

DESIGN AND IMPLEMENTATION OF EVENT DRIVEN NEWS CLASSIFICATION TECHNIQUE FOR BANKING STOCKS ANALYSIS

Thesis Submitted For the Award of the Degree of

DOCTOR OF PHILOSOPHY

in

Computer Applications

By

Varun Dogra

41800180

Supervised By
Dr. Aman Singh

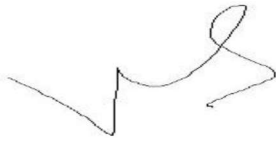
Co-Supervised by
Dr. Sahil Verma



LOVELY PROFESSIONAL UNIVERSITY
PUNJAB
2021

DECLARATION

I hereby certify that the work that is being presented in the thesis entitled “Design and implementation of event driven news classification technique for Banking stocks analysis” submitted to the Department of Research at Lovely Professional University, Phagwara is an authentic record of my work carried out during a period from August 2018 to Dec 2021 under the supervision of Dr. Aman Singh, Associate Professor, Division of Research and Development, Lovely Professional University and co-supervision of Dr. Sahil Verma, Associate Professor, Chandigarh University. The matter presented in this thesis has not been submitted to any other University/Institute for the award of a PhD degree.



Signature of the Candidate:

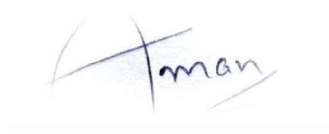
Varun Dogra

Date: 21/12/2021

Place: Phagwara

CERTIFICATE

This is to certify that the thesis entitled “Design and implementation of event driven news classification technique for Banking stocks analysis”, which is being submitted by Mr. Varun Dogra for the award of the degree of Doctor of Philosophy in Computer Applications from the Faculty of Technology and Sciences, Lovely Professional University, Punjab, India, is entirely based on the work carried out by him under our supervision and guidance. The work reported embodies the original work of the candidate and has not been submitted to any other university or institution for the award of any degree or diploma, according to the best of my knowledge.



Dr. Aman Singh

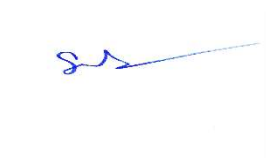
Associate Professor

Division of Research and Development

Lovely Professional University,

Phagwara, Punjab-144411

(India)



Dr. Sahil Verma

Associate Professor

Chandigarh University,

Mohali, Punjab- 140413

(India)

Acknowledgments

First and foremost, I want to thank God for the successful completion of my research work and thesis.

The present work was done due to the support and cooperation of many people. I would like to convey my gratitude to my supervisor, Dr. Aman Singh, and co-supervisor, Dr. Sahil Verma, for their direction, encouragement, and support throughout my research. They guided me in carrying out the research and completing the work in many ways, explaining when necessary, thanks to their enthusiasm, motivation, and encouragement. I would also want to express my gratitude for their efforts, which inspired me to widen my research to include a variety of views. It would be difficult to accomplish this study without their foresight.

I would not have been able to complete the thesis without the help of my parents, brothers, wife, and children. I admire their moral support and concern during this research work. I also want to convey my gratitude to my colleagues and friends for their constant encouragement and support.

As always, it is impossible to thank everyone who has had a positive direct and indirect influence on my work; nonetheless, there are others whose support is much more significant, and I express my gratitude to all of them.

Varun Dogra

List of Figures

Chapter Number	Figure	Page
1.1	Movement of the Stock price of Canara Bank (NSE) for month-wise 2019	3
1.2.	Sub-tasks of the Text Classification process	7
1.3.	The distribution of news article instances amongst 4-classes	9
2.1.	Criteria for feature selection for dimensionality reduction	28
3.1.	Multiclass classification of Financial News	61
3.2.	Steps for News-Events representation and classification using a hybrid approach of Rule-based and machine learning model based on DistilBERT	67
3.3.	Several pre-trained models with parameters count	69
4.1.(a)	Performance metrics P, R, F-1 for various classifiers for Banking Class	77
4.1.(b)	Performance metrics P, R, F-1 for various classifiers for Global Class	77
4.1.(c)	Performance metrics P, R, F-1 for various classifiers for Non-Banking Class	77
4.1.(d)	Performance metrics P, R, F-1 for various classifiers for Governmental Class	78
4.1.(e)	Accuracy for various classifiers with balanced classes using Up-Sampling	79
4.1.(f)	Accuracy for various classifiers with balanced classes using Down-Sampling	80
4.1.(g)	Accuracy for various classifiers with balanced classes using SMOTE Up-Sampling	82

4.1.(h)	Accuracy for various classifiers with balanced classes using ADASYN Up-Sampling	83
4.1.(i)	Accuracy for various classifiers with balanced classes using Near-Miss Down-Sampling	84
4.1.(j)	Comparison of accuracies with Classifiers across different approaches	86
4.1.(k)	Comparison of Precision with Classifiers under each class across different approaches	86
4.1.(l)	Comparison of Recall with Classifiers under each class across different approaches	87
4.1.(m)	Ensemble classifiers vs Random Forest (SMOTE)	87
4.1.(n)	Precision of ensemble classifiers vs Random Forest (SMOTE)	87
4.1.(o)	Recall of ensemble classifiers vs Random Forest (SMOTE)	88

List of Tables

Chapter Number	Table	Page
2.1.(a)	Benefits and limitations of Text Representation or feature Extraction method	44
2.1.(b)	Benefits and limitations of feature selection methods	46
2.1.(c)	Benefits and limitations of the machine and deep learning model	48
3.1.	Sample of news articles from different sources and classes	63
3.2.	List of Events labels and Attributes	71
4.1.(a)	Results for the classifiers for different classes with data imbalances	76
4.1.(b)	Results for the classifiers for different classes with balanced data using Up-sampling	76
4.1.(c)	Accuracy of the classifiers with imbalanced data	76
4.1.(d)	Accuracy of the classifiers with balanced data using Up-sampling	78
4.1.(e)	Accuracy of the classifiers with balanced data using Down-sampling	79
4.1.(f)	Results for the classifiers for different classes with balanced data using Down-sampling	80
4.1.(g)	Results for the classifiers for different classes with balanced data using SMOTE	80
4.1.(h)	Accuracy of the classifiers with balanced data using SMOTE Up-Sampling	81
4.1.(i)	Results for the classifiers for different classes with balanced data using ADASYN	81
4.1.(j)	Results for the classifiers for different classes with balanced data using Near-Miss	81

4.1.(k)	Accuracy of the classifiers with balanced data using SMOTE ADASYN-Sampling	82
4.1.(l)	Accuracy of the classifiers with balanced data using Near-Miss Down-Sampling	84
4.1.(m)	Results for the Ensemble classifiers for different classes with balanced data	85
4.1.(n)	Accuracy for the ensemble classifiers	85
4.2.(a)	List of Events labels and news samples for classification	88
4.2.(b)	Results of the machine learning classifiers fine-tuned with DistilBERT	91
4.2.(c)	Accuracy of the classifiers with DistilBERT	91
4.2.(d)	Results of the machine learning classifiers with TFIDF	92
4.2.(e)	Accuracy of the classifiers with TFIDF	93
4.2.(f)	Results of the DistilBERT fine-tuned with Random Forest classifier and proposed Hybrid model	93
4.2.(g)	Accuracy of the DistilBERT fine-tuned with Random Forest classifier and proposed Hybrid model	94
4.3.(a)	Results of the DistilBERT fine-tuned on banking news-events sentiments with machine learning classifiers	95
4.3.(b)	Accuracy of the classifiers with DistilBERT	95
4.3.(c)	Results of the machine learning classifiers with TF-IDF	96
4.3.(d)	Accuracy of the classifiers with TF-IDF	96
4.4.(a)	The impact of banking news events with different polarities or tones of communication on Bank stock or Bank Nifty returns	97
4.4.(b)	The impact of Negative polarity or tone of online banking news events publication on Bank stock or Bank Nifty returns	99

4.4.(c)	The impact of Positive polarity or tone of online banking news events publication on Bank stock or Bank Nifty returns	100
4.4.(d)	The impact of Neutral polarity or tone of online banking news events publication on Bank stock or Bank Nifty returns	102

Contents

Abstract	vi
Acknowledgments	vii
Contents	viii
List of Figures	xiii
List of Tables	xv
Chapter 1 Introduction	1
1.1.Overview	1
1.2.Text Classification	5
1.3.Text Classification Framework	7
1.3.1. Multiclass Classification	10
1.3.2. Learning from Imbalanced Dataset	10
1.4.Sentiments in News Events	12
1.5.Event Study	13
1.6.Problem statement and Motivation	13
1.7.Research Objectives	15
1.8.Research Contribution	16
1.9.Thesis Organization	16
Chapter 2 Literature Review	18
2.1.Event extraction and text classification	18
2.1.1. Rule-based approaches	19
2.1.2. Data-driven approaches	19
2.1.3. Text Document Representation: Features construction and weighting	20

2.1.3.1.Context-independent approaches	20
2.1.3.2.Context-aware approaches	23
2.1.4. Data Preprocessing: Data Cleaning	26
2.1.5. Data Preprocessing: Dimensionality Reduction	27
2.1.5.1.Standard Feature Selection methods	27
2.1.5.2.Text Feature Extraction methods	34
2.1.6. Classifiers for Classification Task	35
2.1.7. Evaluation	43
2.1.8. Comparative Analysis	44
2.1.8.1.Comparative Analysis of Standard Text Representation or Feature Extraction methods	44
2.1.8.2.Comparative Analysis of Standard Feature Selection methods	45
2.1.8.3.Comparison of State-of-Art Machine and Deep Learning models for Text Classification	48
2.1.9. Hybrid approach	50
2.1.10. Techniques for dealing with Imbalanced data in Multi-class Text classification	51
2.1.10.1. Data level technique	51
2.1.10.2. Algorithms-based decomposition techniques	52
2.1.10.3. Algorithms-based Ensemble techniques	53
2.1.10.4. Existing solutions or software for classification with imbalanced datasets	56
2.2.Text Sentiments classification	56
2.3.Named Entities Recognition	57
2.4.Event Study	58
2.5.Summary	59

Chapter 3 Research Methodology	60
3.1.Banking News Extraction by Multiclass Classification with Imbalanced Datasets of Financial News: Challenges and Solutions	61
3.1.1. Data Pre-processing	61
3.1.1.1.Data Collection	63
3.1.1.2.Labeling	63
3.1.1.3.Data Cleaning	64
3.1.1.4.Data Transformation	64
3.1.2. Training Classifiers	65
3.1.3. Testing classifiers and their performance	66
3.2.Banking news-events classification with a novel hybrid model using DistilBERT and rule-based features	67
3.2.1. DistilBERT for banking news-events representation	68
3.2.2. Machine learning classifiers	69
3.2.3. Fine-tuning with rules	70
3.3.Sentiments classification	71
3.4.NER and analysis of banking stocks and the banking index in Indian Stock market based on past and current linguistic or sentiments of news events	72
Chapter 4 Results and Discussions	75
4.1.Banking News Extraction by Multiclass Classification with Imbalanced Datasets of Financial News	75
4.1.1. Results of multiclass classification with data-imbalances	75
4.1.2. Results of multiclass classification with data balance using Data-level technique: Random Over-Sampling with replacement	78

4.1.3. Results of multiclass classification with data balance using Data-level technique: Random Down-Sampling without replacement	79
4.1.4. Results of multiclass classification with data balance using Data-level technique: hybrid over-sampling technique SMOTE	80
4.1.5. Results of multiclass classification with data balance using Data-level technique: over-sampling technique ADASYN	82
4.1.6. Results of multiclass classification with data balances using Data-level technique: down-sampling technique Near-Miss	83
4.1.7. Results of multiclass classification with data balance using Ensemble classifiers	84
4.2. Banking news-events representation and classification with a novel hybrid model using DistilBERT and rule-based features	88
4.2.1. Data sets	88
4.2.2. Data transformation and classification	89
4.2.3. Applying Rules	90
4.2.4. Results of Banking News-Events classification with DistilBERT	91
4.2.5. Results of banking news-events classification with TFIDF	92
4.2.6. Results of banking news-events classification with a hybrid approach	93
4.3. Sentiment Classification of Banking Financial News	94
4.3.1. DistilBERT fine-tuned on banking news-events sentiments for classification with machine learning classifiers	94
4.3.2. Results of Banking News-Events sentiments classification using machine learning classifiers with TF-IDF	95
4.4. Named entities recognition and analysis of banking stocks and the banking index in Indian Stock market based on past and current linguistic or sentiments of news events	96
Chapter 5 Conclusion and future scope	104
5.1. Conclusions	104

5.2.Future scope	106
References	107
Research Publications	127

Abstract

Machine learning has grown in popularity in recent years as a method for evaluating financial text data, with promising results in stock price projection from financial news. Various research has looked at the relationship between news events and stock prices, but there is little evidence on how different sentiments of such events impact investor behavior. The main objective of this study is to analyze how a specific banking news event (such as a fraud or a bank merger) and other co-related news events (such as government policies or national elections), as well as the framing of both the news event and news-events sentiment, impact the formation of the respective bank's stock and the banking index i.e. Bank Nifty in Indian stock markets over time. The task is achieved through four phases.

