557
Reliance	**0.182***	**0.026***	0.269	0.038	2.589	0.355	2.070	0.284	3.006	0.415	2.196	0.314	2.723	0.347
Tata Steel	**0.422***	**0.039***	0.496	0.051	12.82	1.030	7.47	0.662	10.159	1.086	10.247	0.921	8.986	0.817
TCS	**0.136***	**0.0378***	0.524	0.169	3.37	0.878	3.26	0.780	4.72	1.371	3.22	0.841	2.610	0.727
SBI	**0.292***	**0.024***	0.533	0.045	4.954	0.371	3.206	0.239	9.106	0.691	4.855	0.374	5.45	0.360
ONGC	0.360	0.067	**0.209***	**0.068***	1.446	0.235	0.829	0.152	2.329	0.443	1.21	0.19	1.780	0.298
Infosys	**0.131***	**0.024***	0.846	0.186	3.52	0.597	3.11	0.536	2.785	0.475	3.553	0.621	2.113	0.411
Sun Pharma	**0.104***	**0.012***	0.527	0.088	2.472	0.317	1.986	0.284	2.779	0.333	2.715	0.358	3.057	0.360
Spice Jet	**0.192***	**0.007***	0.215	0.009	3.385	0.173	2.134	0.152	4.338	0.213	2.709	0.146	0.705	0.045
Jet Airways	0.318	0.028	**0.244***	**0.019***	3.195	0.279	2.428	0.214	6.485	0.533	2.598	0.216	4.549	0.332

selected through Chi-square and MRMR methods are presented in Table 8.10. The models that are statistically significant compared to case (i) (full features case) are highlated.

From Table 8.10, it can be inferred that the psycholinguistic features having highest frequency of occurrence across different data sets are as follows: achieve, Analytic, male, relig, Comma and QMark.

The excellent performance of GMDH in most of the datasets is attributed to the fact that it is one of the earliest Deep learning neural networks thereby possessing very high predictive power. The non-parametric regression, which is at the heart of the GRNN does the trick for its second best performance behind GMDH. In order to determine the usefulness of feature subset selection methods employed here with LIWC features, we conducted a statistical significance test called Diebold-Mariano (DM) test [296] between **Case (i)** of GMDH (LIWC) and all other cases of GMDH (LIWC) in a pair-wise manner for all datasets except ONGC and Jet Airways. For these 2 datasets, DM test was performed between **Case (i)** of GRNN (LIWC)and all other cases of GRNN (LIWC) in a pair-wise manner. GMDH and GRNN were chosen because of their superior performance over other models in terms of MAPE and NRMSE as seen in Tables 8.5 through 8.9. The code for the test is available at [297]. The DM test values are reported (with LIWC features) in Table 8.11.

As seen in Table 8.11, the absolute value of the DM statistic (Chen et al. [298]) is less than 1.96 in the following cases: Tata Motors (case (iii) and (iv)), Reliance Industries (case (iv)), Tata Steel (case (ii)), TCS (case (ii)) and Spice Jet (case (v)) datasets. It indicates that there is no statistically significant difference between GMDH (case (i)) and GMDH (cases mentioned above) or GRNN (case (i)) and the GRNN (cases referred to above) as the case may be at 5%level of significance. Therefore, for these datasets, the corresponding cases of feature subset selection methods turned out to be better than the case (i) in terms of MAPE and NRMSE. However, in the rest of the cases in Table 8.11, the absolute of DM statistic is greater than 1.96 which indicates that case (i) of full features is statistically significantly better than all feature subset selection cases in terms of MAPE and NRMSE at 5%level of significance

Table 8.6: Stock Prediction Results with Ch-25 Features from LIWC Features

Dataset	GMDH		GRNN		RF		QRRF		RPART		SVR		MLP	
	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE
Airtel	**0.934***	**0.165***	1.101	0.296	4.06	0.61	2.31	0.420	4.29	0.794	3.49	0.634	2.77	0.546
Tata Steel	**0.806***	**0.098***	1.060	0.117	12.186	0.971	7.327	0.697	20.908	1.758	9.142	0.854	8.257	0.694
TCS	**0.226***	**0.066***	0.505	0.166	2.956	0.830	2.336	0.707	4.331	1.291	2.723	0.767	2.992	0.728
SBI	0.918	0.072	**0.838***	**0.071***	4.94	0.355	3.257	0.266	9.66	0.697	4.52	0.378	5.45	0.360
ONGC	0.796	0.123	**0.232***	**0.072***	1.65	0.278	0.83	0.157	3.37	0.621	1.33	0.235	2.25	0.332
Infosys	**0.567***	**0.097***	0.865	0.181	3.665	0.63	2.95	0.531	2.536	0.408	3.87	0.721	1.912	0.402
Sun Pharma	**0.260***	**0.030***	0.603	0.095	2.407	0.318	2.137	0.282	2.325	0.276	2.216	0.297	2.848	0.301
Spice Jet	0.909	0.035	**0.614***	**0.037***	3.371	0.173	2.830	0.144	4.414	0.210	2.910	0.151	2.462	0.155
Jet Airways	1.040	0.075	**0.604***	**0.053***	2.846	0.246	2.11	0.196	5.81	0.48	2.89	0.244	4.121	0.312

Table 8.7: Stock Prediction Results with MRMR-25 Features from LIWC Features

Dataset	GMDH		GRNN		RF		QRRF		RPART		SVR		MLP	
	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE
Airtel	**0.355***	**0.069***	0.614	0.149	4.45	0.761	2.83	0.511	4.34	0.737	3.68	0.666	2.202	0.462
Mahindra	**1.42***	**0.054***	6.49	0.247	14.19	0.510	12.61	0.455	17.39	0.615	16.12	0.58	12.43	0.455
Tata Motors	**0.618***	**0.061***	0.679	0.067	6.567	0.553	4.94	0.449	5.99	0.629	6.94	0.600	4.004	0.402
Reliance	**0.435***	**0.066***	0.698	0.115	2.516	0.351	2.129	0.299	3.346	0.426	2.139	0.312	2.327	0.291
Tata Steel	**0.654***	**0.064***	0.786	0.075	11.311	0.927	7.130	0.687	9.78	1.031	7.96	0.801	7.713	0.706
TCS	**0.279***	**0.078***	0.522	0.167	2.860	0.785	2.950	0.695	4.111	1.164	2.813	0.763	2.26	0.621
SBI	0.918	0.072	**0.838***	**0.071***	4.865	0.356	3.207	0.254	9.664	0.697	4.527	0.378	4.85	0.331
ONGC	0.736	0.133	**0.303***	**0.077***	1.53	0.249	0.86	0.157	1.34	0.313	1.23	0.193	2.136	0.321
Infosys	**0.443***	**0.078***	0.860	0.182	3.216	0.553	2.844	0.508	3.613	0.596	3.136	0.533	2.59	0.432
Sun Pharma	**0.320***	**0.035***	0.517	0.074	2.357	0.296	2.026	0.265	2.868	0.327	2.644	0.350	2.529	0.253
Spice Jet	0.838	0.034	**0.294***	**0.014***	3.120	0.166	2.005	0.124	4.105	0.212	2.774	0.137	4.444	0.179
Jet Airways	0.975	0.068	**0.404***	**0.047***	2.98	0.263	2.41	0.226	5.04	0.422	2.669	0.212	4.497	0.331

Table 8.8: Stock Prediction Results with Ch-10 Features from LIWC Features

Dataset	GMDH		GRNN		RF		QRRF		RPART		SVR		MLP	
	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE
Airtel	**1.03***	**0.209***	1.32	0.274	4.54	0.809	2.72	0.538	6.75	1.13	3.41	0.630	3.39	0.569
Mahindra	**4.31***	**0.207***	8.45	0.303	13.41	0.48	12.15	0.45	14.07	0.537	15.78	0.570	9.452	0.092
Tata Motors	**1.29***	**0.142***	1.61	0.143	6.95	0.588	5.832	0.487	6.642	0.652	6.894	0.603	6.703	0.555
Reliance	1.119	0.150	**1.031***	**0.153***	2.695	0.362	2.106	0.282	2.823	0.379	2.520	0.351	2.712	0.365
Tata Steel	1.892	0.199	**1.878***	**0.183***	12.188	0.981	9.623	0.916	16.419	1.405	10.691	0.901	9.969	0.835
TCS	**0.519***	**0.137***	0.732	0.215	2.69	0.731	1.906	0.604	2.96	0.758	2.503	0.707	1.961	0.517
SBI	1.82	0.134	**1.33***	**0.105***	5.901	0.427	4.345	0.283	8.234	0.612	6.005	0.474	5.327	0.353
ONGC	1.406	0.214	**0.351***	**0.080***	1.565	0.254	0.685	0.135	1.88	0.311	1.327	0.219	1.954	0.301
Infosys	**1.22***	**0.197***	1.312	0.248	3.69	0.648	2.92	0.533	3.57	0.617	4.06	0.761	3.507	0.584
Sun Pharma	**0.765***	**0.087***	1.32	0.164	2.222	0.287	1.636	0.222	2.640	0.292	2.416	0.294	2.113	0.225
Spice Jet	2.26	0.093	**1.10***	**0.052***	2.946	0.167	2.758	0.157	4.55	0.236	3.544	0.153	3.678	0.151
Jet Airways	2.09	0.169	**1.351***	**0.114***	3.09	0.263	2.78	0.279	6.057	0.488	3.63	0.288	4.194	0.310

Table 8.9: Stock Prediction Results with MRMR-10 Features from LIWC Features

Dataset	GMDH		GRNN		RF		QRRF		RPART		SVR		MLP	
	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE
Airtel	**1.86***	**0.347***	2.44	0.424	4.505	0.78	2.47	0.446	5.41	0.96	4.50	0.82	3.09	0.618
Mahindra	**2.77***	**0.107***	8.90	0.321	13.60	0.48	12.08	0.429	13.68	0.49	15.38	0.553	8.52	0.334
Tata Motors	1.26	0.125	**1.01***	**0.097***	6.338	0.544	4.95	0.441	4.78	0.482	6.91	0.629	4.586	0.429
Reliance	1.582	0.208	**1.063***	**0.152***	2.510	0.351	2.112	0.298	2.842	0.449	2.272	0.350	2.407	0.348
Tata Steel	**0.933***	**0.097***	1.120	0.1109	10.756	0.885	7.035	0.696	9.028	0.891	6.334	0.687	5.158	0.450
TCS	**0.408***	**0.099***	0.667	0.198	2.799	0.752	2.210	0.631	4.141	1.180	2.658	0.751	1.725	0.464
SBI	1.82	0.134	**1.33***	**0.105***	5.672	0.415	3.505	0.283	8.234	0.612	6.005	0.474	4.97	0.332
ONGC	1.603	0.242	**0.571***	**0.111***	1.44	0.261	0.968	0.179	2.39	0.424	1.35	0.222	2.151	0.328
Infosys	1.209	0.197	**1.047***	**0.203***	3.124	0.518	2.746	0.479	3.13	0.491	3.012	0.491	2.49	0.419
Sun Pharma	**0.603***	**0.065***	0.663	0.088	2.173	0.270	2.039	0.266	3.134	0.329	2.354	0.314	1.547	0.1507
Spice Jet	1.16	0.064	**0.716***	**0.040***	3.564	0.190	2.085	0.162	5.562	0.262	2.857	0.141	2.744	0.153
Jet Airways	1.718	0.131	**0.846***	**0.072***	3.03	0.260	3.11	0.239	6.05	0.488	3.632	0.288	4.603	0.325

Table 8.10: Features (LIWC) Selected through two Feature Selection Methods

Dataset	Feature Selection Method	Features
Tata Motors	Chi-10 (8)	Analytic, you, quant, female, cogproc, sexual, focuspast, Exclam
	MRMR-25	Focuspresent, quant, health, time, Comma, focuspast, WC, Period, Exclam, you, body, focusfuture, cause, conj, discrep, see, achieve,swear, ipron, male, leisure, posemo, QMark, Apostro, friend
Reliance	MRMR-25	Focusfuture, relig, OtherP, body, space, focuspast, health, drives, motion, friend, Colon, nonflu, Period, Comma, focuspresent, feel, QMark, bio, relativ, Parenth, affiliation, anx, cause, leisure, Dash
Tata Steel	Chi-25	Ipron, sexual, ppron, hear, percept, affiliation, QMark, death, pronoun, feel, shehe, they, we, article, negemo, SemiC, Analytic, male, Apostro, Quote, compare, i, achieve, affect, WPS
TCS	Chi-25	Shehe, relig, WC, OtherP, female, SemiC, Colon, death, differ, sexual, see, i, quant, Dic, AllPunc, male, Comma, achieve, netspeak, interrog, space, certain, QMark, family, adverb
Spice Jet	MRMR-10	Power, adverb, they, Sixltr, Analytic, discrep, AllPunc, relig, Period, SemiC
ONGC	Chi-25	Clout, ipron, pronoun, insight, achieve, feel, Tone, informal, work, health, bio, motion, OtherP, focusfuture, Quote, Apostro, certain, conj, differ, article, drives, prep, cogproc, family, social
Jet Airways	MRMR-25	Certain, Dash, Parenth, motion, we, Sixltr, verb, posemo, home, anger, Analytic, money, risk, Comma, Quote, number, see, article, male, ingest, ppron, relig, WC, family, netspeak
Airtel*	Chi-25 (16)	Swear, leisure, body, Apostro, focusfuture, percept, Quote, quant, affiliation, feel, space, i, assent, AllPunc, QMark, sexual
	MRMR-25	Focuspast, death, anx, body, assent, affiliation, insight, motion, ingest, auxverb, Apostro, swear, anger, bio, ipron, family, risk, posemo, differ, leisure, home, SemiC, QMark, work, adverb
Mahindra*	Chi-10	Anger, female, hear, sexual, function1, death, informal, nonflu, QMark, Exclam
	MRMR-25	Affiliation, leisure, see, death, they, Dic, nonflu, feel, anx, ingest, friend, discrep, Period, we, anger, Apostro, swear, sad, SemiC, hear, space, function1, relig, Parenth, informal
SBI*	Chi-25	Compare, reward, discrep, cause, verb, they, family, Tone, cogproc, Apostro, achieve, relig, sexual, negemo, Comma, insight, percept, auxverb, quant, i , Parenth, focuspresent, nonflu, Dash, friend
	MRMR-25	Reward, discrep, cause, family, Tone, cogproc, verb, compare, they, Apostro, achieve, relig, negemo, Comma, insight, percept, auxverb, i , quant, Parenth, focuspresent, nonflu, friend, Dash, sexual
Infosys*	Chi-25	Family, Parenth, Dic, adverb, health, female, AllPunc, sad, anger, Clout, conj, Analytic, adj, QMark, article, assent, verb,auxverb, risk, Tone, time, sexual, ppron, social, relig
	MRMR-25	Anger, reward, quant, Quote, health, differ, power, sad, nonflu, QMark, female, ingest, adj, death, focuspresent, body, relig, we, WPS, friend, negate, informal, money, focusfuture, you
Sun Pharma*	Chi-25	Anger, certain, money, achieve, prep, function, tentat, sexual, we, feel, pronoun, OtherP, you, leisure, adj, ipron, focuspast, i, focusfuture, family, cause, article, see, assent, affect
	MRMR-25	Feel, Authentic, affect, article, adverb, we, prep, body, leisure, relig, percept, pronoun, certain, focuspast, function, adj, ppron, anger, Dash, Period, shehe, hear, WC, Clout, cause