In phase-1, we have performed extensive experiments to classify the financial news articles into the predefined four classes Banking, Non-Banking, Governmental, and Global. The idea of multi-class classification was to extract the Banking news and its most correlated news articles from the pool of financial news articles scraped from various web news portals. The creation of a framework for storing news articles and collecting information for specific domains is an important and untested problem for the Indian stock market. When online news portals produce financial news articles about many subjects simultaneously, finding news articles that are important to the specific domain is nontrivial. A critical component of the aforementioned system should, therefore, include one module for extracting and storing news articles, and another module for classifying these text documents into a specific domain(s).

The news articles divided into the mentioned classes were imbalanced. Imbalance data is a big difficulty with most classifier learning algorithms. However, as recent works suggest, class imbalances are not in themselves a problem, and degradation in performance is often correlated with certain variables relevant to data distribution, such as the existence in noisy and ambiguous instances in the adjacent class boundaries. A variety of solutions to addressing data imbalances have been proposed recently, over-sampling, down-sampling, and ensemble approach. We have presented the various challenges that occur with data imbalances in multiclass classification and solutions in dealing with these challenges. The study has also shown a comparison of the performances of various machine learning models with imbalanced data and data balances using sampling and ensemble techniques. From the result, it's clear that the performance of Random Forest classifier with data balances using the over-sampling technique SMOTE is best in terms of precision, recall, F-1, and accuracy. From the ensemble

classifiers, the Balanced Bagging classifier has shown similar results as the Random Forest classifier with SMOTE. The Random Forest classifier's accuracy, however, was 100% and it was 99% with the Balanced Bagging classifier.

As mentioned above, every day, hundreds of events are published as articles on web news portals or corporate websites, necessitating the need to automatically categorize and comprehend them. Natural Language Processing has been studied in a variety of methods especially with the rise of machine learning algorithms and their ability to solve problems. Event Extraction and Classification, a method of gathering data on recurring events included in documents and automatically extracting information about what happened and when is one of its most prevalent uses. However, due to the unstructured nature of the original data, data transformation is mind-numbing.

In phase-2, The study introduces a unique hybrid method to text classification that combines a machine learning algorithm with DistilBERT, a pre-trained deep learning framework for natural language processing, and provides a basic model that is fine-tuned on Indian Banking News- Filtering false positives and dealing with false negatives are used to improve the results provided by the previous classifier in events with a rule-based approach. The main advantage is that the system may be swiftly fine-tuned by inserting unique rules for particular chaotic or overlapping categories that have not been adequately trained. This study also compares the performance of DistilBERT, a state-of-the-art deep contextual language representation employed in our proposed hybrid model, with TFIDF, the most popular context-independent language representation, on supervised learning of multiclass Banking News-Events classification. Logistic Regression, Linear SVC, Decision Tree, and Random Forest are all machine learning classifiers that use both representations. In comparison to the baseline TFIDF, DistilBERT can better transfer general domain knowledge to other domains, resulting in improved accuracy using Random Forest. As per our proposed approach, the outcome is then passed into the Rules-based technique. Furthermore, our proposed hybrid model enhanced accuracy.

In phase-3, the sentiment classification methodologies are introduced in the areas of Indian banking, government, and global news. On multiclass (positive, negative, and neutral) sentiment classification of news events, the study compares the performance of DistilBERT, a state-of-the-art deep contextual language representation, with TF-IDF, a classic context-independent technique. The DistilBERT model is fine-tuned and fed through four supervised

machine learning classifiers: Random Forest, Decision Tree, Logistic Regression, and Linear SVC, as well as the TF-IDF baseline. The results show that DistilBERT can transfer fundamental semantic knowledge to more domains and achieve higher accuracy than the TF-IDF baseline. In comparison to other ML classifiers, the findings show that Random Forest with DistilBERT has greater accuracy. The accuracy of the Random Forest with DistilBERT is 78%, which is 7% higher than the TF-IDF.

In phase-4, we analyze the impact of news events concerning sentiments or linguistics in the price movement of the respective bank's stock, identified or recognized from these news events, against benchmark index Bank Nifty and the banking index against benchmark index Nifty50 for the short to long term. We explore the deep learning model, bidirectional encoder representations from transformers, and statistical method CAPM for this research. News events or online news pieces may help to develop investor sentiment, online media has a significant impact on financial markets. This study looks at the links between online news events and stock market reversals, focusing on the impact of a few banking and government news events reported in major online news portals between 2018 and 2021 on aberrant stock returns. The research considers how the sort of news event and the polarity or tone of communication may affect the performance of banking stocks. The findings show that the impact of unfavorable financial news items lasted for one month after they were published. The symmetric event window with the significant mean CAR of -5.13% is presented (-30,30). It also shows that the stock market may anticipate bad quarterly or yearly results from banking corporations, which might affect profits in the short term. Significant outcomes before the event date are seen as proof that stock markets may forecast unfavorable sentiment news occurrences. We also found evidence that the negative tone or sentiments of banking news events cause banking stocks or the banking index to respond negatively. Our results have some significance. To begin with, overreactions to unfavorable news in the banking indexes indicate that the market is concerned about excellent and effective government and banking policies. News articles with a favorable or even neutral tone or sentiment on the events 'RBI Policies' and 'Governmental' had a statistically significant positive influence on banking stocks or banking indices, with CARs of 10.32% and 3.79%, respectively, in the event window (D,60). According to the report, investors dislike merger-related news for even prospering banks, therefore they opt to sell their stocks.

Chapter-1

Introduction

1.1.Overview

Since the emergence of Web technology, Internet-based news media has become a significant medium for discovering and sharing information. Moreover, the surveillance of such online text data has of great importance in many domains such as the stock market [1],[2],[3],[4],[5] healthcare [6], politics [7], education [8], tourism [9] and e-Commerce [10]. Because of the quick and widespread transmission method, it is a potentially useful source of information about companies' underlying values especially in the context of financial markets. The impact of Internet-based news articles on stock prices has been studied by several researchers. The stock markets are influenced by a variety of news events, including domestic and international news. Some announcements or news are company-specific or influence a specific sector. In the Indian stock market, the news on RBI policies and loan frauds can impact particularly the stocks of the banking sector.

The banking sector accounts for 7%-8% of India's overall GDP. Since the liberalization of the economy in 1991, the Indian banking sector has experienced significant growth and transformation. Even though the banking sector is well-regulated, it faces several challenges, including economic difficulties and a lack of ethical norms. Recent studies show that financial crises harm growth and that assessing and measuring bank risk on a regular and timely basis is critical to preserving the banking system's stability and profitability [11],[12]. Another study mentions that Fraud has grown significantly in number and value during the previous 10 years, according to the Reserve Bank of India's annual report (2019) and the Economic Times' article (2019) [13]. It also mentions that during 2009–2010, banks reported 4669 frauds worth 1998.94 crores, which increased to 6,801 frauds worth 71542.93 crores by the end of 2018–2019. From 2009–2010 to 2018–2019, the number of incidents of fraud increased by 45.66%, with the sum implicated increasing by 35 times or more. Bank fraud has the potential to have severe economic and social effects. To measure the bank risks, one of the most often used indicators is the return volatility of banking stocks. These reasons have attracted us to study banking events and their impacts. Our study focuses on understanding and analyzing the impact of such events on the valuation of banks through the observations of banking stocks listed in Indian stock exchanges NSE and BSE.

Economists have conducted a significant number of researches to determine the correlations between news events and stock returns. Banking Stocks can exhibit aberrant behavior as a reaction to socio-political events, bank risks, and other news events, according to economists. There has been no systematic effort to automate the tracking of such events and utilize them to analyze banking stocks. There have been individual initiatives to analyze Twitter sentiments or various banking activities, but to our knowledge, no integrated effort to examine the influence of specific categories of news events on banking stocks movements exists. This might be owing to them being unable to compile a database of classifying several potentially important news events from the overall financial news articles that will have an influence on banking stocks in the immediate future.

Every day, hundreds of events are presented as articles on web news portals or corporate websites, necessitating the need to automatically arrange and interpret them. Natural Language Processing has been studied in several ways as machine learning algorithms and their problem-solving capabilities have advanced. Event Extraction and Classification, a method of gathering data on recurring events included in documents and automatically extracting information about what happened and when is one of its most popular uses. However, because of the unstructured nature of the original data, data transformation is mind-numbing.

Extraction of events from large datasets, like news items, might be beneficial to IE systems in a variety of ways. Researchers have suggested that the events might be beneficial in monitoring system applications [14]. The authors [15] looked at risk analysis in context with events, they [16] demonstrated the influence of US news events on the German stock market. Financial markets react to news in a very nuanced way. Economic events like mergers and acquisitions, share splits, bonus declarations, and so on have a significant impact on broker daily decisions, whether the brokers are humans or machines. The writers looked into the impact of obtaining a company's news on stock prices [17]. Computers can manage greater dimensions of important news, have better information than people, and make better up-to-date choices, in addition to being able to interpret news more quickly.

Automatic financial news monitoring entails the constant monitoring of information sources connected to banking and non-banking activities, as well as the prompt disclosure of contradictory events. The system's goal is to keep users up to date on banking news events so they can take precautions based on the nature of the occurrence. Events are occasionally referenced explicitly in news articles; however, some news publications do not include

information directly stating the event's name. It is critical to develop a system that can comprehend published news items, extract events, and classify them. The researchers developed a FIDS system that can automatically categorize financial news into distinct categories based on content and extract information [18].



Fig 1.1. Movement of the Stock price of Canara Bank (NSE) for month-wise 2019

Given the positive outlook for event extraction applications and the belief that the tasks of real-time extraction and event organization can be effectively addressed, it's worth looking into whether text mining techniques are appropriate for this purpose. The extraction of data on financial transactions, such as mergers and acquisitions, stock splits, dividend announcements, and other abnormalities represented in terminology related to common (complicated) money and risk words, is critical. This can also be linked to participants, times, and places. News has a huge impact on the financial markets [19]. For example, as mentioned in Fig. 1.1, in 2019, the stock price of Canara Bank on the NSE ranged between INR 200 and INR 300, owing in large part to the impact of financial events involving Canara Bank that occurred during the same fiscal year. Stock values are prone to large price fluctuations, which are usually accompanied by big financial events.

Canara Bank announced its Q4 results in May 2019, with a reduced deficit for the fiscal year 2018-19. In a month, the stock has risen from 280 INR to 300 INR. During the same period, the Indian government announced a proposal to inject capital into public sector banks. It had improved investor sentiment. The PSU banks, on the other hand, took a big hit once the Union budget was released around the same time. In addition, the news began targeting PSU banks on non-performing loans, which peaked in FY20. Canara Bank's shares have plummeted to INR 210 in September 2019. Canara Bank reducing 30% of its share in Can Fin Homes with a premium was reported at the start of December 09, 2019. In December 2019, the Indian government also addressed the problem of economic instability. Canara Bank was given a buy signal by the expert in mid-December 2019, and the stock has since recovered to 230 INR levels on December 23, 2019. This example demonstrates how events contained in news messages may be useful in financial applications. This may be applied to applications like stock price prediction and algorithmic trading. The use of computer programs to combine commercial transactions using algorithms that decide on parameters like time, size, and the order amount is known as algorithmic trading.

Text classification is a related aspect of information extraction in which content is allocated to one of several classes based on its content. This topic has been thoroughly researched, and different strategies, including machine learning approaches, have been investigated in the literature by [20] Naïve Bayes, [21] based on the selection of local features, [22] offered classification based on the selection of feature relation, [23] offered technique based on Deep Learning. In certain cases, information extraction is considered as a particular type of text classification in which the number of categories is limited to two. The text comprehension technique, on the other hand, takes on a more difficult task by attempting to summarize a document by extracting just the important information. It varies from text summarization and content filtering in a significant way. To find the desired content, a more complete examination of the document is necessary, which will almost certainly include the employment of natural language processing tools. It makes the work more difficult.

The ability to comprehend event descriptions is largely reliant on context. To produce accurate estimates, a design must take into consideration the mentions of events by the sense of discourse. For example, the following headline from financialexpress.com:

“FE Online | February 16 2018 4:43 PM HDFC Securities is bullish on the shares of Max Financial Services after the company reported strong **Q3 results**. Last week the company reported a 10.3 percent jump in its **Q3 net profit** at Rs. 88.8 crore against Rs. 80.5 crore in the same period fiscal.”

The occurrence of the company's quarterly Q3 results is mentioned in the news title. It represents the entity Max Financial Services, which has strong results, and HDFC Securities' perspective on this stock, as well as the company's net profit, which is made up of event-related data in the first line.

1.2. Text Classification

In the equity market, stocks or funds belong to different business sectors. And sector-based news has become an inseparable part of the management of financial assets, with news-driven stock and bond markets explosively growing. Fund managers take advantage of this reality and make use of sector-oriented news to select individual stocks to diversify their investment portfolios to optimize returns. As we mentioned earlier that there is no such structured framework available for classifying the news on specific sectors of someone's interest. This problem is increasing by the day, necessitating a system for news classification methodology for specific sectors. Machine learning (ML) techniques have demonstrated impressive performance in the resolution of real-life classification problems.

However, the vast majority of articles on the internet about text classification are binary text classification [24] such as email filtering [25], political preferences [26], sentiment analysis [27], etc. Our real-world problem is in most cases much more complex than the binary classification. More formally, if some d refers to a document in a whole set of documents D and C refers to a set of categories i.e. $C = \{c_1, c_2, c_3, \dots, c_n\}$ the classification of text assigns one category c_i to the document d . Such a classification function with more than two classes is known as multiclass classification; for example, identify a set of news categories like business, political, economic, or entertainment. Multi-class classification presupposes that each sample is allocated to a single label: a news article may be a business or a political, but not both at the same time.

Indeed, the majority of the digital facts are available as text documents and which are usually unstructured or semi-structured [28]. Thus, to make data for decision making, structuring this textual data became a necessity [29],[30]. However, because of a high volume of data, it's quite impossible to process the data manually. Text classification has evolved due to this challenge. It is defined as assigning the text documents to one or more categories (called labels) according to their content and semantics. Traditionally, majority classification tasks

were used to solve manually but it was expensive to scale. Classification can be thought of as writing rules for assigning a class to similar text documents. These rules include some related information that identifies a class. Handwritten rules can be performed well but creating and maintaining over time requires much manpower. A technical expert can frame rules by writing regular expressions that could maximize the accuracy of the classifier. The existing studies have proposed various techniques to automatically classify text documents using machine learning [31],[32]. In this approach, the set of rules or criteria for selecting a classifier is learned automatically from the training data. Under each class, it requires a lot of training documents and expertise to label the documents. The labeling is a process of assigning each document to its associated class. Researchers, in their studies, mentioned that the labeling process was easier than writing hand-crafted rules. They further presented variously supervised and semi-supervised learning approaches to diminish the burden of manual labeling [33], [34]. Automated text-classification methods may be divided into three groups; **Rule-based** methods, **Data-driven** methods, and **Hybrid** methods.

Using a set of pre-defined rules, rule-based methods classify text into several categories. For example, the ‘fruit’ label is applied to any document with the words ‘apple’, ‘grapes’, or ‘orange’. These techniques need a thorough understanding of the domain, and it is challenging to retain the systems. Data-driven methods, on the other hand, learn to do classifications based on preceding data values. A machine-learning algorithm may understand the underlying relationships between texts and their labels by using pre-labeled samples as training data. It can discover hidden patterns in the data, is further flexible, and may be used for different tasks. As the title indicates, hybrid approaches use a mixture of machine learning methods (data-driven) and designed rules for making likelihoods.

In recent decades, machine learning models have attracted much interest. A two-step method is used in most traditional machine learning models, in which specific features are retrieved from text documents in the first phase, and then supplied to a classifier in the second step to generate a prediction. The popular features representation models are TF-IDF (Term Frequency-Inverse Document Frequency), BOW (Bag-of-words), etc. And the common classifiers are Naïve Bayes, SVM, KNN, Decision Trees, Random forests, etc. These models are discussed in detail in the next sections. Since the last decade, deep learning models have also been used in a wide range of NLP applications, thus enhancing the state of the art [35], [36], [37]. These models are attempting, in an end-to-end fashion, to learn the feature representations and perform classification. Not only do they have the ability to reveal hidden

patterns in data, but they are also considerably more reusable from one project to the next. Quite significantly, in recent years these models have become the mainstream paradigm for the various tasks of text classification. Each sub-task of text classification is shown in Fig 1.2.

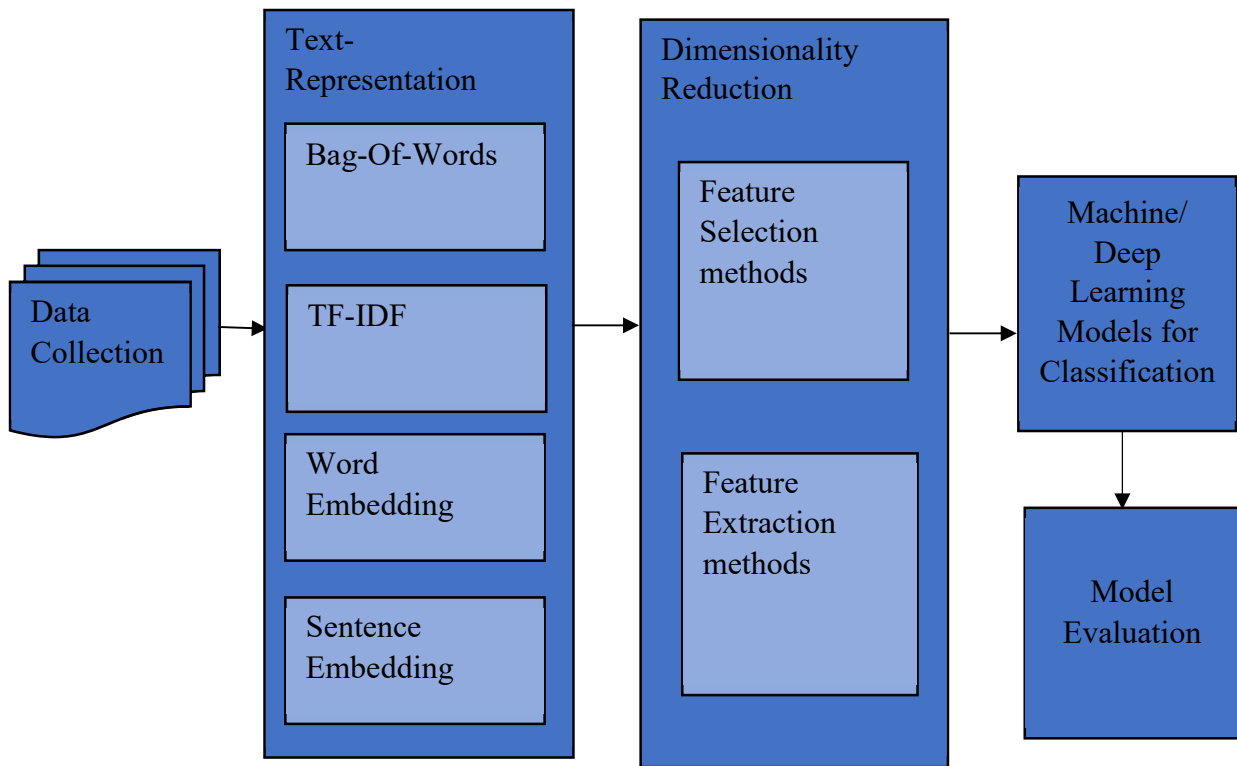


Fig. 1.2: Sub-tasks of the Text Classification process

1.3.Text Classification Framework

Text classification is a learning process in which a classifier is trained to distinguish between preset classes based on characteristics extracted from a collection of text documents [38]. The accuracy of the classifier depends upon the classification granularity and how well separated are the training documents amongst classes [39],[40]. In-text classification a set of labels or classes are given, we need to evaluate which class/label a particular text document relates to. Usually, a class or label is a general topic like Sports or Business. But it may be significantly more difficult to distinguish between documents that are about more similar classes such as Networks and Internet of Things. Certain features represent the potential overlap between classes; the learning task would be simplified by removing such overlapping features. If the gap between classes could be increased, the classification performance would increase. This can be achieved through features weighting and selecting valuable features. Many researchers have investigated and used text classification in real-world situations, such as sentiment categorization of stock market news and its influence [41], news classification for syndromic

surveillance [2], microblog topic classification [42], sentiment classification using domain adaptation [43] and brand promotion based on social media sentiments [44].

A text classification process is described as classifying a set N of documents; first, we build a classifier T . There is a collection of text documents D , and every text document is given a class/label by an expert. Secondly, we need to train a classifier for each class/label by giving input as a corresponding set of documents in D . Now we need to apply trained classifier C to classify N documents. We will get each document in N assigned to a predefined class/label by C . The process of text classification includes phases or sub-tasks. The process starts from the collection of textual content from various sources. The textual content may be belonging to a domain(s) representing some events, business processes, or public information. Then these text documents require pre-processing to generate appropriate text representation for the learning model. This is done in two phases, in phase-1 a feature extraction algorithm is used to extract features from the processed text, and in phase-2, the features are reduced by applying feature selection techniques. This reduction of features tends to decrease the dimensions of data required for the learning method. After these phases, the learning algorithms are chosen to train on data to generate the best classifier for recognizing a target category or class. This text data required to train a classifier is known as training data. The data is divided into two sets; the majority of data is taken for the training model and the rest part of the data is taken for testing the classifier, known as testing data. Similarly, the model is trained to recognize each target class representing its data available in the associated text documents. During the testing phase, when a classification method is developed, it is executed on test data to define the target class of input text and the result is produced in the form of weights or probabilities. Finally, the result is evaluated for its accuracy, of text classifier, using evaluation techniques.

In our study, we're interested in isolating news on the banking sector and its most associated domains from the pool of financial news articles. We feel that 'banking news' of any nation is most correlated with their 'governmental news events' that covers news on government initiatives for good governance, state or national elections, change or new development of governmental policies, and 'global' financial news that covers global trade, changes in currency-commodities prices, and global sentiments. So, we have a 4-class classification problem of a set of news articles to extract banking, and its most correlated news i.e. Government, and Global from entire financial news articles. We decide to label the news articles into banking, governmental, global, and non-banking classes with a total of 10000

instances. The non-banking news covers all the financial news scrapped from various new portals divergent from these 3 categories (banking, governmental and global).

The news reports on different categories are usually imbalanced. The distribution of the news articles in our dataset is shown in Fig.3. The news articles are manually labeled into these four classes. [33] mentions that labeling is normally done manually by human experts (or users), which is a labor-intensive and time-consuming method but it results in higher accuracy due to expert knowledge being involved in labeling text articles with appropriate. In the process, we label a set of representative news articles for each class. The labelers are experts in the financial domain and financial markets. We perform feature selection to identify important or representative words for each class used in a 4-class classification, followed by inspecting each text document and labeling it to the respective class based on representative words for each class. The process is used to derive a set of documents from entire unlabeled documents for each class to form the initial training package. The different machine learning models are then used to build and compare the classifiers.

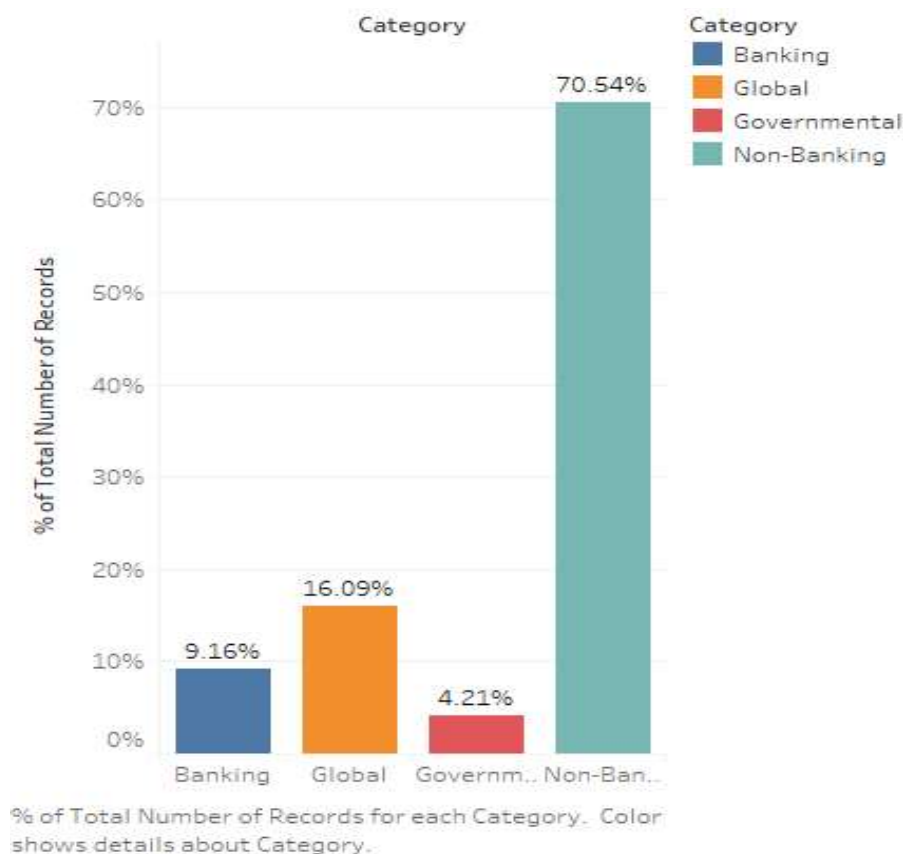


Fig.3: The distribution of news article instances amongst 4-classes

1.3.1. Multiclass Classification

For machine learning, the problem of classifying instances into three or more classes in multiclass classification. Although some classification algorithms of course allow the use of more than two classes, some are by definition binary algorithms; however, a variety of strategies may transform these into multi-classification. In a multiclass classification problem, only a few samples may be represented in some classes (the minority class), while the remainder belongs to the other class (called the majority class). The data disparity in machine learning creates difficulties in nearly all disciplines of real-world problems when it comes to data analytics. The problem of classifying textual news articles is a two-step procedure. In our experiment, in step-1, the documents are collected from various websites like Bloomberg, Financial Express, and Moneycontrol using web scrapping code written in Python. It is followed by partitioned news articles into their respective category of banking, non-banking, global, and governmental using manual labeling. In step-2, the news articles are trained and tested using machine learning approaches to achieve the classification goal for a new sample of news articles. A comparative analysis is performed based on the results of the experiment to rate the tested machine learning algorithms in descending order so they may be used to evaluate news classification tasks with imbalanced datasets.