*=Statistically significant

Table 8.11: DM Test Values of the Models with LIWC Features

Dataset	**GMDH (Full Features) vs. GMDH (a/ b/ c/ d)**			
	Ch_25^a	**Ch**_10^b	**MRMR**_25^c	**MRMR**_10^d
Airtel	-2.99	-2.02	-2.25	-2.13
Mahindra	NA	-2.26	-3.17	-2.95
Tata Motors	NA	-1.86	-0.94	-2.36
Reliance Industries	NA	-3.41	-1.80	-3.80
Tata Steel	-1.74	-2.76	-2.48	-2.54
TCS	-1.73	-3.54	-2.72	-3.55
SBI	-2.76	-2.98	-2.76	-2.98
Infosys	-3.21	-4.25	-2.92	-3.63
Sun Pharma	-2.04	-2.61	-2.02	-2.87
Spice Jet	-4.58	-3.57	-3.09	-1.92
	GRNN (Full Features) vs. GRNN (a/ b/ c/ d)			
ONGC	-0.99	-2.43	-4.47	-2.25
Jet Airways	-2.19	-2.97	-1.08	-2.64

NA=Not Applicable, a = Ch_25, b = Ch_10, c = MRMR_25, d = MRMR_10

Table 8.12: Stock Prediction Results with TAALES Full Features

Dataset	GMDH		GRNN		RF		QRRF		RPART		SVR		MLP	
	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE
Airtel	**0.259***	**0.046***	1.169	0.243	6.76	1.067	2.607	0.48	3.394	0.586	6.288	1.001	4.99	0.752
Mahindra	**1.782***	**0.073***	10.34	0.349	16.91	0.592	16.431	0.578	17.57	0.618	17.19	0.605	15.85	0.579
Tata Motors	**1.372***	**0.141***	7.17	0.613	10.25	0.827	5.41	0.495	10.05	0.829	8.97	0.748	9.092	0.729
Reliance	**0.446***	**0.068***	1.905	0.254	2.95	0.365	2.818	0.337	3.36	0.411	2.79	0.349	2.157	0.318
Tata Steel	**3.007***	**0.308***	5.513	0.581	16.23	1.31	11.04	1.022	15.60	1.45	16.47	1.33	17.21	1.346
TCS	**0.619***	**0.190***	3.058	0.882	5.397	1.358	4.312	1.085	6.53	1.818	5.55	1.395	5.345	1.370
SBI	**2.113***	**0.156***	1.246	0.088	6.207	0.397	3.651	0.252	12.38	0.760	5.66	0.378	4.98	0.334
ONGC	**0.711***	**0.113***	1.055	0.1506	2.46	0.352	1.788	0.257	4.193	0.624	2.138	0.316	1.935	0.294
Infosys	**0.410***	**0.076***	4.934	0.807	4.49	0.750	3.010	0.518	5.146	0.788	4.418	0.720	4.224	0.725
Sun Pharma	**0.867***	**0.084***	1.960	0.208	5.49	0.494	4.46	0.403	5.43	0.497	5.198	0.472	10.95	0.998
Spice Jet	**2.090***	**0.098***	8.023	0.303	11.623	0.40	5.135	0.244	9.756	0.395	8.252	0.317	72.33	2.37
Jet Airways	2.014	0.178	**1.842***	**0.153***	3.95	0.311	3.79	0.275	5.714	0.464	4.295	0.328	4.507	0.339

Table 8.12 presents the MAPE and NRMSE values yielded on TAALES features by various prediction models without feature selection. It can be observed that except on one dataset(JetAirways), GMDH outperformed the other techniques on all the datasets in terms of MAPE and NRMSE. For the JetAirways dataset, GRNN performed better than GMDH. Table 8.13 presents the results obtained by employing various models on the top-25 features obtained by Chi-square feature selection method. From Table 8.13, we can observe that except for the SBI dataset, GMDH yielded the best predictions for all datasets. For this dataset, GRNN outperformed all other techniques. The results obtained with top-25 features selected by MRMR method are presented in Table 8.14. In this combination, GMDH performed the best in terms of MAPE and NRMSE on all datasets except SBI, and ONGC datasets, for which, GRNN performed the best. Table 8.15 summarizes the results yielded by the models on the top-10 features selected by Chi-square method. In this table, it can be observed that the GMDH outperformed all other techniques on the datasets of Airtel, Mahindra, Reliance Industries, TCS, SBI, Infosys, Sun Pharma, Spice Jet. Whereas, GRNN yielded the best predictions on Tata Motors, Tata Steel, ONGC and Jet Airways in terms of both MAPE and NRMSE values. Table 8.16 presents the results obtained with top-10 features selected by MRMR method. In this table, it can be observed that GRNN outperformed other models in terms of both MAPE and NRMSE on four companies' stocks (Reliance Industries, SBI, ONGC, and Sun Pharma) and GMDH performed the best on the remaining eight datasets.

Using the features extracted using TAALES as the feature subset, the DM test was conducted between case (i) (all features) of GMDH and all other cases of GMDH in a pairwise manner for all datasets except Jet Airways. For this dataset, we performed DM test between case (i) of GRNN and all other cases of GRNN in a pair wise manner. We chose GMDH and GRNN because of their superior performance over other models in terms of MAPE and NRMSE as seen in Tables 8.12 through 8.16. The DM test values are reported in Table 8.17. As seen in Table 8.17, the absolute value of the DM statistic [298] is less than 1.96 in the following cases. Tata Steel (case (iv)), TCS (case (ii) and (iii)), SBI (case (ii) and (iii)), ONGC (case (iv)), Infosys (case(ii)), Spice Jet (case (ii) and (iv)), and Jet Airways (case (ii) and (iii)). It indicates that there is no statistically

Table 8.13: Stock Prediction Results with Ch-25 Features from TAALES Features

Dataset	GMDH		GRNN		RF		QRRF		RPART		SVR		MLP	
	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE
Airtel	**1.163***	**0.200***	2.234	0.412	7.248	1.165	3.004	0.46	7.88	1.239	6.71	1.093	4.64	0.713
Mahindra	**3.53***	**0.151***	13.37	0.487	16.99	0.594	16.183	0.547	17.63	0.609	17.67	0.614	13.85	0.509
Tata Motors	**3.74***	**0.367***	4.45	0.437	9.102	0.773	6.109	0.550	7.99	0.777	8.15	0.744	7.757	0.656
Reliance	**1.124***	**0.145***	1.746	0.221	3.359	0.442	2.661	0.330	4.122	0.535	3.386	0.415	2.82	0.33
Tata Steel	**4.665***	**0.472***	4.81	0.516	14.63	1.208	12.07	1.151	18.47	1.563	18.31	1.148	15.344	1.341
TCS	**0.947***	**0.310***	3.369	0.905	5.59	1.42	3.661	0.983	5.34	1.52	5.322	1.364	5.312	1.314
SBI	3.186	0.206	**1.478***	**0.113***	6.763	0.451	3.724	0.248	11.02	0.684	5.69	0.393	4.706	0.307
ONGC	**1.251***	**0.182***	1.798	0.279	3.04	0.429	1.508	0.24	2.421	0.384	2.21	0.356	2.272	0.335
Infosys	**0.697***	**0.123***	4.696	0.727	4.582	0.749	3.676	0.613	4.599	0.734	4.308	0.678	3.872	0.658
Sun Pharma	**1.464***	**0.1503***	2.060	0.216	5.716	0.5008	4.608	0.433	7.35	0.619	5.060	0.457	3.46	0.355
Spice Jet	**3.017***	**0.174***	7.572	0.284	14.109	0.469	5.809	0.265	7.697	0.346	8.245	0.317	4.341	0.206
Jet Airways	**2.0307***	**0.1591***	2.852	0.220	5.184	0.380	3.676	0.613	5.980	0.477	5.46	0.379	4.532	0.332

Table 8.14: Prediction Results with MRMR-25 Features from TAALES Features

Dataset	GMDH		GRNN		RF		QRRF		RPART		SVR		MLP	
	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE
Airtel	**0.884***	**0.148***	1.743	0.315	7.226	1.104	2.985	0.482	6.172	0.916	5.96	0.913	4.41	0.678
Mahindra	**5.037***	**0.202***	8.98	0.304	17.07	0.596	16.18	0.569	18.14	0.634	16.52	0.580	16.51	0.573
Tata Motors	**2.93***	**0.278***	3.626	0.298	11.341	0.89	7.084	0.600	11.11	0.966	9.538	0.789	5.835	0.524
Reliance	**1.293***	**0.168***	1.768	0.218	2.79	0.347	2.506	0.307	3.361	0.474	2.88	0.351	3.305	0.392
Tata Steel	**3.821***	**0.366***	4.78	0.430	16.31	1.29	16.509	1.372	17.43	1.469	16.55	1.32	14.64	1.163
TCS	**0.975***	**0.293***	1.875	0.617	5.155	1.295	3.99	1.055	4.107	1.076	4.839	1.232	5.562	1.372
SBI	3.398	0.231	**1.627***	**0.122***	6.139	0.404	3.609	0.242	8.102	0.572	6.081	0.393	5.102	0.335
ONGC	1.037	0.162	**0.863***	**0.151***	3.03	0.446	1.526	0.234	4.809	0.656	2.485	0.403	2.257	0.332
Infosys	**0.845***	**0.143***	4.858	0.749	4.901	0.791	3.884	0.643	5.373	0.837	4.013	0.654	4.372	0.681
Sun Pharma	**1.412***	**0.157***	2.036	0.213	4.427	0.429	4.006	0.396	4.92	0.470	4.898	0.450	5.064	0.475
Spice Jet	**2.256***	**0.089***	8.502	0.309	14.43	0.475	5.28	0.236	7.41	0.303	7.22	0.278	14.39	0.475
Jet Airways	**1.860***	**0.135***	2.654	0.199	6.326	0.459	3.884	0.643	9.501	0.712	5.315	0.401	4.481	0.334

Table 8.15: Stock Prediction Results with Ch-10 Features from TAALES Features

Dataset	GMDH		GRNN		RF		QRRF		RPART		SVR		MLP	
	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE
Airtel	**1.201***	**0.230***	2.93	0.478	7.509	1.205	5.127	0.825	8.126	1.263	7.763	1.262	4.68	0.722
Mahindra	**5.513***	**0.214***	13.03	0.474	17.26	0.603	15.98	0.558	15.03	0.532	17.91	0.629	15.35	0.547
Tata Motors	4.837	0.438	**4.209***	**0.391***	8.92	0.757	5.95	0.511	9.54	0.771	8.117	0.718	8.068	0.685
Reliance	**1.881***	**0.241***	2.152	0.260	3.407	0.429	2.834	0.352	3.76	0.472	3.134	0.395	3.427	0.414
Tata Steel	7.234	0.639	**5.666***	**0.679***	15.97	1.303	14.326	1.215	13.67	1.22	20.09	1.59	13.58	1.275
TCS	**1.094***	**0.375***	3.156	0.824	5.378	1.413	4.525	1.167	6.04	1.759	5.77	1.504	5.115	1.284
SBI	**2.591***	**0.197***	6.136	1.825	6.237	0.426	3.862	0.256	10.97	0.641	5.53	0.366	5.22	0.346
ONGC	2.115	0.306	**1.160***	**0.200***	3.37	0.477	1.61	0.246	3.93	0.545	2.79	0.436	2.278	0.334
Infosys	**1.208***	**0.211***	4.587	0.712	4.751	0.772	4.283	0.700	6.245	1.084	4.682	0.726	4.413	0.681
Sun Pharma	**1.464***	**0.150***	2.93	0.303	5.69	0.505	5.371	0.480	7.30	0.645	5.26	0.477	5.053	0.477
Spice Jet	**3.101***	**0.161***	6.381	0.262	13.23	0.444	7.973	0.295	7.582	0.342	8.74	0.314	4.635	0.230
Jet Airways	2.605	0.204	**2.224***	**0.169***	4.92	0.349	4.283	0.700	7.281	0.564	5.73	0.405	4.541	0.334

Table 8.16: Prediction Results with MRMR-10 Features from TAALES Features

Dataset	GMDH		GRNN		RF		QRRF		RPART		SVR		MLP	
	MAPE	NRMSE	MAPE		MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE	MAPE	NRMSE
Airtel	**2.51***	**0.487***	3.06	0.560	7.013	1.098	2.818	0.569	8.045	1.255	7.018	1.125	4.344	0.667
Mahindra	**7.32***	**0.281***	10.23	0.351	18.15	0.629	16.21	0.572	18.88	0.664	17.33	0.609	16.75	0.585
Tata Motors	**4.462***	**0.391***	5.24	0.444	9.691	0.795	6.668	0.571	11.38	1.01	9.57	0.807	7.32	0.647
Reliance	1.857	0.246	**1.788***	**0.235***	3.091	0.375	2.557	0.305	3.99	0.520	2.97	0.352	2.59	0.323
Tata Steel	**6.448***	**0.592***	6.818	0.655	15.92	1.288	12.62	1.123	15.83	1.33	17.06	1.37	11.26	0.976
TCS	**1.682***	**0.541***	3.107	0.945	5.483	1.39	3.97	1.004	7.062	1.821	5.591	1.41	5.461	1.374
SBI	4.758	0.318	**3.053***	**0.215***	6.91	0.474	3.686	0.253	9.77	0.667	6.61	0.436	5.563	0.365
ONGC	1.921	0.326	**1.352***	**0.205***	2.62	0.391	1.706	0.262	3.93	0.56	2.45	0.370	2.269	0.332
Infosys	**1.0301***	**0.180***	4.868	0.748	5.184	0.802	3.911	0.617	7.014	1.098	4.648	0.742	4.115	0.675
Sun Pharma	2.99	0.312	**2.72***	**0.267***	4.781	0.453	4.132	0.395	5.24	0.495	5.431	0.504	5.60	0.531
Spice Jet	**2.734***	**0.148***	8.509	0.309	11.978	0.404	5.16	0.241	6.784	0.295	6.91	0.270	12.74	0.431
Jet Airways	**2.146***	**0.203***	3.776	0.280	6.85	0.493	3.911	0.617	8.92	0.628	5.67	0.429	4.352	0.322

Table 8.17: DM Test Values of the Models with TAALES Features

Dataset	GMDH (Full Features) vs. GMDH (a/ b/ c/ d)			
	Ch_25^a	**Ch_10**^{b}	**MRMR_25**^{c}	**MRMR_10**^{d}
Airtel	-3.5448	-1.9618	-2.9054	-2.2615
Mahindra	-2.3672	-2.9913	-2.8336	-3.583
Tata Motors	-2.3954	-2.6174	-3.0627	-3.662
Reliance Industries	-3.529	-3.8573	-4.222	-3.9516
Tata Steel	-2.0483	-4.6712	-1.0015	-3.5876
TCS	-1.8943	-1.8918	-2.22	-2.4727
SBI	-1.6882	-0.9737	-2.1516	-3.2354
ONGC	-2.4633	-3.9292	-1.6057	-2.8267
Infosys	-1.6201	-3.1324	-2.6704	-3.0838
Sun Pharma	-3.2006	-3.0988	-2.0877	-3.5302
Spice Jet	-1.9323	-2.2338	0.40826	-2.1279
	GRNN (Full Features) vs. GRNN (a/ b/ c/ d)			
Jet Airways	-1.5835	-0.6087	-2.188	-3.2475

significant difference between GMDH (case (i)) and GMDH (cases mentioned above) or GRNN (case (i)) and the GRNN (cases referred to above) as the case may beat 5% level of significance. Therefore, in these datasets, the corresponding cases of feature subset selection methods turned out to better than the case (i) in terms of MAPE and NRMSE. However, in the rest of the cases in Table 8.17, the absolute of DM statistic is greater than 1.96 which indicates that case (i) of full features is statistically significantly better than all feature subset selection cases in terms of MAPE and NRMSE at 5% level of significance.