1.3.2. Learning from Imbalanced Dataset

A dataset is considered class-imbalanced if the number of examples that represent each class is not equal. Dealing with an imbalanced dataset has been a popular subject in the research study of classifying news articles. The conventional machine learning algorithms may introduce biases while dealing with imbalanced datasets [45]. The accuracy of many classification algorithms is considered to suffer from imbalances in the data (i.e. when the distribution of the samples is significantly distorted across classes) [46]. Most binary text classification applications are of this kind, with the negative examples far outnumbering positive examples of the class of interest [47]. Many classifiers assume that examples are evenly distributed among classes and assume an equal cost of misclassification. For example, someone works in an organization and is asked to create a model that predicts whether news belongs to class A, based on the distribution of news in classes A and B at your side. He chooses to use his favorite classifier, train it on data, and before he knows it, he gets an accuracy of 95%. Without further testing, he wants to use the model. A couple of days later he underlines the model's uselessness. Indeed, from the time it was used to gather news, the model he created did not find any news belonging to class A. He figures out after some investigations that there

is only about 5 percent of the news produced in the pool that belongs to Class A and that the model always responds to Class "B," resulting in 95 percent accuracy. The kind of ‘guileless’ findings that he obtained were due to the imbalanced dataset with which he works. The goal of this section is to introduce the various methods that can be used with imbalanced groups to tackle classification problems.

In the imbalanced data set, basically with this problem, a classifier’s output leans to be partial towards certain classes (majority class) [48]. In Machine Learning and Natural Language Processing (NLP), the problems of imbalanced classification, under which the number of items in each class for a classification process differs extensively and the capacity to generalize on dissimilar data remained critical issues [49]. Most classification data set do not have precisely the same number of instances in each class but a slight variation is often insignificant. There are problems where class inequality is believed to not just normal.

Also, classifiers are typically built to optimize precision, which in the situation of imbalanced training data is not a reasonable metric for determining effectiveness. Therefore, we are presenting the comparison of various machine learning classification techniques which might result in high accuracy even with imbalanced datasets, however, it is worth mentioning certain challenges we find to deal with imbalanced data and evaluating certain measures along with accuracy to evaluate performance. Also, we conduct machine learning on documents to perform multi-classification, where the data sample belongs to one of the multiple categories exactly.

The key points are as follows:

- Imbalanced classification is the classification issue when the training dataset has an uneven distribution of the classes. As a result, appropriate sampling techniques must be implemented to balance the distribution by taking into consideration the various characteristics and the balanced performance of all of them.
- The class distribution imbalance may vary, but a serious imbalance is more difficult for modeling and may require advanced techniques. It is possible to introduce an efficient hybrid ensemble classifier architecture that incorporates density-based under-sampling or over-sampling and cost-effective techniques by examining state-of-the-art solutions using a “multi-objective” optimization algorithm.
- Most real-world classification problems, such as scam detection, news headlines

categorization, and churn prediction, have an imbalanced class distribution. In the case of class imbalances, several concerns should be considered while building multi-class classifiers.

1.4.Sentiments in the news events

News articles, as well as the sentiment expressed in such articles, include a lot of information that may or may not be properly represented in stock market statistics. The finance community has shown an increased interest in using linguistic and text analysis tools to analyze this data automatically in the prospects of introducing contextual information to existing financial models. Integrating approaches from two fields, text and econometric analysis, has received a lot of attention in recent years[50][51], with systems being created to automate the process and combine the techniques and data processing activities [52]. Developing a system that can take these massive quantities of news content and calculate a model that can use it, is an exciting problem. The task is about developing a technique and implementing it to do text analysis and statistical modeling, by designing a framework that may be generalized to facilitate the understanding of several text sources, financial and time-series data, and news categories.

Text sentiment analysis is a fast-growing field of natural language processing (NLP) that is currently broadly utilized in a variety of business applications, including algorithmic trading, social media analytics, consumer behavior, sales, and marketing. Such technologies have started to be utilized in economic and financial research in recent years. The study uses text sentiment analysis to determine the objective function of the central bank using meeting transcripts from the Federal Open Market Committee [53]. Another research study focuses on “lexical” methods that determine the sentiment of a piece of text based on the tone of the words within it. Such methods are based on lexicons, which are pre-defined collections of terms and sentiment ratings allied with them. They test several sentiment-scoring algorithms using a set of news items that have been hand-labeled with positive and negative sentiment [54]. While there are several techniques to generate sentiment, and the influence of news has been studied in multiple markets using diverse methodologies, some methods have made an effort to apply the same methodology to assess the impact of news in various markets in a common fashion. For example, classification-based techniques must be retrained for correct application across domains.

1.5.Event Study

On the surface, this appears to be a challenging process, however, an event study may be used to quickly build a measure that analyzes the effects of a news event on the value of related stocks. An event study analyzes the influence of a specific news category on the worth of a stock using financial market data. The value of such research derives from the statistic that, given market logic, the consequences of an event will be instantly observed in the stock price. Thus, stock prices monitored over time may be used to create a measure of the news event's impact. Event studies have been used for several sector-specific events in the financial markets. Mergers and acquisitions, earnings releases, and macroeconomic factors such as the trade imbalance are just a few examples [55].

1.6.Problem statement and Motivation

Online portals provide an enormous amount of news articles every day. Over the years, numerous studies have concluded that news events have a significant impact on forecasting and interpreting the movement of stock prices. The creation of a framework for storing news articles and collecting information for specific domains is an important and untested problem for the Indian stock market. When online news portals produce financial news articles about many subjects simultaneously, finding news articles that are important to the specific domain is nontrivial. A critical component of the aforementioned system should, therefore, include one module for extracting and storing news articles, and another module for classifying these text documents into a specific domain(s).

The banking sector accounts for 7%-8% of India's overall GDP. Since the liberalization of the economy in 1991, the Indian banking sector has experienced significant growth and transformation. Even though the banking sector is well-regulated, it faces several challenges, including economic difficulties and a lack of ethical norms. Recent studies show that financial crises harm growth and that assessing and measuring bank risk on a regular and timely basis is critical to preserving the banking system's stability and profitability [11],[12]. Another study mentions that Fraud has grown significantly in number and value during the previous 10 years, according to the Reserve Bank of India's annual report (2019) and the Economic Times' article (2019) [13]. It also mentions that during 2009–2010, banks reported 4669 frauds worth 1998.94 crores, which increased to 6,801 frauds worth 71542.93 crores by the end of 2018–2019. From 2009–2010 to 2018–2019, the number of incidents of fraud increased by 45.66%, with the sum implicated increasing by 35 times or more. Bank fraud has the potential to have severe

economic and social effects. To measure the bank risks, one of the most often used indicators is the return volatility of banking stocks. These reasons have attracted us to study banking events and their impacts. And to analyze the impact of such events on the valuation of banks through the observations of banking stocks listed in Indian stock exchanges NSE and BSE, the framework is required to isolate the Banking news from the pool of financial news articles published every day on online news portals.

First, the multi-class classification was needed to isolate the Banking news and its most correlated news articles from the pool of financial news articles scraped from various web news portals like financialexpress.com, moneycontrol.com, Bloomberg, yahoofinance.com, and economictimes.com. However, the multiclass classification suffers from data imbalances. Imbalance data is a big difficulty with most classifier learning algorithms. However, as recent works suggest, class imbalances are not in themselves a problem, and degradation in performance is often correlated with certain variables relevant to data distribution, such as the existence in noisy and ambiguous instances in the adjacent class boundaries. A variety of solutions to addressing data imbalances have been proposed recently, over-sampling, down-sampling, and ensemble approach. Our thesis addresses the various challenges that occur with *data imbalances* in multiclass classification and provide solutions in dealing with these challenges.

Second, the banking news articles are required to classify into various events like (Results, Merger_Or_Acquisition, RBI_Policies, Governmental, Global, Fraud, and RatingsAgencies_Experts_View). Although there are several existing text classification methods (covered in Chapter 2), however, there is always a scope to improve the results using “state-of-the-art” NLP (natural language processing) models. The author presents DistilBERT, a method for pre-training a smaller general-purpose language representation model that can subsequently be fine-tuned on a range of tasks like its larger equivalents, such as BERT, with satisfactory results [56]. A deep learning model for solving NLP challenges, BERT [57] employs a Transformer, which is an attention mechanism that learns the context links amongst words in a document (or sub-words). The approach is designed to pre-train deep bidirectional representations from unlabeled text sources by mutual conditioning both left and right contexts. This motivated us to use the power of state-of-art NLP models. To use the benefits of both deep learning techniques and traditional hand-written Rules for event extraction, our thesis focuses on constructing a hybrid model with a DistilBERT language representation model fine-tuned

on our data set of Banking news articles with a machine learning classifier and hand-written Rules for the event extraction and classification.

Economists have conducted a significant number of researches to determine the correlations between news events and stock returns. There has been no systematic effort to automate the tracking of such events and utilize them to analyze banking stocks. There have been individual initiatives to analyze Twitter sentiments or various banking activities, but to our knowledge, no integrated effort to examine the influence of specific categories of news events on banking stocks movements exists. This might be owing to them being unable to compile a database of classifying several potentially important news events from the overall financial news articles that will have an influence on banking stocks in the immediate future. Our thesis aims to address this issue and look at the influence of such news events following sentiments on the price movement of the individual bank's stock, as identified or recognized from these news events, against the benchmark index Bank Nifty and the banking index against the benchmark index Nifty50 in the short to long term. The following research question is the goal of this thesis:

Do news events (RBI announcements, government announcements, mergers or acquisitions, frauds, quarterly or annual results, ratings agencies' or experts' views) with varying sentiments (negative, neutral, or positive) have a significant impact on the stock prices of the bank or the banking index that has undergone events?

1.7. Research Objectives

The research topic of this thesis is **“Design and implementation of event driven news classification technique for Banking stocks analysis”**. The following research objectives have been designed to achieve this goal:

- 1) Automatic online financial news-articles monitoring and representing events of in-domain and out-of domain text documents scrapped from financial news-wires and company's websites using hybrid approach of Machine Learning and Natural Language Processing.
- 2) Classification of financial news into predefined sector(s) and stock(s) listed in Indian stock exchange NSE and BSE using modified Named Entity Recognition technique and event-based news classification using Deep Learning approach.

- 3) Analysis of short and long term reaction on banking stocks in Indian equities based on Linguistic and past knowledge of news events using statistical techniques and machine learning algorithms.

1.8.Research Contribution

This thesis work contributes as the following:

- a. We have performed extensive experiments to classify the financial news articles into the predefined four classes Banking, Non-Banking, Governmental, and Global. The state-of-art multi-class classification model is applied to extract the Banking news and its most correlated news articles from the pool of financial news articles scraped from web news portals financialexpress.com, moneycontrol.com, [Bloomberg](http://Bloomberg.com), yahooofinance.com, and economictimes.com. And, we have presented the various challenges that occur with data imbalances in the multiclass classification and solutions in dealing with these challenges.
- b. A new hybrid approach to text classification that combines a machine learning algorithm with DistilBERT, a pre-trained deep learning framework for natural language processing, provides a base model fine-tuned on our data sets on Indian Banking News-Events, as well as a rule-based technique.
- c. The BERT, a pre-trained deep learning framework for named entity recognition, is further fine-tuned to extract the financial named entities i.e. Banking stocks listed in Indian stock exchanges NSE and BSE.
- d. The sentiment classification model is proposed on Banking and other correlated events Results, Merger or Acquisition, RBI Policies, Governmental, Fraud, and RatingsAgencies & Experts' Views.
- e. The event study is conducted to analyze how a specific banking news event (such as a fraud or a bank merger) and other co-related news events (such as government policies or national elections), as well as the framing of both the news-event and news-events sentiment, impair the formation of the respective bank's stock and the banking index i.e. Bank Nifty in Indian stock markets over time.

1.9.Thesis Organization

The thesis is divided into the following five chapters. The thesis begins with the introduction chapter (Chapter 1), which includes an overview of the research work, text classification framework, Multi-class classification, and problem-solving on data imbalances in multi-class

classification, Event extraction and classification, Banking Named Entities recognition, Sentiment classification for Banking news events, and Event study. This chapter also mentions the research objectives, problem statement, and motivation for consideration.

Chapter-2 presents the state-of-art literature of text classification systems, event extraction, and classification, sentiment classification systems, named entities recognition, and event study.

Chapter 3 presents the suggested Research Methodology of a hybrid model based on a deep learning-based pre-trained language representation model with handwritten rules for event extraction and classification. As a result, the proposed hybrid algorithm's findings must be compared to existing text representation and classification models. It also includes an event study for stock analysis in the banking sector of the Indian stock market.

Chapter 4 presents the validation findings of the proposed hybrid model for news event classification and event study for banking stock analysis of the Indian stock market are shown by comparing the movement of their stocks to a benchmark stock index. The hybrid event classification system's performance is compared to existing text representation and classification methods. The experimental setup, data collecting, and data pre-processing activities are also covered. This chapter also includes a thorough description of the method for classifying banking news events using a hybrid model, as well as a functional example with identified challenges.

Chapter 5 summarizes the thesis work based on the evaluated results and discussions, as well as potential future directions.

Chapter-2

2. Literature Review

This chapter describes the current state of the research literature. The literature review is further divided into five subsections which present the work on event extraction, and text classification systems, sentiment classification systems, named entities recognition, and event study. The review of literature on each subsection is presented as follows:

2.1.Event extraction and text classification

The task of automatically assigning one or more predefined labels to a given text written in natural language, based on their similarities to a previously categorized corpus used as a sample collection, is known as automated text classification. Text classification may be done in one of two ways: manually or automatically. The content of the document is evaluated and categorized by a human annotator in the former. This method could provide amazing outcomes, but it is time-consuming and expensive. To recognize text automatically more quickly and cost-effectively, machine learning, natural language processing, and other AI-based technologies are utilized.

There are two typical methods for text categorization [58]. The first strategy, which was the most popular way for developing automated document classifiers in the 1980s, included manually building an expert model capable of making predictions using knowledge engineering approaches. Typically, such an expert scheme will be made up of a set of manually formulated logical rules of various types, one for each class. If (expression) is true, then (class/label) is true. This method also has the benefit of being simple to comprehend for humans. Aside from the intrinsic difficulty of modeling a text category with a list of logical operations on words, considerable expert knowledge of the domain, as well as a thorough understanding of the ruleset's details, is required. The most significant disadvantage, in this case, is that designing a ruleset when dealing with hundreds or thousands of classes or labels is a massive undertaking that, in most real-world situations, puts this method out of reach [59].

On the other hand, machine learning has been the most popular technique since the late 1990s. In this scenario, the system is given a set of pre-classified (labeled) texts for each class, which is used as a training set, and a classifier is produced automatically from them. The advantage is that domain understanding is only required for the assignment of a label to the current text in the training set, which is a far smaller overhead than establishing the rules. The

literature on rule-based and data-driven methods to text categorization is discussed in the following subsections.

2.1.1. Rule-based approaches

Using linguistic rules, rule-based methods are used to write lexico-syntactic or lexico-semantic patterns. They use linguistic pre-processing and morphological analysis to find probable connections between entities. Using syntactic dependencies or semantic connections, events are found by looking for potential combinations of entities. As a result, rules are used to extract named entities, while another approach is used to identify connections or vice versa. A collection of linguistic resources necessary for a knowledge-based event extraction system is proposed by the author [60].

In other words, rule-based approaches use a set of constructed linguistic rules to classify the text into predefined categories. These recommendations suggest that the framework employs semantically meaningful text components to recognize appropriate classes/labels based on their content. Every rule is made up of an anticipated pattern and a class. The CONSTRUE technique is a well-known example of this approach [61]. CONSTRUE utilized many if-else conditional statements to identify documents in the Reuters dataset's specific class. The rule-based method, which employs the embedding technique to convert a document to vector files, has been the subject of the current study [62] and to classify documents using rules.

To begin, these systems require extensive subject expertise. They are frequently time-consuming, as it can be difficult to develop rules for a complicated structure that requires much study and testing. When the consequences of pre-existing rules may be modified by introducing new rules, rule-based systems are typically hard to manage and scale successfully.

2.1.2. Data-driven approaches

Since the 1990s, the alternative, a data-driven method to text classification, has dominated. This is based on machine learning techniques. Rather than depending on human-developed rules, text classification using machine learning learns to produce categories derived from prior observations. Machine learning algorithms may learn the different relationships between a piece of text and that a certain output (i.e. tags) is required for a specific input, by using pre-labeled samples as training data (i.e., text). A tag is a pre-defined class or group into which any text can be placed.

2.1.3. Text Document Representation: Features construction and weighting

For understanding texts written in any natural language, text classification is the most difficult field of machine learning. Texts, on the other hand, cannot be used as input to machine learning models since virtually all algorithms require numerical input in the form of feature vectors of a predetermined size rather than textual data of variable length. To overcome this problem, textual data must first be converted to document vectors. The following sub-topic examines the many representations used in natural language processing, from the earliest models to the most cutting-edge models.

2.1.3.1. Context-independent approaches

The most commonly used approach in document transformation is Bag-of-words [63]. The model considers every word in the text document as a token/feature although words' order and their context are ignored. Also, the meaningless stop-words are removed from the text documents and most importantly the features are salient w.r.t the problem domain. Then documents are converted as feature vectors where every feature is represented with its frequency number. Each word, sometimes tens or hundreds of dimensions, is represented by a real-valued vector. Conversion of the word into a vector in vector space is called embedding. Typically, supervised NLP methods transform a word to a symbolic ID that is subsequently converted to a feature vector using a *one-hot representation* [64]. Only one dimension is active in the feature vector, which has the same length as the vocabulary size. The one-hot representation of a word, on the other hand, suffers from data sparsity: model parameters for words that are rare in the labeled training data will be badly calculated. Moreover, the feature vector consists of n dimensions where ' n ' represents the number of unique words in the whole dictionary. More frequency means the token/feature has more importance. But it is believed by some researchers that using BOW, words with very high frequency may cause biases and dominate results in the model [65].

The authors have provided a method to overcome the weaknesses of BOW which represents a collection of text documents with weighted frequencies, document-term matrix, where a column signifies a token and a row signifies a document. This scheme of assigning weights to token frequencies in the form of a matrix is called TF-IDF (Term frequency-inverse document frequency). It considers that a very high frequency of words may not be able to give much information gain, in another way; rare words contribute more to the model. During the implementation of this model using the matrix, the value w_{ij} in each cell corresponds to the weight of t_j in d_i which is calculated as $tf\langle t_j, d_i | n_{d_i} \rangle$ where $tf(t_j, d_i)$ represents the count of

token t_j in text-document, d_i and n_{d_i} represents the total quantity of token t_j in document d_i . Due to the simplicity of the model, this is preferably used in Natural Language Processing. The researcher has shown improved features subset using this approach which has taken together with the characteristics of term frequency and document frequency [66][67]. However, even a small collection of documents may consist of a large number of meaningful words which leads to the problem of scalability or high dimensionality. Researchers have been working to find effective ways to decrease running time or reduce high dimensionality in large documents.

The primary alternative for modeling complex text classification or other natural language tasks has emerged in the form of statistical language modeling. In its beginning, however, it used to struggle from the curse of dimensionality when studying typical probability functions of language models [68]. The model will be evaluated on a particular word sequence that may differ from all word sequences used during training. This led to the inspiration to learn distributed representations of low-dimensional space terms. A distributed representation is learned for terms that allow each training text to tell the model of infinite semantically surrounding terms, the authors propose representation to counteract the curse of dimensionality. The ***distributed representations*** describe a co-occurrence matrix of words $\times$ words which considers the frequency of each word that seems in the context of another word, with a window size of k [69]. Distributional vectors have the key advantage of detecting similarities between terms. Followed by this, the researchers have applied ***singular value decomposition*** for text representation, where the matrix decomposition technique was used for reducing a given matrix to its constituent matrices via an extension of the polar decomposition with an idea of making subsequent matrix calculations simpler. It gives the top rank- k constituent parts of the original data. The singular value decomposition will split this down into best rank approximations, which will capture data from the most relevant to the least important [69].

The big popularization of word embedding was possibly due to the continuous bag-of-words (CBOW) paradigm to create high-quality distributed vector representations effectively. The authors have suggested a solution to counter the curse of dimensionality where the distributed representation for every word is concurrently studied along with the probability distribution for word sequences represented in terms of such representations [35]. The ***continuous bag-of-words***, a prediction-based model which directly learns word representation. The distributed representations of context (or surrounding words) are combined in the CBOW model to predict the word in the middle. Authors who proposed the CBOW reshaped Word embedding [70]. The authors have applied a continuous bag of representation with a neural

network model to achieve improved accuracy in classification [71],[72]. The CBOW model is a basic neural network possesses one hidden layer that is connected. The input layer, which takes the one-hot context word vector, has V neurons, while in the hidden layer, N neurons are there. The performance layer is the softmax of all vocabulary terms. Another model studied by authors is skip-gram which further reshaped Word embedding [73]; its architecture works in reverse of what the continuous bag-of-words model does. Each context word is predicted by a model based on the target word. It iterates on the words of each sentence in the given corpus and anticipates its neighbors (its context) using the current term, thus the model is called “*Skip-Gram*” (Local Context Window) [74].

The authors proposed a method for creating a word embedding Glove (Global Vector, combines Count and Predict based approaches) model for distributed word representation. The unsupervised learning technique, in which the model is trained on overall statistics of word-word co-occurrence, such as how frequently it appears in a corpus, and the outcome is a vector representation of words with linear substructures of the word vector space [75]. Glove’s solution is to count how many times a term i (context word) in another term j (target word) occurs. The purpose is to establish a meaning for word i and word j as to whether the two words occur close to N -word apart or not. The encoding vector includes the ratio of two words specifically recognized as a count-based system of co-occurrence probabilities. The prediction-based approach receives popularity, but Glove’s authors claim that the count-based methodology incorporates the global statistics and may be more effective because it outperforms word representation testing on word comparison, term similarity, and called entity recognition tasks.

The enhanced text document representation system was developed to work on the issues of traditional feature-based extraction techniques that included only *Nouns* and *Nouns phrases* to represent the important events called event detection techniques [76]. This technique has used fewer tokens or features than Bag-of-words to handle the problem of scalability or high-dimensionality of documents. Further, this technique has led to another representation based on *Named Entities*. The authors have presented the classification of tweets using Named Entity Recognition to filter out noiseless required information [40][77]. It works by finding proper Nouns in the documents that belong to pre-defined categories. This process involves systematically assigning categories to each term or entity while developing a corpus or labeling process. But this corpus-based representation was unable to represent some domain-specific

words that are in-frequent during training. The authors have proposed techniques to deal with in-frequent or unseen words during labeling [78].

Previous representations were not considering the morphological relation of words to disambiguate the unseen words. Many studies have presented methods that automatically extract features from the documents. These have used in-frequent words which produce a high variety of low anticipated relations between the text documents. This kind of information once aggregated, provides potentially less obvious and hidden relations in the text documents. Using less frequent words with lexical constraints has reduced the associated cost of knowledge re-engineering and it was able to process several documents from a large number of domains [78][26]. These methods help for better representations of texts documents especially handling unseen or less-frequent words. And the problem of scalability was also controlled associated with word's semantical approach. The next approach which was proven most efficient in a domain-specific text representation, **Proper Nouns**, an intermediate solution between Nouns Phrases and Named Entities. This technique has reduced the ambiguity that occurred due to particular associated Nouns with more than one Named Entity category [41]. Recent approaches are concentrating on capturing context beyond the word level to produce performance by giving the more structured and semantic notion of text [79].

2.1.3.2.Context-aware approaches

We have come up with embedding for texts, like word2vec, which transforms a word into an n-dimensional vector. To map the words into a Euclidean space; we can go through an approach to creating **sequence embedding** which brings a sequence into a Euclidean space. With these embedding approaches, we can conduct traditional Machine Learning and Deep Learning on sequence datasets. Usually, sequential data involves significant development of its embedding before it is fed into data mining algorithms. One of the sequential data function learning problems is called sequence embedding [80], where the aim is to convert a sequence into a fixed-length embedding. Deep Learning, **RNN**-based models interpret the sequence as a series of words and are anticipated for text classification to recognize word dependencies and text sequences [81]. By adding a memory cell that can remember values across arbitrary periods, as well as three gates (input gate, output gate, and forget gate) that control the flow of data into and out of the cell, **LSTM** addresses gradient vanishing or exploding issues experienced by the RNNs. In a recursive method, the authors expand the chain-structured LSTM to tree structures, using a memory cell to store the background of multiple child cells or multiple descendants. They claim that the new model offers a realistic way to understand contact between hierarchies

between long distances, e.g., language parse structures [82]. To capture text features, the authors also incorporate a ***Bidirectional-LSTM*** (Bi-LSTM) model with two-dimensional max-pooling [83]. The ***seq2seq*** model is used in various NLP applications [84][85].

Researchers have consistently shown the importance of transfer learning pre-training a neural network model on an established problem in the area of computer vision and then doing fine-tuning utilizing the learned neural network as the foundation for a new purpose-specific model. Researchers have demonstrated in recent years that a related approach may be effective in several tasks relating to natural language. This is another kind of word embedding; the classification algorithms provide a greater sense for learning the features for such embeddings. Yet such embeddings do not take the word order or the word meaning of each sentence into consideration. This is where ***ELMo (Embedding from Models of Language)*** comes into action. ELMo is a contextual embedding that takes into consideration the terms that surround it. It models word use characteristics such as morphology, and how it is used in different contexts. The term vectors are learned features of a deep bidirectional language model (biLM) internal state that is pre-trained on broad text data. The authors have demonstrated that these representations can be readily applied to current frameworks and greatly strengthen the state-of-the-art NLP issues like addressing queries, textual entailment, and interpretation of emotions [86]. A Transformer [87] is another solution to working with long dependencies like LSTM. LSTM is Long Short-Term Memory, a sort of neural network that has a ‘memory unit’ capable of maintaining knowledge in memory over strong periods helping it to learn longer-term dependencies [36]. The Transformer is based through the encoder and decoder on an attention process. The Transformer allows the use of this method to store knowledge about the specific meaning of a given term in its word vector.