Similarly, we also conducted the DM test between the LIWC and TAALES models i.e.,. between case (i) (all features) of GMDH (LIWC) and case (i) (all featuress) of GMDH (TAALES) in a pairwise manner for all datasets. Similarly, case (i) of GRNN (LIWC) vs. case(i) of GRNN (TAALES). We chose GMDH and GRNN because of their superior performance over other models in terms of MAPE and NRMSE as seen in Tables 8.5 and 8.12. The DM Test values are reported in Table 8.18.

Table 8.18: DM Test Values of LIWC vs. TAALES Feature Models

Dataset	Full Features	
	GMDH (LIWC) vs. GMDH (TAALES)	**GRNN (LIWC) vs. GRNN (TAALES)**
Airtel	-2.3504	0.74582
Mahindra	-3.1204	0.54479
Tata Motors	-2.1748	-5.2688
Reliance Industries	-1.4084	-3.183
Tata Steel	-3.385	-2.5014
TCS	-2.9328	-3.591
SBI	-2.7933	-2.5652
ONGC	-1.6129	-2.4851
Infosys	-2.4362	-4.1836
Sun Pharma	-4.105	-2.863
Spice Jet	-2.2874	-5.3706
Jet Airways	-2.0484	-3.3333

As seen in Table 8.18, the absolute value of the DM statistic is less than 1.96 in the following cases: Airtel (GRNN), Mahindra (GRNN), Reliance Industries (GMDH), ONGC (GMDH). It indicates that there is no statistically significant difference between GMDH (LIWC) and GMDH (TAALES) with case (i) or GRNN (LIWC) and GRNN (TAALES) with case (i) at 5% level of significance. Therefore, for these datasets, the corresponding models of both, TAALES and LIWC turned out to be equally good in case (i), in terms of MAPE and NRMSE. However, in the rest of the cases in Table 8.18, the absolute of DM statistic is greater than 1.96 which indicates that case (i) of full features with GMDH or GRNN is statistically significantly better than all feature subset selection cases in terms of MAPE and NRMSE at 5% level of significance.

Thus, the two different feature selection methods adopted here did not perform uniformly well on all datasets because they are filter based approaches and are not as powerful as the wrapper based ones. In this context, one can employ the new elitist quantum inspired differential evolution based wrapper developed by

Srikrishna et al. [299] to see if any significant improvement in prediction accuracy can be obtained. The reason for this suggestion is that, it not only depends on the impressive search capabilities of Differential Evolution, but also the powerful quantum computing principles.

8.7 Conclusions and Future Directions

In this chapter, a novel stock market prediction model based on the psycholinguistic features extracted from selected stock (company) related news articles, is proposed. Various prediction models viz., RF, QRRF, GMDH, SVR, CART, MLP and GRNN were employed for regression. Experiments were conducted on stock prices of 12 companies listed on BSE. Due to non-availability of news articles of some days, for a particular stock, mean-distance based data imputation was employed. In our experiments, it was found that statistically, GMDH yielded the best performance followed by GRNN in terms of MAPE and NRMSE using the DM test. LIWC features models performed better as compared to TAALES features models. Going further, technical indicators can also be included as predictor variables along with the psycholinguistic and lexical features to get higher accuracies. It is important to note that in the current research, we employed filter-based feature subset selection methods. However, wrapper-based feature subset selection methods, which are designed to take inter-variable interaction effects into consideration may prove to be more potent and are worth exploring. Further, ensembling the predictions yielded by some well performing intelligent techniques is also a future research direction. Finally, psycholinguistic features coupled with evolutionary computation based stock prediction models [300] is another direction worth exploring.

Chapter 9

Conclusions and Future Directions

9.1 Conclusions

This thesis proposes a set of novel models for text mining applications. The procedures include:

1. Binary document classification using data mining techniques. Four feature subset selection methods, viz., Chi-square, Gini index, Correlation, t-statistic are invoked separately, followed by classification with SVM/ DT/ K-NN/ NB/ GMDH/ GRNN/ MLP/ RIPPER algorithms. The study revealed that GMDH performs the best, followed by PNN, RIPPER.

2. One-class document classification using One-Class Support Vector Machine (OCSVM) and Latent Semantic Indexing (LSI) in tandem. Support Vectors (SV) are extracted using OCSVM from one-class samples (Negative class) followed by LSI for classification based on the similarity value.

3. A hybrid model for document classification using Principal Component Analysis (PCA) and One-Class Support Vector Machine.

4. A hybrid model for document classification using topic modeling and Class Association Rule Mining (ARM). Feature subsets are extracted by topic

modeling followed by classification with association rules. Rules were generated by Apriori and FP-Growth Algorithms.

5. Various hybrid intelligent models are proposed for stock market prediction using psycholinguistic and lexical sophistication features of the news.

9.2 Future Directions

We conclude the thesis with the following key future directions:

1. Churn prediction problem using text mining is not yet fully explored.
2. Hybridization of methods, ensembling of methods needs to be explored.
3. Construction of models using Deep Learning needs to be investigated.
4. Stock market predictions with various time zones needs to be examined.
5. Ensembling the predictions yielded by some well performing intelligent techniques is also a future research direction.
6. The performance of spiking neural networksin text mining is yet to be studied.
7. Benchmark datasets need to be created to evaluate the performance of new research endeavors in this area.
8. Sentence subjectivity classification and semantic structure identification are the most challenging problems. Developments in these segments will advance the text mining field application to the sentiment analysis.
9. Evolutionary algorithms need to be explored with respect to tasks of text mining.
10. Deep Learning, and Spiking neural networks havenot been applied in text mining in proportion to their spread in other areas. There is a scope to mine this segment.

11. Developments in big data analytics can be exploited successfully in text mining applications to finance too.

12. Exploring LDA together with evolutionary algorithms in single and multi objective framework to generate robust and less number of class association rules for classification can be worthwhile.

References

[1] F. Sebastiani. **Text Mining and its Applications to Intelligence**, 2005.

[2] A-H. Tan. **Text mining: The state of the art and the challenges**. In *Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases*, **8**, pages 65–70, 1999.

[3] M. Alazab, S. Venkataraman, and P. Watters. **Towards understanding malware behaviour by the extraction of API calls**. In *Cybercrime and Trustworthy Computing Workshop (CTC), 2010 Second*, pages 52–59. IEEE, 2010.

[4] J. Dörre, P. Gerstl, and R. Seiffert. **Text mining: finding nuggets in mountains of textual data**. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 398–401, 1999.

[5] R. Feldman and J. Sanger. *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge university press, 2007.

[6] F. Sebastiani. **Machine learning in automated text categorization**. *ACM computing surveys (CSUR)*, **34**(1):1–47, 2002.

[7] P. M. Chinta and M. N. Murty. **Discriminative feature analysis and selection for document classification**. In *International Conference on Neural Information Processing*, pages 366–374. Springer, 2012.

[8] M. E. Maron. **Automatic indexing: an experimental inquiry**. *Journal of the ACM (JACM)*, **8**(3):404–417, 1961.

[9] H. Borko and M. Bernick. **Automatic document classification**. *Journal of the ACM (JACM)*, **10**(2):151–162, 1963.

[10] J. R. Quinlan. **Simplifying decision trees**. *International journal of man-machine studies*, **27**(3):221–234, 1987.

[11] C. Cortes and V. Vapnik. **Support-vector networks**. *Machine learning*, **20**(3):273–297, 1995.

[12] P. Domingos and M. Pazzani. **On the optimality of the simple Bayesian classifier under zero-one loss**. *Machine learning*, **29**(2):103–130, 1997.

[13] T. Cover and P. Hart. **Nearest neighbor pattern classification**. *IEEE transactions on information theory*, **13**(1):21–27, 1967.

[14] C. Goodhart. **News and the foreign exchange market**. Technical report, Financial Markets Group, 1990.

[15] G. P. C. Fung, J. X. Yu, and W. Lam. **News sensitive stock trend prediction**. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 481–493. Springer, 2002.

[16] M. D. D. Evans and R. K. Lyons. **How is macro news transmitted to exchange rates?** *Journal of Financial Economics*, **88**(1):26–50, 2008.

[17] T-T. Vu, S. Chang, Q. T. Ha, and N. Collier. **An Experiment in Integrating Sentiment Features for Tech Stock Prediction in Twitter**. In *24th International Conference on Computational Linguistics*, page 23, 2012.

[18] F. Jin, N. Self, P. Saraf, P. Butler, W. Wang, and N. Ramakrishnan. **Forex-foreteller: Currency trend modeling using news articles**. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1470–1473, 2013.

[19] Yang Yu, Wenjing Duan, and Qing Cao. **The impact of social and conventional media on firm equity value: A sentiment analysis approach**. *Decision Support Systems*, **55**(4):919–926, 2013.

[20] A. Chatrath, H. Miao, S. Ramchander, and S. Villupuram. **Currency jumps, cojumps and the role of macro news**. *Journal of International Money and Finance*, **40**:42–62, 2014.

[21] E. F. Fama. **The behavior of stock-market prices**. *The journal of Business*, **38**(1):34–105, 1965.

[22] B. Back, J. Toivonen, H. Vanharanta, and A. Visa. **Comparing numerical data and text information from annual reports using self-organizing maps**. *International Journal of Accounting Information Systems*, **2**(4):249–269, 2001.

[23] G. P. C. Fung, J. X. Yu, and W. Lam. **Stock prediction: Integrating text mining approach using real-time news**. In *Computational Intelligence for Financial Engineering, 2003. Proceedings. 2003 IEEE International Conference on*, pages 395–402. IEEE, 2003.

[24] A. Kloptchenko, T. Eklund, J. Karlsson, B. Back, and A. Vanharanta, H.and Visa. **Combining data and text mining techniques for analysing financial reports**. *Intelligent systems in accounting, finance and management*, **12**(1):29–41, 2004.

[25] M. Koppel and I. Shtrimberg. **Good news or bad news? let the market decide**. *Computing attitude and affect in text: Theory and applications*, pages 297–301, 2006.

[26] S. Wang, Z. Zhe, Y. Kang, H. Wang, and X. Chen. **An ontology for causal relationships between news and financial instruments**. *Expert Systems with Applications*, **35**(3):569–580, 2008.

[27] L. Dey, A. Mahajan, and S. M. Haque. **Document clustering for event identification and trend analysis in market news**. In *Advances*

in Pattern Recognition, 2009. ICAPR'09. Seventh International Conference on, pages 103–106. IEEE, 2009.

[28] M. Fasanghari and G. A. Montazer. **Design and implementation of fuzzy expert system for Tehran Stock Exchange portfolio recommendation**. *Expert Systems with Applications*, **37**(9):6138–6147, 2010.

[29] S. Mellouli, F. Bouslama, and A. Akande. **An ontology for representing financial headline news**. *Web Semantics: Science, Services and Agents on the World Wide Web*, **8**(2):203–208, 2010.

[30] E. Gilbert and K. Karahalios. **Widespread Worry and the Stock Market.** In *ICWSM*, pages 59–65, 2010.

[31] S. Wang, K. Xu, L. Liu, B. Fang, S. Liao, and H. Wang. **An ontology based framework for mining dependence relationships between news and financial instruments**. *Expert Systems with Applications*, **38**(10):12044–12050, 2011.

[32] S. W. K. Chan and J. Franklin. **A text-based decision support system for financial sequence prediction**. *Decision Support Systems*, **52**(1):189–198, 2011.

[33] P. S. M. Nizer and J. C. Nievola. **Predicting published news effect in the Brazilian stock market**. *Expert Systems with Applications*, **39**(12):10674–10680, 2012.

[34] A. Moniz and F. De Jong. **Classifying the influence of negative affect expressed by the financial media on investor behavior**. In *Proceedings of the 5th Information Interaction in Context Symposium*, pages 275–278. ACM, 2014.

[35] 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*, **41**(16):7653–7670, 2014.

[36] A. K. Nassirtoussi, S. Aghabozorgi, T. Y. Wah, and D. C. L. Ngo. **Text mining of news-headlines for FOREX market prediction: A Multi-layer Dimension Reduction Algorithm with semantics and sentiment**. *Expert Systems with Applications*, **42**(1):306–324, 2015.

[37] S. Deng, A. P. Sinha, and H. Zhao. **Adapting sentiment lexicons to domain-specific social media texts**. *Decision Support Systems*, **94**:65–76, 2017.

[38] R. H. Gálvez and A. Gravano. **Assessing the usefulness of online message board mining in automatic stock prediction systems**. *Journal of Computational Science*, **19**:43–56, 2017.

[39] N. Oliveira, P. Cortez, and N. Areal. **The impact of microblogging data for stock market prediction: Using Twitter to predict returns, volatility, trading volume and survey sentiment indices**. *Expert Systems with Applications*, **73**:125–144, 2017.

[40] B. Weng, M. A. Ahmed, and F. M. Megahed. **Stock market one-day ahead movement prediction using disparate data sources**. *Expert Systems with Applications*, **79**:153–163, 2017.

[41] B. Li, K. C. C. Chan, C. Ou, and S. Ruifeng. **Discovering public sentiment in social media for predicting stock movement of publicly listed companies**. *Information Systems*, **69**:81–92, 2017.

[42] Q. Song, A. Liu, and S. Y. Yang. **Stock portfolio selection using learning-to-rank algorithms with news sentiment**. *Neurocomputing*, 2017.

[43] M-A. Mittermayer. **Forecasting intraday stock price trends with text mining techniques**. In *system sciences, 2004. proceedings of the 37th annual hawaii international conference on*, pages 10–pp. IEEE, 2004.

[44] W. Antweiler and M. Z. Frank. **Is all that talk just noise? The information content of internet stock message boards**. *The Journal of Finance*, **59**(3):1259–1294, 2004.

[45] S. R. Das and M. Y. Chen. **Yahoo! for Amazon: Sentiment extraction from small talk on the web**. *Management science*, **53**(9):1375–1388, 2007.