Unlike past deep contextualized language representation studies [86] that take into account the backward model as in the ELMo bidirectional LSTM, the authors proposed a new type of language representation named ***BERT***, which stands for ***Bidirectional Transformer Encoder Representations***. BERT uses a Transformer, a mechanism of attention that learns contextual connections between words (or sub-words) in a document. The authors have argued that conventional technologies limit the power of the pre-trained representations, especially for the approaches to fine-tuning. The main constraint is the unidirectional existence of modern language models, which restricts the range of frameworks that can be used during pre-training. By conditioning across both layers in both the left and right contexts, BERT is built to pre-train deep bidirectional representations from unlabeled text sources [57].

In setting language modeling, transformers can acquire longer-term dependence but are constrained with a fixed-length context. The authors suggested a novel *Transformer-XL* neural architecture that allows learning dependence to interrupt temporal coherence beyond a fixed length. It consists of a recurrence function at the segment level and a novel positional encoding scheme [87]. Learning regarding the inductive transfer has significantly impacted computer vision, but current techniques in NLP also need complex task modifications and preparation from scratch. The authors suggested *Universal Language Model Fine-tuning (ULMFIT)*, an important transfer learning approach that can be extended to any NLP task, and implemented techniques that are essential to fine-tuning a model of language [88].

Before feeding word sequences into BERT, 15% of the words in each sequence are replaced with a mask token. Based on the context provided by the other, unmasked words, the model then attempts to determine the true context of the masked words in the list. The BERT loss function only finds the estimation of the masked terms and excludes the estimation of the unmasked phrases. As a result, the model converges slower than directional ones, a feature offset by its enhanced understanding of the context. Centered on BERT's masking technique [57], authors developed a novel language representation model enhanced by knowledge masking techniques named *ERNIE (Enhanced Representation by Knowledge Integration)* [89], which involves masking at the entity level and masking at the phrase level. Strategy at the entity level covers entities that typically consist of several terms. The phrase-level technique covers the whole phrase consisting of many words that serve as a cohesive entity together. Their experimental findings indicate that ERNIE outperforms other standard approaches by obtaining modern “state-of-the-art” outcomes on NLP activities, including natural language inference, conceptual similarity, named entity identification, emotion analysis, and question answering.

The Transformer paradigm is generally popular on several tasks relating to natural language processing. Using a transformer is therefore also an expensive operation since it requires the method of self-attention. Self-attention assumes we’re conducting the task of attention on the sentence itself, as compared to 2 separate sentences. Self-attention allows defining the connection in a single sentence between the words. It is the function of self-attention that adds to the expense of utilizing a transformer. The quadratic structure of self-attention, however, constrains its operation on long text. The authors follow a method of fine to coarse attention on multi-scale spans by binary partitioning (BP), they suggest *BP-Transformer*. BP-Transformer has a strong balance between the complexity of computations

and the capability of models. The authors performed a series of experiments on text classification and language processing, showing BP-Transformer performs superior to previous self-attention models for long text [90]. By viewing the transformer as a graph neural network, the Binary-Partitioning Transformer aims to improve the utility of the self-attention mechanism. An input token is represented by any node in this graph.

Another research suggests a variant of the Neural Attentive Bag-of-Entities, which is a neural network algorithm that uses entities in a knowledge base to conduct text classification. Entities include unambiguous and specific syntactic and semantic signs that are useful for catching semantics in documents. The authors put together easy high-recall dictionary-based entity recognition, with a neural attention system that helps the model to concentrate on a limited number of unambiguous and specific entities in a text [91]. The model first identifies entities to whom this name might be addressed (e.g., Ap Inc., Apple (food)), and then describes the entity using the weighted average of all entities' embeddings. The weights are measured using a modern method of neural attention that helps the model to concentrate on a specific subset of entities that are less ambiguous in context and more important to the document.

It's the time for NLP when the transition started as we listed the unsupervised pre-trained language models that had made breakthroughs in different tasks of understanding the natural language, such as named entity identification, emotion interpretation, and question-answer records for art performance beginning one after another in that short period. These NLP models indicate that a lot more is yet to come, and the authors look forward to researching and implementing them.

2.1.4. Data Preprocessing: Data Cleaning

There is a lot of text data available today, and data is being grown daily in structured, semi-structured, or fully unstructured form. It is always necessary to process the raw corpus data to execute the text classification operation. Data processing entails several steps, including data cleaning (organizing data according to structure and removing unnecessary sub-texts), tokenization (breaking up text into words), normalization (converting all texts into the same case), removing punctuation (stemming leaves out root forms of the verb and lemmatization), and substitution (finding candidate words for translation and performing word sense disambiguation) [92]. In one of the studies [93], The researchers also looked at how machine learning approaches may help designers detect comparable sentences when text data is obtained from a variety of sources. The text documents are labeled using two typical techniques in the labeling task: one is to label each section of text separately, and the other is to label the entire

collection of texts. The first method is known as Supervised Learning, while the second is known as Multi-Instance Learning [94],[95].

2.1.5. Data Preprocessing: Dimensionality Reduction

Dimensionality reduction is a crucial approach for preparing data for text classification in data preparation. It is done to reduce the classifier's memory needs and execution time, therefore increasing the learning model's efficiency and efficacy. The dimensions of data are increasing as a result of the increased volume of data. To map large dimensions to space with low dimensions, it becomes necessary to reduce the dimensions of the data [96],[97]. The objective of decreasing high-dimensional space is to find a sub-dimensional space with the least amount of complexity and the greatest ability to modify the learning model. In certain cases, the number of characteristics in samples is significantly larger than the number of samples, according to several studies. This leads to a problem known as overfitting [21]. As a result, dimensionality reduction becomes necessary to avoid overfitting. Feature Selection and Feature Extraction are two key sub-tasks in decreasing dimensionality. The approach of giving the part of the original characteristics that are important for the task is known as feature selection. Feature extraction is a technique for converting a space with many dimensions into a new space with few dimensions, to increase data variance [98].

2.1.5.1. Standard Feature Selection methods

The following are the goals of the feature selection technique:

- i. To improve the classifiers' prediction accuracy
- ii. To create a cost-effective classifier while also speeding up the process
- iii. To improve the clarity of the data-gathering approach

As illustrated in Fig. 2.1, features selection is a way of identifying a part of the original characteristics that are essential for the training set. It's used to create an effective classifier while keeping important lexical characteristics in mind [99]. It removes the extraneous characteristics that tend to reduce accuracy [100]. The goal of feature selection is to limit the feature space by picking a subset of basic features that avoids overfitting while maintaining text classification performance [101].

For instance, a feature set

$$X = \{X_1, X_2, \dots, X_N\}$$

N is a set of feature sets that are divided into $2N$ feature subsets, each of which is represented by a vector of size N . The methods locate a feature subset of size K , KN without affecting the

accuracy of the original feature set. It's a research topic, and the writers have so far provided several techniques. Filter, Wrapper, and Embedded are the three types of techniques available. [102],[103]. Filter-based Feature Selection offers many options for evaluating each feature's information value. The filter technique chooses the top-N features based on the results of multiple statistical tests to identify a connection with the target label or to establish which characteristics are more predictive of the target label, and it is independent of any learning algorithms. Because this technique ignores feature dependencies, it is computationally efficient. The Wrapper approach ranks a subset of features according to their predictive value using a learning model, but it is computationally very expensive owing to the repetitive learning process and cross validation. Embedded techniques are similar to wrapper methods, except that they include feature selection during the training phase [104].

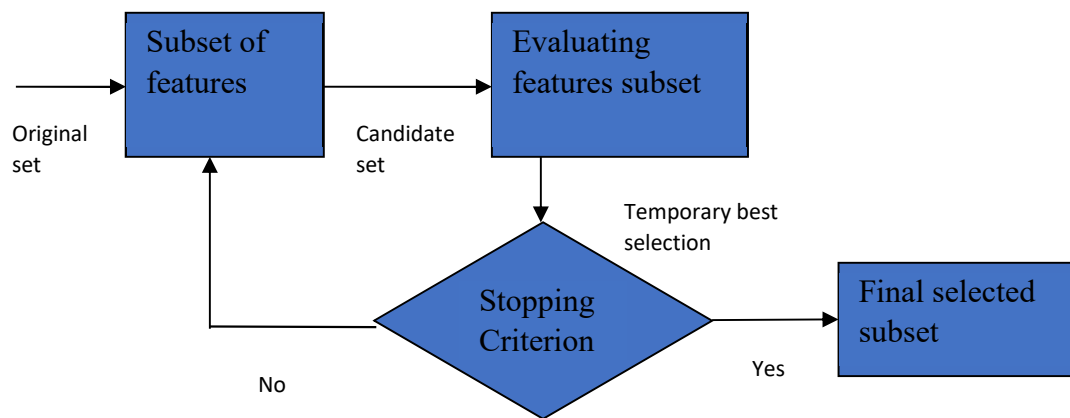


Fig. 2.1. Criteria for feature selection for dimensionality reduction

A. Filter methods

a) Univariate feature selection

Filters perform without taking the classifier into account. They make the classifier work more efficiently in terms of computing. Univariate and Multivariate are the two most often utilized filter techniques.

Univariate methods look at each feature in a text document independently, whereas multivariate methods look for relationships between them.

Univariate feature selection analyzes each feature separately to determine the features' connection to the target variable. The following Univariate feature selection approaches are linear since they pertain to linear classifiers built using just one variable.

The filter technique chooses features without focusing on the core goal of improving any classifier's performance. It scores the features based on the data's most important attributes. If

d features are given represented as S, the goal of the filter-based method is to choose $m < d$ features, represented by T, that maximize some function F:

$$\tau^* = \underset{\tau \subseteq S}{arg \max} F(\tau), \text{ s. t. } |\tau| = m \quad (2.1)$$

It finally settles on the top-m rated features with the highest scores.

The most widely used linear Univariate filter techniques for text classification are discussed here:

The **information gain** technique selects features based on the item's frequency concerning the class/label prediction. Researchers have demonstrated that by removing unnecessary features without modifying the features, the technique may lower the vector-dimensionality of text and enhance classification results [105]. It is defined as the amount of knowledge it generates in terms of the word's availability or non-availability while deciding on any classifier. When text maps to a certain label or class and a word are also accessible in the document, it adds the most value. The equation 2.1 can be formed as:

$$IG(t) = - \sum_{i=1}^m P(C_i) \log p(C_i) + p(t) \sum_{i=1}^m P(C_i|t) \log p(C_i|t) + p(\bar{t}) \sum_{i=1}^m P(C_i|\bar{t}) \log p(C_i|\bar{t}) \quad (2.2)$$

Information Gain is represented as IG , C_i is the i th class, $P(C_i)$ is the probability of an i th class, and m is the number of target classes. The probability of feature t appears in the documents is $P(t)$ and whereas the probability of feature t does not appear in the document is $P(\bar{t})$. The conditional probability of the feature t being in the i th class is $P(C_i|t)$. The conditional probability of feature t does not exist in the i th class is $P(C_i|\bar{t})$.

The **Chi-square** test is a statistical tool for assessing the association between a collection of categorical features using their frequency distribution and determining how much the findings differ from the expected output [106],[107]. This may be determined for events A and B that are considered to be independent if:

$$p(AB) = p(A)p(B) \quad (2.3)$$

Chi-square can be calculated from the following equation 4:

$$CHI^2(t, C) = \sum_{t \in \{0,1\}} \sum_{C \in \{0,1\}} \frac{(N_{t,C} - E_{t,C})^2}{E_{t,C}} \quad (2.4)$$

t represents the feature, C represents the specific class, $N_{t,C}$ refers to the frequency of feature t and class C occurred together and $E_{t,C}$ refers to the frequency of feature t appears without

class C . The chi-square is calculated between every feature and class, and then select the features with the best chi score.

Fisher score calculates the variance of the expected value from the observed value to determine the information score, which indicates how much information one variable has about the unknown parameter on which the variable is dependent. When the variance is the minimum, the information score is the maximum. For the Support Vector-based feature ranking model, researchers applied Fisher's linear discriminant [108],[109],[110].

For instance, let μ_j^k and σ_j^k is the mean and standard deviation of k^{th} class, for the j^{th} feature. Let μ^j and σ^j represents the mean and standard deviation, respectively, of the entire training data for the j^{th} feature.

The Fisher equation for determining the j^{th} feature's score is as follows:

$$F(x^j) = \frac{\sum_{k=1}^c n_k (\mu_k^j - \mu^j)^2}{\sigma_j^2} \quad (2.5)$$

Here, σ_j^2 is computed as $\sum_{k=1}^c n_k (\sigma_k^j)^2$. Top-m features with higher fisher scores are chosen by the algorithms.

Pearson's correlation coefficient is used to calculate linear dependency between two continuous variables by dividing their covariance by the product of their Standard Deviation, and its value ranges from -1 to +1. In two variables, a -1 value denotes negative correlation, a +1 value shows a positive correlation, and a 0 value represents no linear correlation [104].

Using vectors, the Pearson's coefficient r can be computed as:

$$r = \frac{(x_1 - \bar{x}_1 L)(x_2 - \bar{x}_2 L)}{(|x_1||x_2|)} \quad (2.6)$$

Here, $\bar{x}_1$ is the mean of vector x_1 and similarly for x_2 . L is the vector of 1s. $|x|$ is the magnitude of vector x .

To decrease the vector dimensionality, the variance threshold technique is used to remove all low-variance features. Features that have a lower training-set variance than the threshold will be eliminated [111],[112].

$$s = \sqrt{\sum (x_i - \bar{x})^2 / (n - 1)} \quad (2.7)$$

The equation may be used to find features that have a variation below a specific threshold. When the feature does not vary much within itself, it is seen to have low predictive power.

b) Multivariate filter methods

During the assessment of the multivariate filter selection technique, the interdependencies of features are also taken into account to pick relevant features.

mRMR (Minimal redundancy maximum relevance) based on mutual information, the features in the feature set with the highest dependency on the target label are found using maximum dependency. However, it is not appropriate for applications where the goal is to achieve high accuracy with fewer features.

The alternative is to leverage max relevance, which discovers features with dependence by averaging all mutual information values between all features x_i and target label c . S stands for a set of features and I stands for mutual information, which is computed in equation 8 between feature i and class c .

$$\min R(S), R = \frac{1}{|S|^2} \sum_{x_i, x_j \in S} I(x_i, x_j) \quad (2.8)$$

However, this results in a high level of redundancy, i.e. a higher level of dependency between features. As a result, to find mutually exclusive features, minimum redundancy can be used [113].

$$\min R(S), R = \frac{1}{|S|^2} \sum_{x_i \in S} I(x_i, x_j) \quad (2.9)$$

$I(x_i, x_j)$ is the mutual information between feature i and j .

Multivariate Relative Discriminative Criterion, the author presents a multivariate selection approach that takes into account both feature relevance and redundancy in the selection process. The RDC is used to assess the relevance, whereas Pearson's correlation is used to assess redundancy between features [114].

$$RDC(w_i, tc_j(w_i)) = \frac{(|d f_{pos}(w_i) - d f_{neg}(w_i)|)}{\min(d f_{pos}(w_i), d f_{neg}(w_i)) * tc_j(w_i)} \quad (2.10)$$

Here, $d f_{pos}(w_i), d f_{neg}(w_i)$ is the collection of positive and negative text documents that include the term w_i . The term may appear many times in specific documents and represent by $tc_j(w_i)$.

Researchers have been looking for novel techniques to enhance classification accuracy while also reducing processing time. The author has provided a differentiate feature selector, a filter-based approach that has picked unique features that have term properties while

eliminating uninformative ones [103]. It provided efficiency by reducing processing time and improving classification accuracy.

B. Wrapper Methods

Wrappers techniques are directly tied to a certain classifier; the methods pick a subset of features depending on their influence on the classifier, which is determined by assessing the predicting performance of all potential feature subsets in a given space. It indicates that the features subset will be assessed by interacting with the classifier, which can improve the classification technique's accuracy. As the feature space expands, the computing efficiency suffers as a result of this technique. Wrappers are used to choose features for other models as filters.

The process can be accomplished in three ways: the first methodology employs the best-first search approach; the second methodology employs a stochastic approach such as random selection, and the third methodology employs heuristics such as forward and backward passes to include and exclude features.

a) Multivariate Feature Selection

Univariate feature selection approaches are computationally efficient, but they reject features owing to a lack of interaction between features that, when combined, may have contributed significant information regarding classification [115],[116].

When evaluating the performance of features, multivariate takes into account the interdependencies between them. "Linear multivariate" employs linear classifiers made up of a subset of features, with the score of feature subsets being calculated based on classification performance. Non-linear multivariate, on the other hand, use non-linear classifiers to complete the task.

In-text classification commonly preferred **linear Multivariate wrapper** methods are:

Recursive Feature Elimination is a recursive technique that ranks features according to a key metric. During each cycle, the relevance of features is assessed, and less significant features are removed. To design ranking, the inverse technique is employed, in which features are removed. From this evaluation, this method extracts the top-N features. [117]. This is a greedy optimization that seeks for the highest performing feature subset.

Forward/Backward stepwise selection is an iterative procedure that begins with the evaluation of each feature and then picks the model that performs the best based on a set of evaluation criteria. Then, all potential combinations of that selected feature and the following feature are examined, and the second feature is chosen if it improves the model. The model

continuously appending the list of features that best increases the model's performance in each iteration until a subset of the needed features is picked. Every iteration, the backward feature selection starts with the complete collection of features and discards the least suitable feature, improving the method's performance even further. This procedure is repeated until no improvement is shown when features are removed, and the best subset of features is found. In comparison to prior approaches, the researcher devised a rapid forward selection strategy for identifying the optimal subset of a feature that required less computing effort [118].

The **genetic algorithm** uses a feature set to generate a better subset of features that are free of noise. At each step, a new subset is generated by choosing individual features in the correct sequence and merging them using natural genetics-based processes. The result is cross-validated variance divided by the accurate prediction percentage. The end outcome may be mutated. This procedure aids in the creation of an individual feature set that is better suited to the model than the original feature set. The chaotic genetic algorithm was designed to simplify the feature selection process and improve the classification technique's accuracy [119][120].

The following are some of the most popular **Non-Linear Multivariate Wrapper** techniques for text classification:

Non-Linear kernel multiplicative updates entail iteratively training a classifier and rescaling the feature set by multiplying them by a scaling factor that decreases the relevance of the feature that is less affected. Non-linear methods can outperform linear algorithms by selecting a subset of features [121].

Relief is considered the concept of instance-based learning. For each feature's relevance to the target label, the algorithm assigns a score ranging from -1 to +1. The algorithm's scope is suitable for binary classification [122],[123].

C. Embedded Methods

In terms of computing performance, "embedded methods" outperform wrappers, but they execute selection features as a sub-part of the learning approach, which is largely particular to the learning model and may not run with any other classifier.

The following are the most widely used embedded methods for text classification:

In social sciences, the **LASSO** approach is widely utilized [124]. To decrease the dimensionality problem, it makes features with large coefficients undesirable by introducing a penalty during the log-likelihood maximization procedure. By picking an appropriate weight

and reducing dimensionality, LASSO assigns zero to some coefficients. When there is a significant correlation between several features, it becomes an issue [125].

By downsizing the coefficients, **Ridge Regression** decreases the model's complexity while keeping all of its features. The issue with ridge regression is that features are retained. If the feature collection is big, the problem becomes more complicated [126].

As a compromise between lasso and ridge penalties, **Elastic Net** applies a penalty. The elastic net penalty may be readily handled to increase the power of either lasso or ridge penalties. The features with a significant association are likely to be in or out of the feature subset, exhibiting a grouping effect. It incorporates both L1 and L2 regularization techniques (Lasso and Ridge). By fine-tuning the settings, it aids in the implementation of both approaches [127].

2.1.5.2. Text Feature Extraction methods

After selecting the features and representing N -documents by d -dimensional features *vectors* $X_1, X_2, \dots, X_N$. Sometimes original terms in the form of features may not be ideal dimensions for text representation. The text feature extraction methods attempt to solve these issues by developing new feature space Y or artificial terms [58]. It requires a) a method to convert old terms to new terms b) a method to convert document representation from old to new. A popular example commonly used for this purpose is Principal component analysis (PCA) where a feature set Y_i is chosen in a manner that the original feature vectors' variance is maximized in the direction of new feature vectors. This is done by computing eigenvectors of the covariance matrix of the original vectors. The drawback of the PCA is the time it takes to evaluate eigenvalue decomposition to compute the principal component for each class/label when applied to a large dataset. This is overcome by the researcher by using the Power factorization method (PFM) to find a fixed quantity of eigenvectors from a dataset [25]. Another commonly preferred method for feature extraction is latent semantic indexing (LSI) which uses singular value decomposition of the term correlation matrix computed from a big text corpus. This technique is used to adhere to the problem of developing from the use of synonymous and polysemous words as dimensions of the text representation. But the disadvantage is to compute the exact total of latent components and it proves costly (computationally). Another method that helps for optimal discrimination of data is Linear discriminant analysis (LDA) [110]. It identifies the linear collection of features that best explain the data. It tries to find the model which can explicitly differentiate between the classes of data. Latent Dirichlet allocation is

another method that explains that each text document is a mixture of latent subjects and each word in that document is attributable to one of the subjects of that document. This is most preferred for topic modeling [128]. This is a generative probabilistic model.

2.1.6. Classifiers for Classification Task

The classifier is trained based on the selected features from the text documents. The selection of appropriate features in feature space decides the performance of learning models. Machines understand numbers easily than texts as input. So, texts as tokens are required to be converted into numbers (vectorization) for most of the learning algorithms. Vectors are combined to originate vector space to apply statistical methods for checking document relatedness. Each algorithm offers a different document representation for text classification. Researchers offer many classification methods that work on the vector representation of texts. The basic assumption for vector representation of texts is known as the continuity hypothesis. It states that texts documents belonging to the same class develop a contiguous region and regions of different classes do not overlap. The relatedness of the documents can be evaluated on 2D space based on cosine similarity or Euclidean distance.

The authors have presented Naive Bayes, a linear classifier, to vectorize text documents according to probability distributions using two widely known models: To minimize dimensionality, the features with the highest probability were taken for the Multivariate Bernoulli Event and Multinomial Event [20].

$$P\langle C_k | d_i \rangle = P\langle d_i | C_k \rangle * \frac{P(C_k)}{P(d_i)} \quad (2.11)$$

The output of the classifier is the probability of the text document d_i is belonging to each class C_k and it is the vector of C elements. In a way of text classification scenario, we could compute $P\langle d_i | C_k \rangle$ using Bag-of-words.

$$P\langle d_i | C_k \rangle = P(\langle BoW(d_i) | C_k \rangle) = P\langle w_{1,i} w_{2,i}, \dots, w_{|V|,i} | C_k \rangle \quad (2.12)$$

The problem may be reduced to compute the likelihood of each word $w_{j,i}$ in class C_k .

$$P\langle d_i | C_k \rangle = \prod_j^{|V|} P\langle w_{j,i} | C_k \rangle \quad (2.13)$$

The Naïve Bayes has more bias to a non-linear problem due to its nature of modeling one type of class i.e. a linear hyperplane. The bias is the statistical method of evaluating the performance of a classifier that how accurately a classifier classifies the texts into correct class with less error. The learning task is the last activity in classification but reducing the feature dimensions is more concerned with the efficiency of the classifier model. The possibility of

salient feature reduction caused by using classifiers can be overcome by using the model SVM which identifies the best decision-boundary between features vectors of the document with their categories [129]. The SVM entails an optimal classifier that guarantees the lowest classification error. The SVM computes a hyperplane that falls between the positive and negative example of the training set.

$$D = \{(x_i, y_i) | x_i \in R^p, y_i \in \{-1, 1\}\} \quad i = 1, \dots, n \quad (2.14)$$

Here, the minus sign represents the negative hyperplane, and the positive sign points to the positive hyperplane. (x_i, y_i) represents features vectors of the document and D is evaluated to +1 and -1 for positive and negative hyperplanes.

The Naïve Bayes itself results in the best classification model if it is trained on the high volume of data. However, feature reduction remains an issue. So, Naïve Bayes is used as a pre-step to SVM which converts text documents into vectors before the classification task starts. This resulted in improving the whole system while spending quite an appropriate classification time by reducing to low dimensional space. But in certain cases the majority of features are redundant with each other, the author has presented a divergence-based feature selection method without relying on a complex dependence model where the maximum marginal relevance-based feature selection is outperformed by the SVM [130]. The paper has suggested the need for novel criteria to determine the relevance and newness of features separately and provided the linear combination as the metric.