[46] A. Mahajan, L. Dey, and S. M. Haque. **Mining financial news for major events and their impacts on the market**. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 423–426, 2008.

[47] P. C. Tetlock, M. Saar-Tsechansky, and S. Macskassy. **More than words: Quantifying language to measure firms' fundamentals**. *The Journal of Finance*, **63**(3):1437–1467, 2008.

[48] R. P. Schumaker and H. Chen. **Textual analysis of stock market prediction using breaking financial news: The AZFin text system**. *ACM Transactions on Information Systems (TOIS)*, **27**(2):12, 2009.

[49] M. Butler and V. Keselj. **Financial Forecasting Using Character N-Gram Analysis and Readability Scores of Annual Reports.** In *Canadian Conference on AI*, pages 39–51, 2009.

[50] J. Bollen, H. Mao, and X. Zeng. **Twitter mood predicts the stock market**. *Journal of computational science*, **2**(1):1–8, 2011.

[51] C-J. Huang, J-J. Liao, D-X. Yang, T-Y. Chang, and Y-C. Luo. **Realization of a news dissemination agent based on weighted association rules and text mining techniques**. *Expert Systems with Applications*, **37**(9):6409–6413, 2010.

[52] F. Li. **The information content of forward-looking statements in corporate filings—A naïve Bayesian machine learning approach**. *Journal of Accounting Research*, **48**(5):1049–1102, 2010.

[53] S. S. Groth and J. Muntermann. **An intraday market risk management approach based on textual analysis**. *Decision Support Systems*, **50**(4):680–691, 2011.

[54] R. P. Schumaker, Y. Zhang, C-N. Huang, and H. Chen. **Evaluating sentiment in financial news articles**. *Decision Support Systems*, **53**(3):458–464, 2012.

[55] M. Hagenau, M. Liebmann, and D. Neumann. **Automated news reading: Stock price prediction based on financial news using context-capturing features**. *Decision Support Systems*, **55**(3):685–697, 2013.

[56] J. Riedl, J. A. Konstan, and J. B. Schafer. **Electronic commerce recommender applications**. *Journal of Data Mining and Knowledge Discovery*, **5**(1/2):115–152, 2000.

[57] B. Pang, L. Lee, and S. Vaithyanathan. **Thumbs up?: sentiment classification using machine learning techniques**. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[58] M. Hu and B. Liu. **Mining and summarizing customer reviews**. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[59] B. Liu, M. Hu, and J. Cheng. **Opinion observer: analyzing and comparing opinions on the web**. In *Proceedings of the 14th international conference on World Wide Web*, pages 342–351, 2005.

[60] A-M. Popescu and O. Etzioni. **Extracting product features and opinions from reviews**. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 339–346. Association for Computational Linguistics, 2005.

[61] R. Ghani, K. Probst, Y. Liu, M. Krema, and A. Fano. **Text mining for product attribute extraction**. *ACM SIGKDD Explorations Newsletter*, **8**(1):41–48, 2006.

[62] A. Devitt and K. Ahmad. **Sentiment polarity identification in financial news: A cohesion-based approach**. In *ACL*, **7**, pages 1–8, 2007.

[63] K. Coussement and D. Van den Poel. **Improving customer complaint management by automatic email classification using linguistic style features as predictors**. *Decision Support Systems*, **44**(4):870–882, 2008.

[64] B. Pang and L. Lee. **Opinion mining and sentiment analysis**. *Foundations and Trends® in Information Retrieval*, **2**(1–2):1–135, 2008.

[65] H. Sayyadi, A. Sahraei, and H. Abolhassani. **Event detection from news articles**. In *Advances in Computer Science and Engineering*, pages 981–984. 2008.

[66] A. Bifet and E. Frank. **Sentiment knowledge discovery in twitter streaming data**. In *International Conference on Discovery Science*, pages 1–15. Springer, 2010.

[67] L. Dey, S. M. Haque, and N. Raj. **Mining customer feedbacks for actionable intelligence**. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, **3**, pages 239–242. IEEE, 2010.

[68] D. Thorleuchter and D. Van Den Poel. **Predicting e-commerce company success by mining the text of its publicly-accessible website**. *Expert Systems with Applications*, **39**(17):13026–13034, 2012.

[69] B. Liu and L. Zhang. **A survey of opinion mining and sentiment analysis**. In *Mining text data*, pages 415–463. Springer, 2012.

[70] M. Ghiassi, J. Skinner, and D. Zimbra. **Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network**. *Expert Systems with applications*, **40**(16):6266–6282, 2013.

[71] K. Ikeda, G. Hattori, C. Ono, H. Asoh, and T. Higashino. **Twitter user profiling based on text and community mining for market analysis**. *Knowledge-Based Systems*, **51**:35–47, 2013.

[72] W. He, S. Zha, and L. Li. **Social media competitive analysis and text mining: A case study in the pizza industry**. *International Journal of Information Management*, **33**(3):464–472, 2013.

[73] I. Verma, L. Dey, R. S. Srinivasan, and L. Singh. **Event Detection from Business News**. In *International Conference on Pattern Recognition and Machine Intelligence*, pages 575–585, 2015.

[74] M. Ballings and D. Van den Poel. **CRM in social media: Predicting increases in Facebook usage frequency**. *European Journal of Operational Research*, **244**(1):248–260, 2015.

[75] Kumar Ravi and Vadlamani Ravi. **A survey on opinion mining and sentiment analysis: tasks, approaches and applications**. *Knowledge-Based Systems*, **89**:14–46, 2015.

[76] **Net Losses: Estimating the Global cost of Cybercrime**. `www.mcafee.com/in/resources/reports/rp-economic-impact-cybercrime2.pdf`, 2004.

[77] **Symantec Internet Security Threat Report (ISTA)**. `https://www.symantec.com/security-center/threat-report`, 2016.

[78] Y. Pan and X. Ding. **Anomaly based web phishing page detection**. In *Computer Security Applications Conference, 2006. ACSAC'06. 22nd Annual*, pages 381–392. IEEE, 2006.

[79] R. Dhamija, J. D. Tygar, and M. Hearst. **Why phishing works**. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 581–590. ACM, 2006.

[80] M. Chandrasekaran, K. Narayanan, and S. Upadhyaya. **Phishing email detection based on structural properties**. In *NYS Cyber Security Conference*, **3**, 2006.

[81] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair. **A comparison of machine learning techniques for phishing detection**. In *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*, pages 60–69, 2007.

[82] 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*, pages 639–648, 2007.

[83] C. Ludl, S. McAllister, E. Kirda, and C. Kruegel. **On the effectiveness of techniques to detect phishing sites**. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 20–39, 2007.

[84] S. Garera, N. Provos, M. Chew, and A. D. Rubin. **A framework for detection and measurement of phishing attacks**. In *Proceedings of the 2007 ACM workshop on Recurring malcode*, pages 1–8, 2007.

[85] I. Fette, N. Sadeh, and A. Tomasic. **Learning to detect phishing emails**. In *Proceedings of the 16th international conference on World Wide Web*, pages 649–656, 2007.

[86] D. Miyamoto, H. Hazeyama, and Y. Kadobayashi. **An evaluation of machine learning-based methods for detection of phishing sites**. In *International Conference on Neural Information Processing*, pages 539–546. Springer, 2008.

[87] R. Basnet, S. Mukkamala, and A. H. Sung. **Detection of phishing attacks: A machine learning approach**. In *Soft Computing Applications in Industry*, pages 373–383. Springer, 2008.

[88] G. Xiang and J. I. Hong. **A hybrid phish detection approach by identity discovery and keywords retrieval**. In *Proceedings of the 18th international conference on World wide web*, pages 571–580, 2009.

[89] S. Sheng, B. Wardman, G. Warner, L. F. Cranor, J. Hong, and C. Zhang. **An empirical analysis of phishing blacklists**. In *Proceedings of Sixth Conference on Email and Anti-Spam (CEAS)*, 2009.

[90] A. Bergholz, J. De Beer, S. Glahn, M-F. Moens, G. Paass, and S. Strobel. **New filtering approaches for phishing email**. *Journal of computer security*, **18**(1):7–35, 2010.

[91] M. Aburrous, M. A. Hossain, K. Dahal, and F. Thabtah. **Intelligent phishing detection system for e-banking using fuzzy data mining**. *Expert systems with applications*, **37**(12):7913–7921, 2010.

[92] M. He, S-J. Horng, P. Fan, M. K. Khan, R-S. Run, J-L. Lai, R-J. Chen, and A. Sutanto. **An efficient phishing webpage detector**. *Expert Systems with Applications*, **38**(10):12018–12027, 2011.

[93] X. Chen, I. Bose, A. C. M. Leung, and C. Guo. **Assessing the severity of phishing attacks: A hybrid data mining approach**. *Decision Support Systems*, **50**(4):662–672, 2011.

[94] R. M. Mohammad, F. Thabtah, and L. McCluskey. **An assessment of features related to phishing websites using an automated technique**. In *Internet Technology And Secured Transactions, 2012 International Conference for*, pages 492–497, 2012.

[95] M. Pandey and V. Ravi. **Text and data mining to detect phishing websites and spam emails**. In *International Conference on Swarm, Evolutionary, and Memetic Computing*, pages 559–573, 2013.

[96] M. Khonji, Y. Iraqi, and A. Jones. **Phishing detection: a literature survey**. *IEEE Communications Surveys & Tutorials*, **15**(4):2091–2121, 2013.

[97] N. Abdelhamid and F. Thabtah. **Associative classification approaches: review and comparison**. *Journal of Information & Knowledge Management*, **13**(03):1450027, 2014.

[98] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, G. Paliouras, and C. D. Spyropoulos. **An evaluation of naive bayesian anti-spam filtering**. *arXiv preprint cs/0006013*, 2000.

[99] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, and C.D. Spyropoulos. **An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages**. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–167. ACM, 2000.

[100] B. Massey, M. Thomure, R. Budrevich, and S. Long. **Learning Spam: Simple Techniques For Freely-Available Software.** In *USENIX Annual Technical Conference, FREENIX Track*, pages 63–76, 2003.

[101] L. Zhang, J. Zhu, and T. Yao. **An evaluation of statistical spam filtering techniques**. *ACM Transactions on Asian Language Information Processing (TALIP)*, **3**(4):243–269, 2004.

[102] B. Klimt and Y. Yang. **The enron corpus: A new dataset for email classification research**. In *European Conference on Machine Learning*, pages 217–226, 2004.

[103] V. Metsis, I. Androutsopoulos, and G. Paliouras. **Spam filtering with naive bayes-which naive bayes?** In *CEAS*, **17**, pages 28–69, 2006.

[104] A. Brodsky and D. Brodsky. **Trinitya: distributed defense against transient spam-bots**. In *Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing*, pages 378–379. ACM, 2007.

[105] C. Chen, Y. Tian, and C. Zhang. **Spam filtering with several novel bayesian classifiers**. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[106] E. Blanzieri and A. Bryl. **A survey of learning-based techniques of email spam filtering**. *Artificial Intelligence Review*, **29**(1):63–92, 2008.

[107] Thiago S Guzella and Walmir M Caminhas. **A review of machine learning approaches to spam filtering**. *Expert Systems with Applications*, **36**(7):10206–10222, 2009.

[108] D. E. Denning. **An intrusion-detection model**. *IEEE Transactions on software engineering*, (2):222–232, 1987.

[109] J. Zhan, B. J. Oommen, and J. Crisostomo. **Anomaly detection in dynamic systems using weak estimators**. *ACM Transactions on Internet Technology (TOIT)*, **11**(1):3, 2011.

[110] G. Caruana and M. Li. **A survey of emerging approaches to spam filtering**. *ACM Computing Surveys (CSUR)*, **44**(2):9, 2012.

[111] Y. Tan, G. Mi, Y. Zhu, and C. Deng. **Artificial immune system based methods for spam filtering**. In *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, pages 2484–2488, 2013.

[112] **Gartner's Magic Quadrant report**. `http://www.computerlinks.co.uk/FMS/22855.magic_quadrant_for_endpoint_protection_platforms.pdf`, 2011.

[113] K. Wang and S. J. Stolfo. **Anomalous payload-based network intrusion detection**. In *International Workshop on Recent Advances in Intrusion Detection*, pages 203–222. Springer, 2004.

[114] A. Vasudevan and R. Yerraballi. **Spike: engineering malware analysis tools using unobtrusive binary-instrumentation**. In *Proceedings of the 29th Australasian Computer Science Conference-Volume 48*, pages 311–320. Australian Computer Society, Inc., 2006.

[115] Y. Ye, D. Wang, T. Li, and D. Ye. **IMDS: Intelligent malware detection system**. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1043–1047, 2007.

[116] N. Idika and A. P. Mathur. **A survey of malware detection techniques**. *Purdue University*, **48**, 2007.

[117] F. Ahmed, H. Hameed, M. Z. Shafiq, and M. Farooq. **Using spatio-temporal information in API calls with machine learning algorithms for malware detection**. In *Proceedings of the 2nd ACM Workshop on Security and Artificial Intelligence*, pages 55–62, 2009.

[118] Y. Ye, T. Li, Y. Chen, and Q. Jiang. **Automatic malware categorization using cluster ensemble**. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 95–104. ACM, 2010.

[119] Y-T. Hou, Y. Chang, T. Chen, C-S. Laih, and C-M. Chen. **Malicious web content detection by machine learning**. *Expert Systems with Applications*, **37**(1):55–60, 2010.

[120] W. Zhuang, Y. Ye, Y. Chen, and T. Li. **Ensemble clustering for internet security applications**. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, **42**(6):1784–1796, 2012.

[121] G. G. Sundarkumar and V. Ravi. **Malware detection by text and data mining**. In *Computational Intelligence and Computing Research (ICCIC), 2013 IEEE International Conference on*, pages 1–6, 2013.

[122] G. Suarez-Tangil, J. E. Tapiador, P. Peris-Lopez, and J. Blasco. **Dendroid: A text mining approach to analyzing and classifying code structures in android malware families**. *Expert Systems with Applications*, **41**(4):1104–1117, 2014.

[123] G. G. Sundarkumar, V. Ravi, I. Nwogu, and V. Govindaraju. **Malware detection via API calls, topic models and machine learning**. In *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1212–1217. IEEE, 2015.

[124] Y. Liao and V. R. Vemuri. **Using text categorization techniques for intrusion detection.** In *USENIX Security Symposium*, **12**, pages 51–59, 2002.

[125] G. Helmer, J. S. K. Wong, V. Honavar, and L. Miller. **Automated discovery of concise predictive rules for intrusion detection**. *Journal of Systems and Software*, **60**(3):165–175, 2002.

[126] Z. Liu, G. Florez, and S. M. Bridges. **A comparison of input representations in neural networks: a case study in intrusion detection**. In *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on*, **2**, pages 1708–1713. IEEE, 2002.

[127] W-H. Chen, S-H. Hsu, and H-P. Shen. **Application of SVM and ANN for intrusion detection**. *Computers & Operations Research*, **32**(10):2617–2634, 2005.

[128] J. J. G. Adeva, J. M. Pikatza, S. Florez, and F. J. Sobrado. **Intrusion detection using text mining in a web-based telemedicine system**. *Lecture notes in computer science*, **3809**:1009, 2005.

[129] S. Rawat, V. P. Gulati, A. K. Pujari, and V. R. Vemuri. **Intrusion detection using text processing techniques with a binary-weighted cosine metric**. *Journal of Information Assurance and Security*, **1**(1):43–50, 2006.

[130] A. Sharma, A. K. Pujari, and K. K. Paliwal. **Intrusion detection using text processing techniques with a kernel based similarity measure**. *computers & security*, **26**(7):488–495, 2007.

[131] J. J. G. Adeva and J. M. P. Atxa. **Intrusion detection in web applications using text mining**. *Engineering Applications of Artificial Intelligence*, **20**(4):555–566, 2007.

[132] C. Y. Shirata, H. Takeuchi, S. Ogino, and H. Watanabe. **Extracting key phrases as predictors of corporate bankruptcy: Empirical analysis of annual reports by text mining**. *Journal of Emerging Technologies in Accounting*, **8**(1):31–44, 2011.

[133] S. Appavu, R. Rajaram, M. Muthupandian, G. Athiappan, and K. S. Kashmeera. **Data mining based intelligent analysis of threatening e-mail**. *Knowledge-Based Systems*, **22**(5):392–393, 2009.

[134] S. S. Kamaruddin, A. R. Hamdan, and A. A. Bakar. **Text mining for deviation detection in financial statements**. In *Proceedings of the International Conference on Electrical Engineering and Informatics, Institut Teknologi Bandung, Indonesia*, pages 17–19, 2007.

[135] M. Cecchini, H. Aytug, G. J. Koehler, and P. Pathak. **Making words work: Using financial text as a predictor of financial events**. *Decision Support Systems*, **50**(1):164–175, 2010.

[136] F. H. Glancy and S. B. Yadav. **A computational model for financial reporting fraud detection**. *Decision Support Systems*, **50**(3):595–601, 2011.

[137] P. Saha, I. Bose, and A. Mahanti. **A knowledge based scheme for risk assessment in loan processing by banks**. *Decision Support Systems*, **84**:78–88, 2016.

[138] P. Hajek and R. Henriques. **Mining corporate annual reports for intelligent detection of financial statement fraud–A comparative study of machine learning methods**. *Knowledge-Based Systems*, **128**:139–152, 2017.

[139] L. F. Rau, P. S. Jacobs, and U. Zernik. **Information extraction and text summarization using linguistic knowledge acquisition**. *Information Processing & Management*, **25**(4):419–428, 1989.

[140] A. Díaz and P. Gervás. **User-model based personalized summarization**. *Information Processing & Management*, **43**(6):1715–1734, 2007.

[141] U. Hahn and I. Mani. **The challenges of automatic summarization**. *Computer*, **33**(11):29–36, 2000.

[142] J. Kupiec, J. Pedersen, and F. Chen. **A trainable document summarizer**. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 68–73. ACM, 1995.

[143] J-Y. Yeh, H-R. Ke, W-P. Yang, and I-H. Meng. **Text summarization using a trainable summarizer and latent semantic analysis**. *Information processing & management*, **41**(1):75–95, 2005.

[144] S. Ye, T-S. Chua, M-Y. Kan, and L. Qiu. **Document concept lattice for text understanding and summarization**. *Information Processing & Management*, **43**(6):1643–1662, 2007.

[145] J. Zhan, H. T. Loh, and Y. Liu. **Gather customer concerns from online product reviews–A text summarization approach**. *Expert Systems with Applications*, **36**(2):2107–2115, 2009.

[146] M. A. Fattah and F. Ren. **GA, MR, FFNN, PNN and GMM based models for automatic text summarization**. *Computer Speech & Language*, **23**(1):126–144, 2009.

[147] M. S. Binwahlan, N. Salim, and L. Suanmali. **Fuzzy swarm diversity hybrid model for text summarization**. *Information processing & management*, **46**(5):571–588, 2010.

[148] Y. J. Kumar, N. Salim, A. Abuobieda, and A. T. Albaham. **Multi document summarization based on news components using fuzzy cross-document relations**. *Applied Soft Computing*, **21**:265–279, 2014.

[149] M. A. Mosa, A. Hamouda, and M. Marei. **Ant colony heuristic for user-contributed comments summarization**. *Knowledge-Based Systems*, **118**:105–114, 2017.

[150] H. Jeong, Y. Ko, and J. Seo. **How to Improve Text Summarization and Classification by Mutual Cooperation on an Integrated Framework**. *Expert Systems with Applications*, **60**:222–233, 2016.

[151] Z. Wu, L. Lei, G. Li, H. Huang, C. Zheng, E. Chen, and G. Xu. **A topic modeling based approach to novel document automatic summarization**. *Expert Systems with Applications*, **84**:12–23, 2017.

[152] M. Yousefi-Azar and L. Hamey. **Text summarization using unsupervised deep learning**. *Expert Systems with Applications*, **68**:93–105, 2017.

[153] R. Rautray and R. C. Balabantaray. **An evolutionary framework for multi document summarization using Cuckoo search approach: MDSCSA**. *Applied Computing and Informatics*, 2017.

[154] Y-H. Hu, Y-L Chen, and H-L. Chou. **Opinion mining from online hotel reviews–A text summarization approach**. *Information Processing & Management*, **53**(2):436–449, 2017.

[155] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. **From data mining to knowledge discovery in databases**. *AI magazine*, **17**(3):37, 1996.

[156] P. Willett. **Recent trends in hierarchic document clustering: a critical review**. *Information Processing & Management*, **24**(5):577–597, 1988.

[157] C. Apté, F. Damerau, and S. M. Weiss. **Automated learning of decision rules for text categorization**. *ACM Transactions on Information Systems (TOIS)*, **12**(3):233–251, 1994.

[158] L. Guthrie, E. Walker, and J. Guthrie. **Document classification by machine: theory and practice**. In *Proceedings of the 15th conference on Computational linguistics-Volume 2*, pages 1059–1063. Association for Computational Linguistics, 1994.

[159] L. S. Larkey and W. B. Croft. **Combining classifiers in text categorization**. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 289–297, 1996.

[160] Y. Yang and J. O. Pedersen. **A comparative study on feature selection in text categorization**. In *ICML*, **97**, pages 412–420, 1997.

[161] T. Joachims. **A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization**. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 143–151. Morgan Kaufmann Publishers Inc., 1997.

[162] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. **Inductive learning algorithms and representations for text categorization**. In *Proceedings of the seventh international conference on Information and knowledge management*, pages 148–155, 1998.

[163] A. McCallum and K. Nigam. **A comparison of event models for naive bayes text classification**. In *AAAI-98 workshop on learning for text categorization*, **752**, pages 41–48. Madison, WI, 1998.

[164] Y. Yang and X. Liu. **A re-examination of text categorization methods**. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49, 1999.

[165] E. Gabrilovich and S. Markovitch. **Text categorization with many redundant features: using aggressive feature selection to make SVMs competitive with C4. 5**. In *Proceedings of the twenty-first international conference on Machine learning*, page 41, 2004.

[166] **TechTC - Technion Repository of Text Categorization Datasets**. `http://techtc.cs.technion.ac.il/`, 2008. [Online; accessed 07-July-2013].

[167] **Rapid Miner**. `https://rapidminer.com`, 2013.

[168] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, K. Thiel, and B. Wiswedel. **KNIME-the Konstanz information miner: version 2.0 and beyond**. *AcM SIGKDD explorations Newsletter*, **11**(1):26–31, 2009.

[169] **Neuroshell**. `http://try.neuroshell.com/index/`, 2012.

[170] J. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, and P. S. Meltzer. **Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks**. *Nature medicine*, **7**(6):673–679, 2001.

[171] C. Gini. **Variabilità e mutabilità**. *Reprinted in Memorie di metodologica statistica (Ed. Pizetti E, Salvemini, T). Rome: Libreria Eredi Virgilio Veschi*, 1912.

[172] F. R. Helmert. *Mathematical and physical theories of higher geodesy*. Geo-Sciences Branch, Chart Research Division, Aeronautical Chart and Information Center, 1980.

[173] K. Pearson. **X. Contributions to the mathematical theory of evolution.—II. Skew variation in homogeneous material**. *Phil. Trans. R. Soc. Lond. A*, **38**(1):34–105, 1965.

[174] F. Bonchi, C. Castillo, A. Gionis, and A. Jaimes. **Social network analysis and mining for business applications**. *ACM Transactions on Intelligent Systems and Technology (TIST)*, **2**(3):22, 2011.

[175] K. Dasgupta, R. Singh, B. Viswanathan, D. Chakraborty, S. Mukherjea, A. A. Nanavati, and A. Joshi. **Social ties and their relevance to churn in mobile telecom networks**. In *Proceedings of the 11th international conference on Extending database technology: Advances in database technology*, pages 668–677, 2008.

[176] W. Verbeke, D. Martens, and B. Baesens. **Social network analysis for customer churn prediction**. *Applied Soft Computing*, **14**:431–446, 2014.

[177] N. Abdelhamid, A. Ayesh, and F. Thabtah. **Phishing detection based associative classification data mining**. *Expert Systems with Applications*, **41**(13):5948–5959, 2014.

[178] M. Pandey and V. Ravi. **Detecting phishing e-mails using text and data mining**. In *Computational Intelligence & Computing Research (ICCIC), 2012 IEEE International Conference on*, pages 1–6, 2012.

[179] W. Lee and S. J. Stolfo. **Data Mining Approaches for Intrusion Detection.** In *Usenix security*, 1998.

[180] C. H. Li and S. C. Park. **An efficient document classification model using an improved back propagation neural network and singular value decomposition**. *Expert Systems with Applications*, **36**(2):3208–3215, 2009.

[181] W. Song and S. C. Park. **Genetic algorithm for text clustering based on latent semantic indexing**. *Computers & Mathematics with Applications*, **57**(11):1901–1907, 2009.

[182] D. Thorleuchter and D. Van den Poel. **Technology classification with latent semantic indexing**. *Expert Systems with Applications*, **40**(5):1786–1795, 2013.

[183] **Legitimate Emails**. `http://spamassassin.apache.org/`, 2012.

[184] **Phishing Corpus**. `http://monkey.org/~jose/wiki/doku.php`, 2012.

[185] **Phishing Tank**. `http://www.phishtank.com`, 2012.

[186] **IBM SPSS**. `http://www-01.ibm.com/software/in/analytics/spss/products/data-collection/`.

[187] **CSMINING GROUP**. `http://www.csmining.org/index.php/malicious-software-datasets-.html`, 2010.

[188] **MathWorks-MATLAB**. `www.mathworks.com`, 2012.

[189] **Library for Support Vector Machine (LIBSVM)**. `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`, 2014.

[190] T. Joachims. **Text categorization with support vector machines: Learning with many relevant features**. *Machine learning: ECML-98*, pages 137–142, 1998.

[191] B. Masand, G. Linoff, and D. Waltz. **Classifying news stories using memory based reasoning**. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 59–65, 1992.

[192] D. D. Lewis and W. A. Gale. **A sequential algorithm for training text classifiers**. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12, 1994.

[193] H. Taira and M. Haruno. **Feature selection in SVM text categorization**. In *AAAI/IAAI*, pages 480–486, 1999.

[194] G. Forman. **An extensive empirical study of feature selection metrics for text classification**. *Journal of machine learning research*, **3**(Mar):1289–1305, 2003.

[195] R. Vert and J-P. Vert. **Consistency and convergence rates of one-class SVMs and related algorithms**. *Journal of Machine Learning Research*, **7**(May):817–854, 2006.

[196] M. Lan, C. L. Tan, J. Su, and Y. Lu. **Supervised and traditional term weighting methods for automatic text categorization**. *IEEE transactions on pattern analysis and machine intelligence*, **31**(4):721–735, 2009.

[197] S. Jun, S-S. Park, and D-S. Jang. **Document clustering method using dimension reduction and support vector clustering to overcome sparseness**. *Expert Systems with Applications*, **41**(7):3204–3212, 2014.

[198] G. G. Sundarkumar and V. Ravi. **A novel hybrid undersampling method for mining unbalanced datasets in banking and insurance**. *Engineering Applications of Artificial Intelligence*, **37**:368–377, 2015.

[199] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[200] L. Ferré. **Selection of components in principal component analysis: a comparison of methods**. *Computational Statistics & Data Analysis*, **19**(6):669–682, 1995.

[201] H. Yu, J. Han, and K. C-C. Chang. **PEBL: positive example based learning for web page classification using SVM**. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 239–248, 2002.

[202] F. Denis, R. Gilleron, and M. Tommasi. **Text classification from positive and unlabeled examples**. In *Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'02*, pages 1927–1934, 2002.

[203] W. S. Lee and B. Liu. **Learning with positive and unlabeled examples using weighted logistic regression**. In *ICML*, **3**, pages 448–455, 2003.

[204] L. M Manevitz and M. Yousef. **One-class SVMs for document classification**. *Journal of Machine Learning Research*, **2**(Dec):139–154, 2001.

[205] C. Elkan and K. Noto. **Learning classifiers from only positive and unlabeled data**. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 213–220, 2008.

[206] H. Boström. **Predicate invention and learning from positive examples only**. *Machine Learning: ECML-98*, pages 226–237, 1998.

[207] S. Muggleton. **Learning from positive data**. *Inductive logic programming*, pages 358–376, 1997.

[208] B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu. **Building text classifiers using positive and unlabeled examples**. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 179–186, 2003.

[209] L. M. Manevitz and M. Yousef. **A web navigation system based on a neural network user-model trained with only positive web documents**. *Web Intelligence and Agent Systems: An International Journal*, **2**(2):137–144, 2004.

[210] **Syskill and Webert web page ratings**. `https://archive.ics.uci.edu/ml/machine-learning-databases/SyskillWebert-mld/`, 2012.

[211] T. W. Anderson. **Asymptotic theory for principal component analysis**. *The Annals of Mathematical Statistics*, **34**(1):122–148, 1963.

[212] C. J. C. Burges. **Dimension reduction: A guided tour**. *Foundations and Trends® in Machine Learning*, **2**(4):275–365, 2010.

[213] H. Lian. **On feature selection with principal component analysis for one-class SVM**. *Pattern Recognition Letters*, **33**(9):1027–1031, 2012.

[214] A. K. Jain, M. N. Murty, and P. J. Flynn. **Data clustering: a review**. *ACM computing surveys (CSUR)*, **31**(3):264–323, 1999.

[215] M. Steinbach, G. Karypis, and V. Kumar. **A comparison of document clustering techniques**. In *KDD workshop on text mining*, **400**, pages 525–526. Boston, 2000.

[216] T. Liu, S. Liu, Z. Chen, and W-Y. Ma. **An evaluation on feature selection for text clustering**. In *Icml*, **3**, pages 488–495, 2003.

[217] Y. Zhao and G. Karypis. **Empirical and theoretical comparisons of selected criterion functions for document clustering**. *Machine Learning*, **55**(3):311–331, 2004.

[218] L. Liu, J. Kang, J. Yu, and Z. Wang. **A comparative study on unsupervised feature selection methods for text clustering**. In *Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE'05. Proceedings of 2005 IEEE International Conference on*, pages 597–601. IEEE, 2005.

[219] N. O. Andrews and E. A. Fox. **Recent developments in document clustering**. Technical report, Technical report, Computer Science, Virginia Tech, 2007.

[220] F. Gelgi, H. Davulcu, and S. Vadrevu. **Term Ranking for Clustering Web Search Results.** In *WebDB*, 2007.

[221] V. Jadhao and M. N. Murty. **Hybrid online non-negative matrix factorization for clustering of documents**. In *International Conference on Neural Information Processing*, pages 516–523. Springer, 2012.

[222] C. C. Aggarwal and C. Zhai. **A survey of text clustering algorithms**. In *Mining text data*, pages 77–128. Springer, 2012.

[223] P. Shamsinejadbabki and M. Saraee. **A new unsupervised feature selection method for text clustering based on genetic algorithms**. *Journal of Intelligent Information Systems*, **38**(3):669–684, 2012.

[224] K. K. Bharti and P. K. Singh. **Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering**. *Expert Systems with Applications*, **42**(6):3105–3114, 2015.

[225] T. Basu and C. A. Murthy. **A similarity assessment technique for effective grouping of documents**. *Information Sciences*, **311**:149–162, 2015.

[226] K. K. Bharti and P. K. Singh. **Chaotic gradient artificial bee colony for text clustering**. *Soft Computing*, **20**(3):1113–1126, 2016.

[227] L. M. Abualigah, A. T. Khader, M. A. Al-Betar, and O. A. Alomari. **Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering**. *Expert Systems with Applications*, **84**:24–36, 2017.

[228] S. Lloyd. **Least squares quantization in PCM**. *IEEE transactions on information theory*, **28**(2):129–137, 1982.

[229] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[230] P. J. Rousseeuw and L. Kaufman. *Finding Groups in Data*. Wiley Online Library, 1990.

[231] T. Kohonen. **The self-organizing map**. *Neurocomputing*, **21**(1):1–6, 1998.

[232] T. Kohonen. *Self-organizing maps. Series in information sciences, vol. 30*. Springer, Heidelberg, 1995.

[233] J. C. Dunn. **A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters**. 1973.

[234] J. C. Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media, 2013.

[235] G. Salton and M. J. McGill. **Introduction to modern information retrieval**. 1986.

[236] D. M. Blei, A. Y. Ng, and M. I. Jordan. **Latent dirichlet allocation**. *Journal of machine Learning research*, **3**(Jan):993–1022, 2003.

[237] D. M. Blei. **Probabilistic topic models**. *Communications of the ACM*, **55**(4):77–84, 2012.

[238] R. Agrawal, T. Imieliński, and A. Swami. **Mining association rules between sets of items in large databases**. In *Acm sigmod record*, **22**, pages 207–216. ACM, 1993.

[239] R. Feldman and H. Hirsh. **Mining Associations in Text in the Presence of Background Knowledge.** In *KDD*, pages 343–346, 1996.

[240] X. Wang, K. Yue, W. Niu, and Z. Shi. **An approach for adaptive associative classification**. *Expert Systems with Applications*, **38**(9):11873–11883, 2011.

[241] N. Abdelhamid, A. Ayesh, F. Thabtah, S. Ahmadi, and W. Hadi. **MAC: A multiclass associative classification algorithm**. *Journal of Information & Knowledge Management*, **11**(02):1250011, 2012.

[242] B. Lent, R. Agrawal, and R. Srikant. **Discovering Trends in Text Databases.** In *KDD*, **97**, pages 227–230, 1997.

[243] B. L. W. H. Y. Ma and B. Liu. **Integrating classification and association rule mining**. In *Proceedings of the fourth international conference on knowledge discovery and data mining*, 1998.

[244] X. Yin and J. Han. **CPAR: Classification based on predictive association rules**. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, pages 331–335. SIAM, 2003.

[245] F. A. Thabtah, P. Cowling, and Y. Peng. **MMAC: A new multi-class, multi-label associative classification approach**. In *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*, pages 217–224. IEEE, 2004.

[246] W. Li, J. Han, and J. Pei. **CMAR: Accurate and efficient classification based on multiple class-association rules**. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 369–376. IEEE, 2001.

[247] C. Wang and D. M. Blei. **Collaborative topic modeling for recommending scientific articles**. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 448–456. ACM, 2011.

[248] D. Gujraniya and M. N. Murty. **Efficient classification using phrases generated by topic models**. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2331–2334. IEEE, 2012.

[249] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff. **A sense of self for unix processes**. In *Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on*, pages 120–128, 1996.

[250] D. K. S. Reddy and A. K. Pujari. **N-gram analysis for computer virus detection**. *Journal in Computer Virology*, **2**(3):231–239, 2006.

[251] M. Shankarapani, K. Kancherla, S. Ramammoorthy, R. Movva, and S. Mukkamala. **Kernel machines for malware classification and similarity analysis**. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–6, 2010.

[252] A. Sami, B. Yadegari, H. Rahimi, N. Peiravian, S. Hashemi, and A. Hamze. **Malware detection based on mining API calls**. In *Proceedings of the 2010 ACM symposium on applied computing*, pages 1020–1025, 2010.

[253] P. O'Kane, S. Sezer, K. McLaughlin, and E. G. Im. **SVM training phase reduction using dataset feature filtering for malware detection**. *IEEE transactions on information forensics and security*, **8**(3):500–509, 2013.

[254] Y. Fan, Y. Ye, and L. Chen. **Malicious sequential pattern mining for automatic malware detection**. *Expert Systems with Applications*, **52**:16–25, 2016.

[255] H. Hashemi, A. Azmoodeh, A. Hamzeh, and S. Hashemi. **Graph embedding as a new approach for unknown malware detection**. *Journal of Computer Virology and Hacking Techniques*, pages 1–14, 2016.

[256] S. Huda, S. Miah, M. M. Hassan, R. Islam, M. Yearwood, J.and Alrubaian, and A. Almogren. **Defending unknown attacks on cyber-physical systems by semi-supervised approach and available unlabeled data**. *Information Sciences*, **379**:211–228, 2017.

[257] Z. Salehi, A. Sami, and M. Ghiasi. **MAAR: Robust features to detect malicious activity based on API calls, their arguments and return values**. *Engineering Applications of Artificial Intelligence*, **59**:93–102, 2017.

[258] R. Agrawal, A. Fuxman, A. Kannan, J. Shafer, and P. P. Talukdar. **Associating structured records to text documents**. In *Proceedings of the 21st International Conference on World Wide Web*, pages 451–452. ACM, 2012.

[259] **Windows API calls**. `http://msdn.microsoft.com/en-us/library/aa383723(VS.85).aspx`, 2016.

[260] **Malware dataset**. `http://nexgeneric.org/Datasets/Default.aspx`, 2016.

[261] **R Studio**. `https://www.rstudio.com`, 2014.

[262] R. Agrawal and R. Srikant. **Fast algorithms for mining association rules**. In *Proc. 20th int. conf. very large data bases, VLDB*, **1215**, pages 487–499, 1994.

[263] R. Agrawal and R. Srikant. **Mining sequential patterns**. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*, pages 3–14. IEEE, 1995.

[264] Y. Abu and A. A. Mostafa. **Introduction to financial forecasting–Applied Intelligence, 6 (3): 205–213**, 1996.

[265] R. F. Engle and V. K. Ng. **Measuring and testing the impact of news on volatility**. *The journal of finance*, **48**(5):1749–1778, 1993.

[266] V. Lavrenko, M. Schmill, D. Lawrie, P. Ogilvie, D. Jensen, and J. Allan. **Mining of concurrent text and time series**. In *KDD-2000 Workshop on Text Mining*, pages 37–44, 2000.

[267] J. D. Thomas and K. Sycara. **Integrating genetic algorithms and text learning for financial prediction**. *Data Mining with Evolutionary Algorithms*, pages 72–75, 2000.

[268] D. Peramunetilleke and R. K. Wong. **Currency exchange rate forecasting from news headlines**. *Australian Computer Science Communications*, **24**(2):131–139, 2002.

[269] G. Rachlin, M. Last, D. Alberg, and A. Kandel. **ADMIRAL: A data mining based financial trading system**. In *Computational Intelligence and Data Mining, 2007. CIDM 2007. IEEE Symposium on*, pages 720–725, 2007.

[270] Y. Zhai, A. Hsu, and S. Halgamuge. **Combining news and technical indicators in daily stock price trends prediction**. *Advances in Neural Networks–ISNN 2007*, pages 1087–1096, 2007.

[271] X. Li, C. Wang, J. Dong, F. Wang, X. Deng, and S. Zhu. **Improving stock market prediction by integrating both market news and stock prices**. In *Database and Expert Systems Applications*, pages 279–293, 2011.

[272] X. Li, H. Xie, R. Wang, Y. Cai, J. Cao, F. Wang, and X. Min, H.and Deng. **Empirical analysis: stock market prediction via extreme learning machine.** *Neural Computing and Applications*, **27**(1):67–78, 2016.

[273] Y. Shynkevich, T. M. McGinnity, S. A. Coleman, and A. Belatreche. **Forecasting movements of health-care stock prices based on different categories of news articles using multiple kernel learning**. *Decision Support Systems*, **85**:74–83, 2016.

[274] K-Y. Ho and W. W. Wang. **Predicting stock price movements with news sentiment: An artificial neural network approach**. In *Artificial Neural Network Modelling*, pages 395–403. 2016.

[275] Q. Li, L. Jiang, P. Li, and H. Chen. **Tensor-Based Learning for Predicting Stock Movements.** In *AAAI*, pages 1784–1790, 2015.

[276] **Linguistic Inquiry and Word Count (LIWC)**. `http://www.liwc.net/`, 2015.

[277] K. Kyle, S. Crossley, and C. Berger. **The tool for the automatic analysis of lexical sophistication (TAALES): version 2.0**. *Behavior Research Methods*, pages 1–17, 2017.

[278] V. Ravi, K.and Ravi. **A novel automatic satire and irony detection using ensembled feature selection and data mining**. *Knowledge-Based Systems*, **120**:15–33, 2017.

[279] Y. R. Tausczik and J. W. Pennebaker. **The psychological meaning of words: LIWC and computerized text analysis methods**. *Journal of language and social psychology*, **29**(1):24–54, 2010.

[280] J. W. Pennebaker, R. L. Boyd, K. Jordan, and K. Blackburn. **The development and psychometric properties of LIWC2015**. Technical report, 2015.

[281] K. Kyle and S. A. Crossley. **Automatically assessing lexical sophistication: Indices, tools, findings, and application**. *Tesol Quarterly*, **49**(4):757–786, 2015.

[282] **Yahoo Finance - Business finance, stock market, quotes, news**. `https://in.finance.yahoo.com/`, 2016.

[283] **Web Scraper**. `http://webscraper.io/`, 2016.

[284] R. J. Little. **A and Rubin, DB (1987) Statistical Analysis with Missing Data**. *John A. Wiley & Sons, Inc., New York*, 85.

[285] K. J. Nishanth and V. Ravi. **Probabilistic neural network based categorical data imputation**. *Neurocomputing*, **218**:17–25, 2016.

[286] C. Gautam and V. Ravi. **Counter propagation auto-associative neural network based data imputation**. *Information Sciences*, **325**:288–299, 2015.

[287] V. Ravi and M. Krishna. **A new online data imputation method based on general regression auto associative neural network**. *Neurocomputing*, **138**:106–113, 2014.

[288] W. Ling and F. Dong-Mei. **Estimation of missing values using a weighted k-nearest neighbors algorithm**. In *Environmental Science*

and Information Application Technology, 2009. ESIAT 2009. International Conference on, **3**, pages 660–663, 2009.

[289] B. M. Patil, R. C. Joshi, and D. Toshniwal. **Missing value imputation based on K-mean clustering with weighted distance**. In *Contemporary Computing*. 2010.

[290] P. J. García-Laencina, J-L. Sancho-Gómez, A. R. Figueiras-Vidal, and M. Verleysen. **K nearest neighbours with mutual information for simultaneous classification and missing data imputation**. *Neurocomputing*, **72**(7):1483–1493, 2009.

[291] B. E. Flores. **A pragmatic view of accuracy measurement in forecasting**. *Omega*, **14**(2):93–98, 1986.

[292] **Statistica**. `https://software.dell.com/register/72480`, 2016.

[293] Z. Zheng, X. Wu, and R. Srihari. **Feature selection for text categorization on imbalanced data**. *ACM Sigkdd Explorations Newsletter*, **6**(1):80–89, 2004.

[294] H. Peng, F. Long, and C. Ding. **Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy**. *IEEE Transactions on pattern analysis and machine intelligence*, **27**(8):1226–1238, 2005.

[295] C. Ding and H. Peng. **Minimum redundancy feature selection from microarray gene expression data**. *Journal of bioinformatics and computational biology*, **3**(02):185–205, 2005.

[296] F. X. Diebold and R. S. Mariano. **Comparing predictive accuracy**. *Journal of Business & economic statistics*, **20**(1):134–144, 2002.

[297] **DM Test**. `https://cran.r-project.org/web/packages/forecast/forecast.pdf`, 2017.

[298] H. Chen, Q. Wan, and Y. Wang. **Refined Diebold-Mariano test methods for the evaluation of wind power forecasting models**. *Energies*, **7**(7):4185–4198, 2014.

[299] V. Srikrishna, R. Ghosh, V. Ravi, and K. Deb. **Elitist quantum-inspired differential evolution based wrapper for feature subset selection**. In *International Workshop on Multi-disciplinary Trends in Artificial Intelligence*, pages 113–124, 2015.

[300] G. J. Krishna and V. Ravi. **Evolutionary computing applied to customer relationship management: A survey**. *Engineering Applications of Artificial Intelligence*, **56**:30–59, 2016.

[301] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

[302] J. R. Quinlan. **Induction of decision trees**. *Machine learning*, **1**(1):81–106, 1986.

[303] D. F. Specht. **A general regression neural network**. *IEEE transactions on neural networks*, **2**(6):568–576, 1991.

[304] D. Pradeepkumar and V. Ravi. **FOREX Rate Prediction Using Chaos, Neural Network and Particle Swarm Optimization**. In *International Conference in Swarm Intelligence*, pages 363–375, 2014.

[305] R. Mohanty, V. Ravi, and M. R. Patra. **Hybrid intelligent systems for predicting software reliability**. *Applied Soft Computing*, **13**(1):189–200, 2013.

[306] A. G. Ivakhnenko. **The group method of data handling-a rival of the method of stochastic approximation**. *Soviet Automatic Control*, **13**(3):43–55, 1968.

[307] L. Kaufman and P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, **344**. John Wiley & Sons, 2009.

[308] E. Fix and J. L. Hodges Jr. **Discriminatory analysis-nonparametric discrimination: consistency properties**. Technical report, DTIC Document, 1951.

[309] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman. **Using latent semantic analysis to improve access to textual information**. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 281–285, 1988.

[310] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. **Learning internal representations by error propagation**. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[311] Y. Chen, X. Sean Zhou, and T. S. Huang. **One-class SVM for learning in image retrieval**. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, **1**, pages 34–37, 2001.

[312] C. Liu, G. Wang, W. Ning, X. Lin, L. Li, and Z. Liu. **Anomaly detection in surveillance video using motion direction statistics**. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 717–720, 2010.

[313] C. Wan and A. Mita. **An automatic pipeline monitoring system based on PCA and SVM**. 2008.

[314] D. F. Specht. **Probabilistic neural networks**. *Neural networks*, **3**(1):109–118, 1990.

[315] N. Meinshausen. **Quantile regression forests**. *Journal of Machine Learning Research*, **7**(Jun):983–999, 2006.

[316] T. K. Ho. **Random decision forests**. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, **1**, pages 278–282, 1995.

[317] W. W. Cohen. **Fast effective rule induction**. In *Proceedings of the twelfth international conference on machine learning*, pages 115–123, 1995.

[318] V. Vapnik. *Statistical learning theory*, **1**. Wiley, New York, 1998.

[319] S. R. Gunn. **Support vector machines for classification and regression**. *ISIS technical report*, **14**:85–86, 1998.

[320] H. Yang, L. Chan, and I. King. **Support vector machine regression for volatile stock market prediction**. *Intelligent Data Engineering and Automated Learning—IDEAL 2002*, pages 143–152, 2002.

[321] D. C. Sansom, T. Downs, and T. K. Saha. **Evaluation of support vector machine based forecasting tool in electricity price forecasting for Australian national electricity market participants**. *Journal of Electrical & Electronics Engineering, Australia*, **22**(3):227, 2003.

[322] C-H. Wu, J-M. Ho, and D-T. Lee. **Travel-time prediction with support vector regression**. *IEEE transactions on intelligent transportation systems*, **5**(4):276–281, 2004.

[323] J. S. Pahariya, V. Ravi, and M. Carr. **Software cost estimation using computational intelligence techniques**. In *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, pages 849–854, 2009.

[324] G. Sharma and M. N. Murty. **Mining Sentiments from Songs Using Latent Dirichlet Allocation.** In *IDA*, pages 328–339, 2011.

[325] H. U. Asuncion, A. U. Asuncion, and R. N. Taylor. **Software traceability with topic modeling**. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1*, pages 95–104, 2010.

[326] K. Nagesh and M. N. Murty. **Obtaining single document summaries using latent dirichlet allocation**. In *Neural Information Processing*, pages 66–74, 2012.

[327] M. Rosen-Zvi, C. Chemudugunta, T. Griffiths, P. Smyth, and M. Steyvers. **Learning author-topic models from text corpora**. *ACM Transactions on Information Systems (TOIS)*, **28**(1):4, 2010.

[328] Z. Chen and B. Liu. **Topic modeling using topics from many domains, lifelong learning and big data**. In *International Conference on Machine Learning*, pages 703–711, 2014.

[329] L. Yao, Y. Zhang, B. Wei, H. Qian, and Y. Wang. **Incorporating probabilistic knowledge into topic models**. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 586–597. Springer, 2015.

List of Publications

[1] **B. Shravan Kumar**, and Vadlamani Ravi, Text Classification Using Ensemble Features Selection and Data Mining Techniques, *Proceedings of the* 5^{th} *Internationcal Conference on Swarm, Evolutionary, and Memetic Computing, SEMCCO-2014*, 18-20 December 2014, Bhubaneswar, India, LNCS 8947, pp. 176-186, 2015. (Indexed by SCOPUS, DBLP, ACM Digital Library and Springerlink).

[2] **B. Shravan Kumar**, and Vadlamani Ravi, One-Class Text Document Classification with OCSVM and LSI, *Proceedings of the* 2^{nd} *Internationcal conference on Artificial Intelligence and Evolutionary Computations in Engineering Systems, ICAIECES-2016*, 19-21 May 2016, Chennai, India, LNCS 517 , pp. 597-606, 2016. (Indexed by SCOPUS, and Springerlink).

[3] **B. Shravan Kumar**, and Vadlamani Ravi, Text Document Classification with PCA and One-Class SVM, *Proceedings of the* 5^{th} *International Conference on Frontiers in Intelligent Computing Theory & Applications, FICTA-2016*, 16-17 September 2016, Bhubaneswar, India, pp. 107-115, 2016. (Indexed by ISI Proceedings, EI-Compendex, DBLP, SCOPUS, and Springerlink).

[4] **B. Shravan Kumar**, and Vadlamani Ravi, A Survey of the Applications of Text Mining in Financial Domain, *Knowledge-Based Systems*, Vol. 114, pp. 128-147, 2016, Elsevier. (Indexed by SCI, Scopus, and SCISEARCH).

[5] **B. Shravan Kumar**,and Vadlamani Ravi, LDA Based Feature Selection for Document Clustering, *Proceedings of the* 10^{th} *Annual ACM India Conference, COMPUTE-2017*, 16-17 November 2017, Bhopal, India. (Indexed by DBLP, and ACM Digital Library).

[6] **B. Shravan Kumar**, and Vadlamani Ravi, A Hybrid approach using topic modeling and class-association rule mining for text classification: The case of malware detection, *Proceedings of the* 17^{th} *IEEE International Conference on Cognitive Informatics & Cognitive Computing, ICCI*CC 2018*, 16-18 July 2018, UC Berkeley, USA. (indexed by DBLP, EI and IEEE Xplore)

[7] **B. Shravan Kumar**,Vadlamani Ravi, and Rishabh Miglani, Predicting Indian Stock Market using the Psycho-Linguistic Features of Financial News, *To be Communicated*.

Appendix A

Overview of Techniques Used

This appendix presents various techniques used in the thesis in alphabetical order.

A.1 Association Rule Mining (ARM)

Association rule mining (Agrawal et al. [238]; Agrawal and Srikant [262]) detects a set of association rules existing in the database with the minimum support and confidence that specified already. Association rule mining works as follows: first, it finds left-hand and right-hand associations; second, the support and the confidence of an association rule will be estimated in order to determine interesting rules. An association rule has the form of $X \rightarrow Y$ (if X then Y), where X and Y are two sets of items. Here, X and Y are also known as antecedent or premise and consequent or conclusion respectively. Association rule mining found numerous applications in fields like health, marketing, financial (credit card transactions) etc. For detailed explanations, an interested reader can refer to the articles of (Agrawal et al. [238]; Agrawal and Srikant [262]) and sequential pattern mining by Agrawal and Srikant [263].

A.2 Classification and Regression Trees (CART)

Classification and Regression Tree (CART) is a decision tree that can solve both classification and regression problems (Breiman et al. [301]) which is not possible with other decision trees. It can also generate "if-then" rules. It has the ability to deal with real values, enumerated type data and missing data too. Like general decision trees, it follows the same procedure of splitting node except that it can produce either discrete or continuous output.

A.3 Decision Tree (DT)

Decision Tree [302] is a popular supervised learning method used in many data mining applications. It is made up of nodes including a non-terminal node representing test condition and a terminal node (leaf node) representing class output. The root node of a decision tree is selected using various splitting criteria including Gini Index, Information Gain, Gain Ratio, or Entropy. It not only classifies the data but it also produces "if-then" rules influencing decision making.

A.4 Fuzzy C-means

It is the clustering algorithm proposed by Dunn [233] in 1973. It is also frequently using for clustering applications. In this approach, data points are bounded in the cluster by its membership function mean value, and it is having fuzzy behavior. For detail refer Bezdek [234] article.

A.5 General Regression Neural Network (GRNN)

GRNN, proposed by Specht [303] is a neural network that implements non-parametric regression procedure over the data. It consists of input, pattern, summation and output layers in that order. It has the following features: quick

learning, easy training, and outlier discrimination. It can approximate any function from the past data. GRNN is widely used in various applications including FOREX rate prediction (Pradeepkumar and Ravi [304]), Software Reliability Prediction (Mohanty et al. [305]).

A.6 Group Method Data Handling (GMDH)

GMDH, the first deep learning neural network, proposed by Ivakhnenko ([306]), is a powerful neural network architecture that can solve complex problems. Basically, its functionality is based on polynoimal terms and its output depends on the combination (polynomial) of the inputs. It builds polynomials repeatedly, and the algorithm selects the best one among them. The process is complete only when the algorithm meets the selection criteria.

A.7 k-Means

K-means algorithm proposed by Lloyd [228] in 1957 (published in 1982). It is easy to implement, simple. For that reason, it is widely using in clustering tasks even though it was proposed long ago. It has following disadvantages: Initial centers choosing and the number of iterations. For further details of the algorithm, users can see the works of (Jain and Dubels [229]).

A.8 k-Medoids

It is also known as Partitioning Around Medoids (PAM), proposed by Kaufman and Rouseeuw [307]. It is similar to K-means algorithm instead of centroids it considers the data points/ objects as medoids which are centrally located objects in the cluster. When the data set is large it may not work efficient manner due to its time complexity it is the drawback of this algorithm.

A.9 k-Nearest Neighbors (k-NN)

The k-NN algorithm aims at finding the k nearest neighbors by computing the distance between unlabled and labeled data samples [308]. In k-NN, three components including data samples, distance metric and number of the neighbors (k) play key role. For any classification task, firstly, it computes the distance and later labeled data sample is assigned to the nearest labeled sample using the distance computed.

A.10 Latent Semantic Indexing (LSI)

Dumais et al. [309] proposed the Latent Semantic Indexing for accessing the textual information so that hidden concepts in document data are discovered. Each document and term (word) are expressed as a vectors and each element in a vector gives the degree of participation of the document or term in the corresponding concept. The goal of LSI is not to describe the concepts verbally, but to represent the documents and terms in a unified way for exposing document-document, document-term, and term-term similarities.

A.11 Multilayer Perceptron (MLP)

MLP [310] is a well-known neural network that can solve both classification and regression problems. In this neural network, input layer accepts the input variables, hidden layer(s) accept the outputs from input layer and the outputs of hidden layers along with weights are used to produce output (prediction/class label) at output layer. It is trained using backpropagation algorithm. It is too popular to be described here.

A.12 Naive Bayes (NB)

Naive Bayes is suitable to solve both binary and multi-class problems [12]. The algorithm is based on popular theorem of probability namely Bayes theorem. It

is naive as it assumes all predictors are independent which is not always true. As it is good at prediction, it is employed quite frequently.

A.13 One-Class Support Vector Machine (OCSVM)

OCSVM is similar to SVM except that it deals with a training data consisting of only one class. It builds the boundary space from other class examples. It can be applied in high-dimensional settings when other methods (e.g., density estimation) fail. It is also applied to solve anomaly detection problems and recent interesting application is to speaker diarization (partitioning the audio stream into homogeneous segments) problem. It is being used in retrieval of images [311], anomaly detection in videos [312], document categorization [204] and sound recognition system [313]. Its disadvantages lie in the selection of kernels and inability to perform multiclass classification.

A.14 Principal Component Analysis (PCA)

PCA [211][199] [200] [212] [213]) is the best suitable statistical technique applied in various fields like Image processing, Signal processing, pattern recognition, engineering and sciences to reduce dimensionality and thereby removing the multicollinearity in datasets. It depends on the concept of Principle Component (PC) which is a linear combination of the original features. In PCA, the first PC obtained explains maximum variance (information), and then the second PC explains the second highest variance (information) in the original data and so on. Consequently, if one selects first few PCs then, he is assured of maximum variance accounted for. As a result, one can achieve feature space dimensionality reduction.

A.15 Probabilistic Neural Network (PNN)

It is one of the neural network models. It can be applicable for binary classification as well as multi classification problems. In this it uses the exponential function as activation function. It uses the probability density function. It is much faster than back propagation algorithms [314].

A.16 Quantile Regression Random Forest (QRRF)

QRRF was introduced by Meinshausen [315]. The significant difference between RF and QRRF is as follows: All observations are kept in a node in QRRF whereas in the RF, a node contains the mean of observations only. It is just like an optimization problem i.e. conditional mean estimation is performed by minimizing the squared error so that quantiles reduce the expected loss. Selection of suitable parameters for quantile regression minimizes the empirical loss. The QRRF is non-parametric and yields accurate predictions.

A.17 Random Forest (RF)

Ho [316] proposed Random Forest. It builds multiple trees on a randomly selected feature subset on a sample of data obtained with replacement (also known as bootstrap sampling). It isexpandable for increasing the performance on both training and test data. It performs both classification and regression and also handles higher dimensions of the datasets.

A.18 Repeated Incremental Pruning to Produce Error Reduction (RIPPER)

RIPPER, proposed by William Cohen [317], is a propositional rule learner that builds a rule set using sequential covering algorithm. For further details, the reader is referred to [317].

A.19 Self-Organizing Maps (SOM)

Self-Organizing Map structure was introduced by Kohonen [231]. It is one of the traditional neural network models. It has the following advantages: (i) It identifies the features inherent to the data. So, we can also call it as Self-Organizing Feature Map. (ii) It can recognize the patterns that have never occurred before i.e. generalization. For further details refer Kohonen [232], [231].

A.20 Support Vector Machine (SVM)

SVM, proposed by Vapnik [318], performs the classification task by constructing the hyperpalne in such a way that linearly separable data will be classified into two categories. For dealing with non-linear data, it uses the kernel (sigmoid, radial, polynomial) function for projection of the data into higher dimensions so that data can be linearly separable. It identifies the support vectors such that the distances between them were maximum. Its performance depends on the selection of the kernel and its user-defined parameters.

A.21 Support Vector Regression (SVR)

SVM [11] proved useful for solving classification problems. However, Support Vector Regression (SVR) [319] uses the same methodology as that of SVM barring few changes to solve regression problems. SVR is being employed in various applications including power consumption estimation, financial market forecasting [320], electricity price [321], travel time prediction [322] and software cost estimation [323].

A.22 Topic Modeling/ Latent Dirichlet Allocation (LDA)

Topic modeling [236] or Latent Dirichlet Allocation (LDA) is a probabilistic Bayesian model. It provides an explicit representation of a document and is

widely used in various applications to discover the topics in the documents. It preserves the essential characteristics of text which are useful for regular data mining tasks. For further details on LDA, the reader is referred to Blei et al. [237]. Numerous applications developed using topic modeling include recommender systems [324], software traceability [325], and document summarization [326]. There are also other works on LDA found including Rosen-Zvi et al. [327], Chen and Liu [328], and Yao et al. [329]).

Appendix B

Annexure (Publications Online)

Table B.1: Fact Sheet of SEMCCO-2014 Publication [1]

Title	Text Classification Using Ensemble Features Selection and Data Mining Techniques
Authors	B. Shravan Kumar and Vadlamani Ravi
Publication	In the poceedings of 5^{th} International Conference, SEMCCO 2014, Bhubaneswar, India, December 18-20, 2014, LNCS 8947, pp. 176-186, 2015
DOI	10.1007/978-3-319-20294-5_16
Status	Published
Publisher	Springer International Publishing
Publication Type	SEMCCO 2014 Conference Proceedings, Swarm, Evolutionary, and Memetic Computing

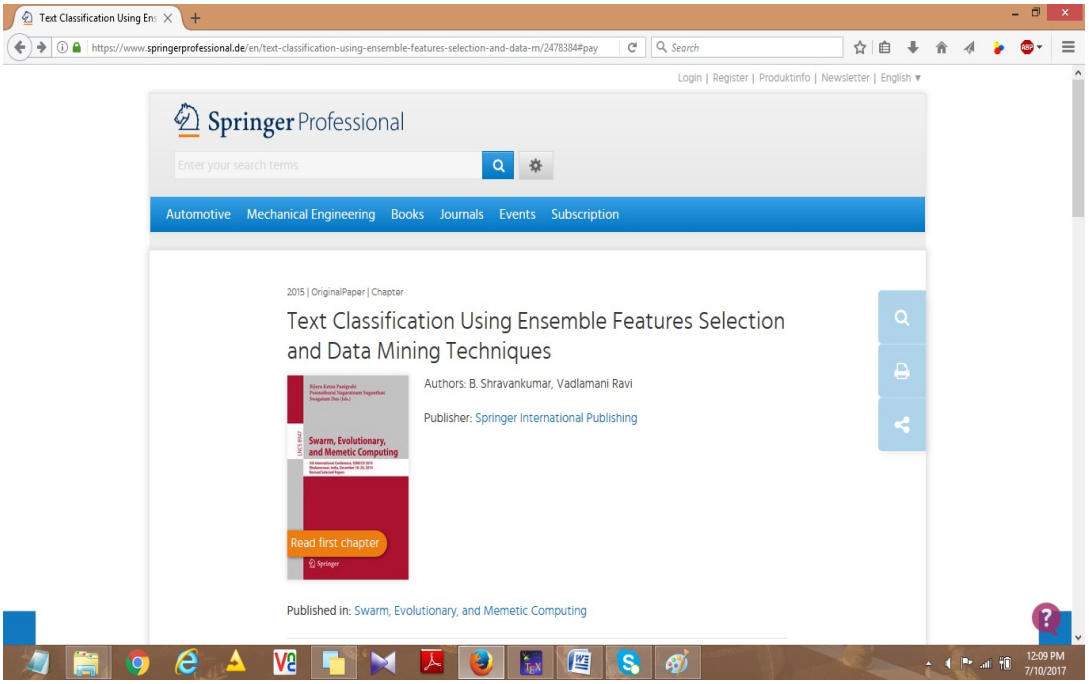

Figure B.1: Screenshot of SEMCCO-2014 Publication [1]

Table B.2: Fact Sheet of FICTA-2016 Publication [3]

Title	Text Document Classification with PCA and One-Class SVM
Authors	B. Shravan Kumar and Vadlamani Ravi
Publication	In poceedings of 5^{th} International Conference, FICTA 2016, Bhubaneswar, India, December 16-17, 2016, LNCS 515, pp. 107-115, 2017
DOI	10.1007/978-981-10-3153-3_11
Status	Published
Publisher	Springer International Publishing
Publication Type	FICTA 2016 Conference Proceedings, Frontiers in Intelligent Computing: Theory and Applications

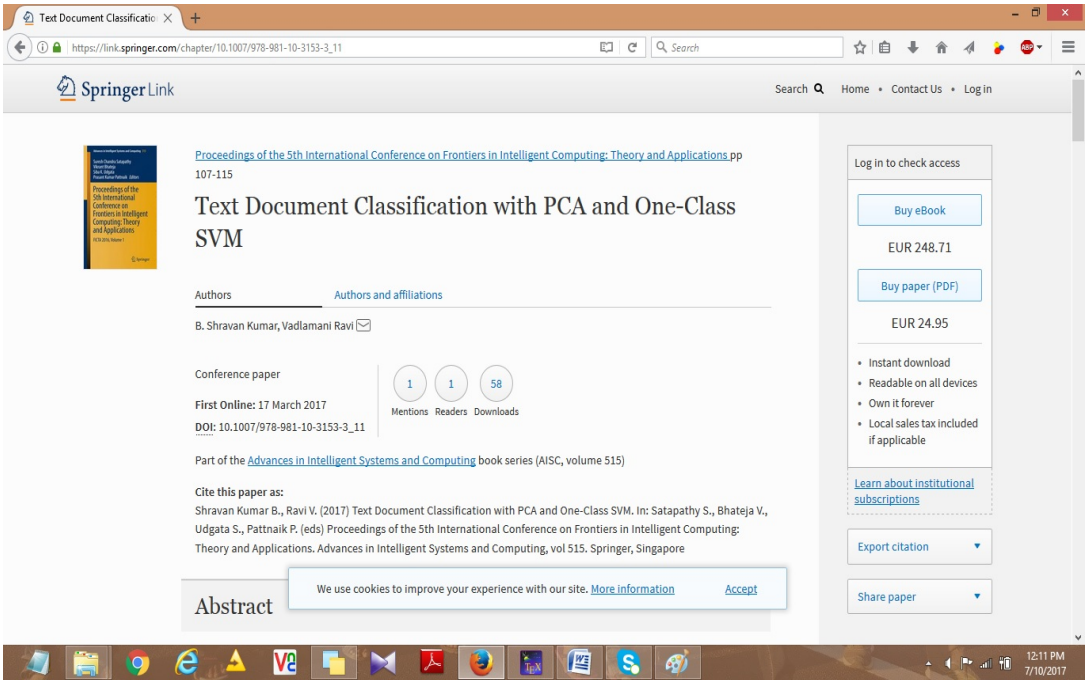

Figure B.2: Screenshot of FICTA-2016 Publication [3]

Table B.3: Fact Sheet of Knowledge-Based Systems Publication [4]

Title	A survey of the applications of text mining in financial domain
Authors	B. Shravan Kumar and Vadlamani Ravi
Publication	Knowledge-Based Systems, December 25, Vol. 114, pp. 128-147, 2016
DOI	10.1016/j.knosys.2016.10.003
Status	Published
Publisher	Elsevier
Publication Type	Journal, Knowledge-Based Systems

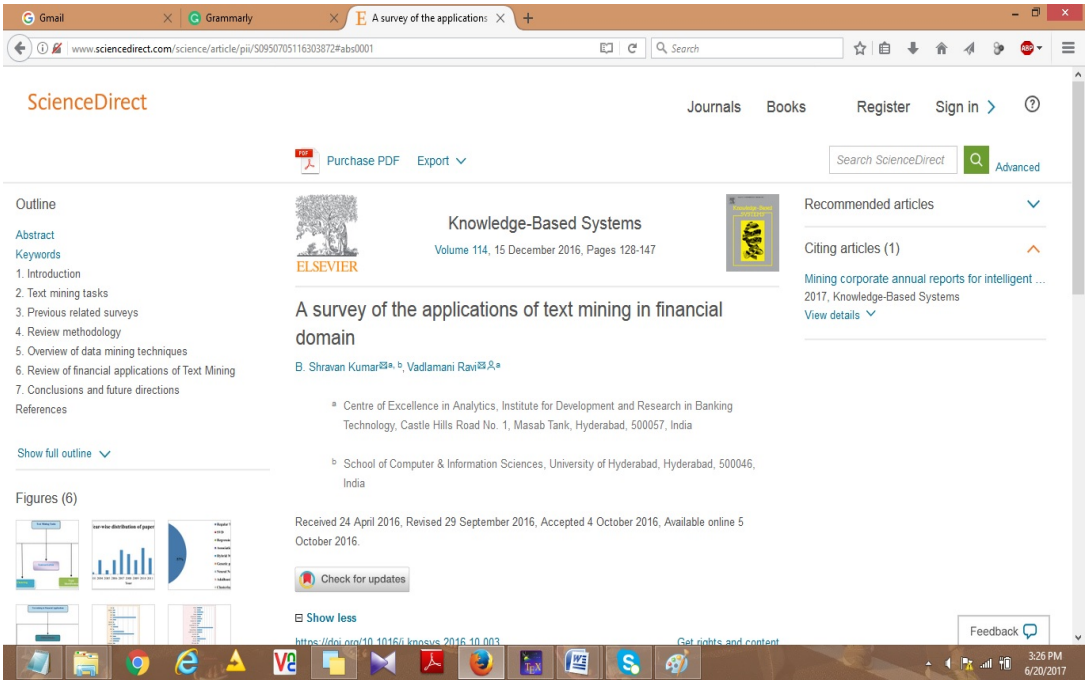

Figure B.3: Screenshot of Knowledge-Based Systems Publication [4]

Table B.4: Fact Sheet of ICAIECES-2016 Publication [2]

Title	One-Class Text Document Classification with OCSVM and LSI
Authors	B. Shravan Kumar and Vadlamani Ravi
Publication	In the proceedings of 2^{nd} International Conference, AIECES 2016, Chennai, India, May 19-21, 2016, LNCS 517, pp. 597-606, 2016
DOI	10.1007/978-981-10-3174-8_50
Status	Published
Publisher	Springer International Publishing
Publication Type	ICAIECES 2016 Conference Proceedings, Artificial Intelligence and Evolutionary Computations in Engineering Systems

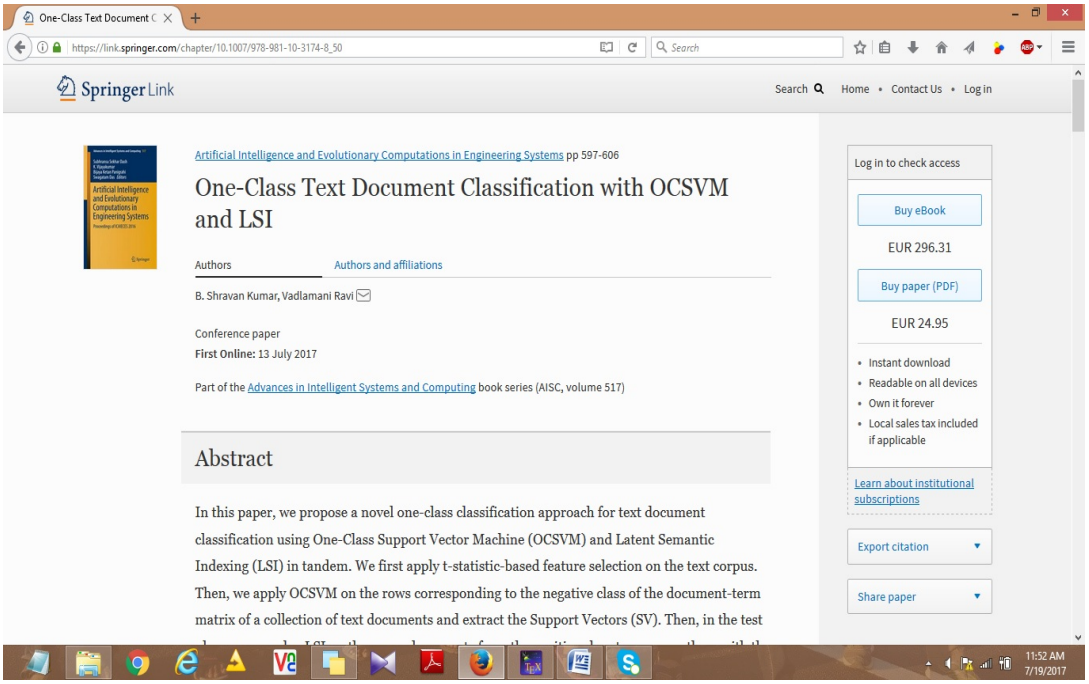

Figure B.4: Screenshot of ICAIECES-2016 Publication [2]

Table B.5: Fact Sheet of COMPUTE-2017 Publication [5]

Title	LDA based feature selection for document clustering
Authors	B. Shravan Kumar and Vadlamani Ravi
Publication	In the proceedings of 10^{th} Annual ACM India Conference, COMPUTE 2017, Bhopal, India, Nov 16-17, 2017, ACM.
DOI	10.1145/3140107.3140129
Status	Published
Publisher	ACM
Publication Type	ACM COMPUTE 2017 Conference Proceedings

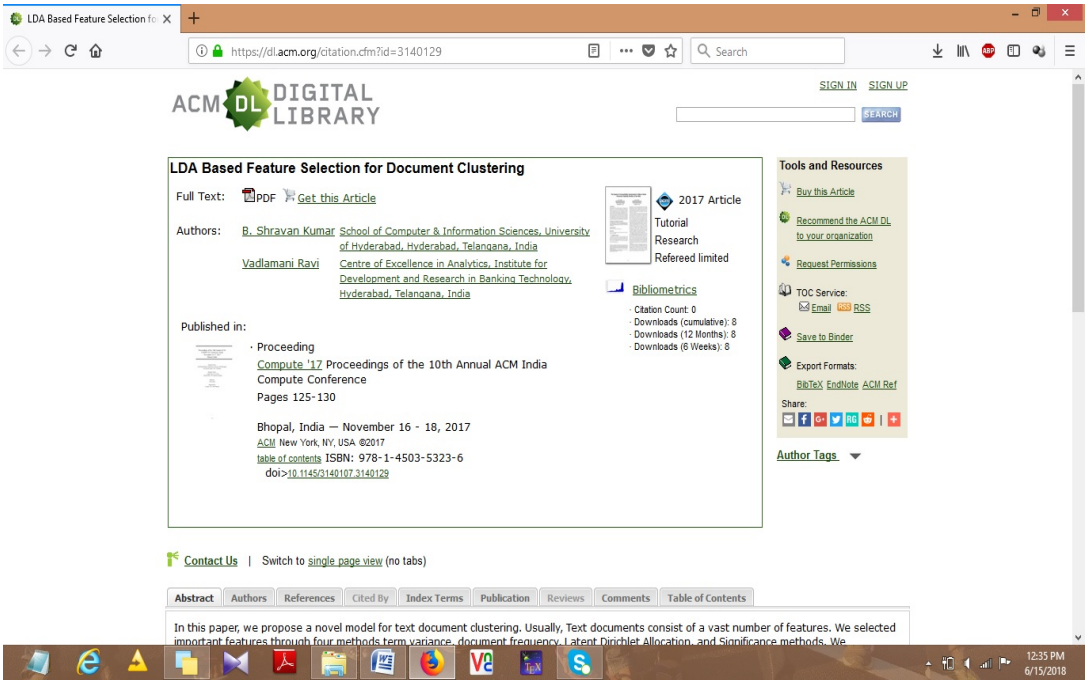

Figure B.5: Screenshot of COMPUTE-2017 Publication [5]

Table B.6: Fact Sheet of ICCI*CC-2018 Publication [6]

Title	A Hybrid approach using topic modeling and class-association rule mining for text classification: The case of malware detection.
Authors	B. Shravan Kumar and Vadlamani Ravi
Publication	In the proceedings of 17^{th} IEEE International Conference on Cognitive Informatics & Cognitive Computing, ICCI*CC, UC Berkeley, USA, July 16-18, 2018, IEEE.
Status	Accepted
Publisher	IEEE
Publication Type	IEEE ICCI*CC 2018 Conference Proceedings