The studies have mentioned the KNN, a non-linear classifier, where the algorithm classifies the document by moving through all the training documents that are like that document. The KNN model frames the documents in the Euclidean space as points so that the distance between two-point u and v can be calculated.

$$D(u, v)^2 = \|u - v\|^2 = (u - v)^T (u - v) = \sum_{i=1}^d (u_i - v_i)^2 \quad (2.15)$$

The classifier finds the K-value which is the factor that represents a collection of documents from all the documents closest to the selected document in that space [131].

$$\hat{y}(x) = y_{n^*} \text{ where } n^* = \arg \min_{n \in D} \text{dist}(x, x_n) \quad (2.16)$$

The study mentions that using KNN increases the overhead to calculate the K-value of all the documents with all other training documents with the largest similarity or closet to the selected document. Also, the difference in the quantity of training sample documents in various categories leads to a declining accuracy. Due to the high variance and complex regions between classes, it becomes sensitive to noise documents. Sometimes the document tends to misclassify

if it occurs very relevant to a noise document in the training set and at times accurately classified if there is no presence of noise documents in the training data close to them. This ends up in high variance from the training data to the training data. High variance leads to overfitting. The goal of finding a good learning classifier is to decrease the learning error. Learning error is calculated from bias and variance or bias-variance tradeoff. The traditional algorithms possess some limitations which attract the researchers to improve the efficiency by a) reducing the computational overheads by establishing low-dimensional space b) speeding up the computational capacity of finding nearest neighbors in KNN or locating decision boundaries in SVM c) increasing efficiency by not compromising accuracy [132]. The model is chosen which optimizes the fit to the training data.

$$H^* = \arg \max_H \text{fit}(H|D) \quad (2.17)$$

The supervised learning classification algorithms Naïve Bayes, SVM, and KNN use a bag of words to design a classifier. None of these methods take the order of words into consideration which can lead to the loss of some valuable information. In Natural Language-based problems, the order of the words e.g. multi-words has a meaning (like names of organization or person) that is not considered by the learning models trained on individual words of the texts. The algorithm n-grams consider the sequence of n-adjacent words from the selected text phrases. The n-grams behave like the individual word's representation as feature vectors. The value of n may range from 1 to upper value (e.g. 5) [133]. This proves very beneficial in short text documents where the number of n-grams is less in number [99]. The author proposes another representation for character n-grams to introduce the enhancement of the skip-gram model which considers sub-word into account. It considers the word morphology while training the model and words are represented by the sum of its character n-gram [74].

Many researchers have used a Decision Tree-based algorithm (decision support tool) that represents a tree-structure-based graph of decisions [27]. The commonly used decision tree algorithms are ID3, C4.5, and C5. The algorithm presents each intermediate node (labeled as terms and branches represent weight) which can split into sub-trees and ends at leaf nodes (represents the class/label/outcome of the problem). The solutions from Decision tree structures are rule-based. A rule can be designed by developing a conjunct of every test that appears on the path between the root node and the leaf node of the tree. The rules are formed after traversing every path from a root to the leaf node. Once the decision tree and rule are framed then it helps to assign the class/label for a new case [134][135]. It was evaluated in the study

that decision trees result better than Naïve Bayes in terms of accuracy but little worse than KNN methods [136].

Neural network or deep learning techniques use many processing layers to build hierarchical data representations and have produced “state-of-the-art” outcomes in various fields. Several model designs and techniques in natural language processing (NLP) have recently advanced. Deep learning architectures and algorithms have generated significant advances in areas like computer vision and pattern recognition. Recent NLP research has centered on the application of new deep learning techniques, which is a continuation of this trend. Deep learning architectures and algorithms have made significant advances in fields such as computer vision and pattern recognition.

In the late 90’s, the researcher has found an application of non-linear neural networks to text classification or topic modeling [137]. In this model, a three-layered neural network was designed to learn the non-linear mapping from training documents to each class/label. Later on, the researcher proposes Convolutional Neural Network (CNN) for text classification by considering the order of words in the phrases and this outperforms SVM in terms of error rate [138]. CNN uses the vector representation of text data considering the order of words. Each word is considered a pixel and the document is treated as an image. Then the image is taken into $|D| \times 1$ pixels and each pixel represents a word as a $|V|$ dimensional vector. For instance, vocabulary $V = \{\text{'classification'}, \text{'course'}, \text{'I'}, \text{'love'}, \text{'NLP'}, \text{'text'}\}$ and words are taken as a dimension of vectors in alphabetical order. And document $D = \text{"I love NLP course"}$. Then the document vector would be:

$X = [0010000 \mid 000100 \mid 00001 \mid 010000]$

The researcher mentions that to reduce the dimensionality of vector space the vocabulary size must be kept low. Also, the n-gram algorithm ignores the fact that some n-grams share the constituent words, this is overcome by CNN which learns the embedding of text regions by providing CNN the constituent words as input and this technique provides higher accuracy. To construct an informative latent semantic representation of the sentence for downstream activities, CNNs have the potential to extract salient n-gram features from the input sentence. The author proposes a three-way enhanced CNN for classification for sentiment analysis, where decisions are divided into three parts accept, reject and delay. The instances in boundary regions that are neither in accept nor reject are reclassified by another classification model. This guarantees the enhancement in CNN to deal with boundary regions in a better way, resulting in model 3W-CNN [139]. Another study has shown the application of CNN in

document modeling for personality detection based on text in the context of sentiment analysis [23]. Overall, in contextual windows, CNNs are highly successful in mining semantic hints. They are very data-heavy models, though. They have a huge range of parameters that are trainable and need tremendous training data. This raises a concern as data shortage happens. Another unresolved concern with CNNs is their failure to model contextual long-distance data and maintain sequential order in their representations [140].

Deep Neural networks are difficult to train on data, it requires a lot of resources, to get high performance. The feed-forward neural network commonly known as multilayer perceptron (MLP) is the most preferred technique in classification problems. In a feed-forward neural network, the information travels in one direction from the input layer to hidden layers and is then followed by the output layer. They have no memory of the input received previously so lack in predicting what comes next. To overcome this RNN (Recurrent Neural Network) is preferred where information moves through the loop. In this paragraph, we discuss the fundamental characteristics that have favored the popularization of RNNs in a variety of NLP tasks. Since an RNN performs sequential processing in sequence by modeling units, it may have the ability to generate the intrinsic sequential structure present in language, where characters, words, or even phrases are units. Based on the previous words in the sentence, words in a language establish their semantic meaning. The disparity in interpretation between "computer" and "computer vision" is a clear example that states this. RNNs are perfect for language and related sequence modeling activities to predict certain context dependencies, which turned out to be a clear incentive for researchers to use RNNs over CNNs in these fields.

RNNs were originally three-layer networks in the NLP sense [141]. While deciding on the current input layer, it considers what it has learned from the previous inputs. Basically, in the architecture of simple RNN, the hidden units create internal representations for the input patterns and recode these patterns in feedforward networks using hidden units and a learning algorithm in a way that allows the network to generate the appropriate output for a given input. Typically, the hidden state of the RNN is assumed to be the most important feature. It can be regarded as the memory portion of the network that accumulates data from other steps.

The formula of the current state in RNN can be written as mentioned below. A non-linear transformation such as $\tanh$, or ReLU, is taken to be the function f .

$$h_t = f(h_{t-1}, x_t) \quad (2.18)$$

Here, h_t is the new state and h_{t-1} is the previous state. x_t is the input at time t .

The $\tanh$ function is commonly used as an activation function. The weights can be defined as the matrix W_{hh} and input is defined by the matrix W_{xh} .

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}X_t) \quad (2.19)$$

The output may be measured during test time as:

$$y_t = W_{hy}h_t \quad (2.20)$$

The outcome is then compared to the original output and then the error value is calculated. The network learns by back-propagating the error through the network to update the weights. But usual RNN has a short-term memory. These basic RNN networks suffer from the issue of vanishing gradient, which makes it very difficult to understand and adjust the parameters of the previous layers in the network. It is used in combination with LSTM which has long-term memory and it gives extension to the memory of the usual RNN. Over the basic RNN, LSTM has additional "forget" gates, allowing the error to back-propagate over an infinite amount of time steps. Comprising three gates: input, forget and output gates, by taking a combination of these three gates, determines the hidden state [36]. RNN and LSTM have been used in many NLP applications due to their capacity of capturing complex patterns within the text [142]. It uses in sequence labeling tasks in POS (part of speech) activity. It is preferably used in topic modeling for fake news [143], [144], sentiment analysis, and negative speech detection on social media. More recently, authors have suggested another type of recurrent unit that they refer to as a gated recurrent unit (GRU) [145]. It has been shown that RNNs employing any of these recurrent units perform well in tasks (such as machine translation, speech recognition, or depending parsing in text documents for NER) requiring long-term dependency capture. The application of Gated RNN is not limited to the mentioned tasks but it can be applied to different NLP challenges [146].

Recurrent neural networks, in particular long short-term memory [36], and gated recurrent neural networks [86] are firmly known as "state-of-the-art" approaches in sequence modeling. Within training examples, the inherently sequential nature of recurring models prevents parallelization, which becomes important at longer sequence lengths, as memory limitations restrict batching through examples. Recent work, through factorization tricks [147] and conditional computation [148], has achieved substantial improvements in computational efficiency, while also improving model output in the case of the latter. The authors have presented two simple ways of reducing the number of parameters and speeding up the training

of large Long-Short-Term Memory networks: the first is the ‘matrix factorization by design’ of the LSTM matrix into two smaller matrices, and the second is the division into separate classes of the LSTM matrix, its inputs, and states. However, the essential restriction of sequential computation persists.

In many applications, attention mechanisms have become an essential part of sequence modeling, allowing dependencies to be represented regardless of their gap in the input or output sequences. The following paragraph examines the few most prominent models of attention that have created new state-of-the-art tasks for text classification. The study's conclusion was largely focused on text classification experiments that could not be extended to many other NLP tasks [149]. The authors mentioned that attention offers a reliable explanation for model predictions, expecting these properties to hold a) Attention weights should align with feature-relevant measurements (e.g. gradient-based measurements); b) Alternative (or counter-factual) weight configurations should result in corresponding prediction changes. They stated that in the sense of text classification, neither property is consistently observed by a BiLSTM with a standard attention mechanism. The layers of attention specifically weigh the representations of the input elements; it is also often believed that attention can be used to classify information that was considered relevant by models. In another study, the authors evaluate if that assumption holds by modifying weights of attention in already trained text classification methods and examining the resulting variations in their estimations. Although experimenting with text classification, the authors note several cases in which higher attention weights correlate with greater impact on model predictions, they also notice several cases this doesn't hold, i.e. where gradient-based attention weight rankings predict their effects better than their magnitudes [150]. In contrast to the current work on interpretability, the authors of another research showed the attention mechanism in a wider range of NLP tasks, including text classification, paired text classification, and text generation tasks like neural machine translation [151].

Although the hidden vectors represented by an attention model through encoding can be interpreted as internal memory entries for the model, memory-augmented networks integrate neural networks with an external memory type that the model could learn from it and respond to. For text classification, the study proposes a memory-augmented neural network called the Neural Semantic Encoder [152]. In another research, the authors introduce neural network architecture, the Dynamic Memory Network (DMN), which processes input sequences and questions, shapes episodic memories and generates appropriate responses. The model

possesses an iterative mechanism of attention that allows the model to condition its attention on the inputs and outcomes of previous iterations. In a hierarchical recurrent sequence model, these outcomes are then reasoned over to produce answers [153]. The authors also stated that it is possible to train the DMN end-to-end and obtain state-of-the-art text classification results using the Stanford Sentiment Treebank data collection.

Sequential processing of text is one of the system inefficiencies experienced by RNNs. Transformers address this constraint by applying self-attention to measure an "attention ranking" in parallel for each word in a phrase or document to model the impact each word has on another. Because of this feature, Transformers allow for far more parallelization than CNNs and RNNs, allowing very large models to be efficiently trained on large quantities of data on GPU stacks [154]. The Transformer architecture is especially suitable for pre-training large text corpus, leading to significant accuracy improvements in downstream tasks, including text classification [87]. They propose a novel Transformer-XL neural architecture that allows learning dependence to interrupt temporal coherence beyond a fixed length. We have seen the emergence of several large-scale transformer-based pre-trained language models in the current scenario. As stated in the section on text representation 2.2.2, these Transformers are pre-trained to learn contextual text representations in much greater volumes of text corpora by predicting terms that are trained on their context. These pre-trained models were fine-tuned using task-specific tags, and in many subsequent NLP tasks, especially text classification produced new state-of-the-art. Fine-tuning is supervised learning, while pre-training is unsupervised.

The authors design the largest model, OpenGPT, 1.5B parameter Transformer that ensures state-of-the-art results and comprises 12 layers of Transformer frames, each composed of a masked multi-head attention unit, followed by a standardization layer and a forward feed layer in place. With the addition of task-specific linear classifiers and fine-tuning with task-specific tags, OpenGPT can be extended to the text classification. Unlike OpenGPT that predicts words based on previous predictions, there is another model that comes into use, i.e. BERT, intended to pre-train deep bidirectional representations from the unlabeled text by conditioning in all layers on both the left and right context together [57]. For text classification, BERT variants have been fine-tuned [155]. ALBERT decreases memory usage and improves BERT's training speed [156]. Another variant of BERT, SpanBERT [157], is a pretraining method developed to accurately signify and forecast lengths of text. It improves BERT by (1) masking consecutive random spans, rather than random tokens, and (2) training the span

boundary representations to estimate the entire content of the masked span without trusting on the individual token representations within it. Deep learning provides a way to manage massive volumes of processing and data with next to no engineering by hand [37]. Unsupervised learning has had a catalytic impact on the growing interest in deep learning, but the contributions of solely supervised learning have since been overshadowed. In the longer term, we expect unsupervised learning to become even more significant.

2.1.7. Evaluation

We have discussed several supervised and unsupervised text classification techniques based on machine and deep learning models so far; however, the shortage of uniform data collection procedures is a big issue when testing text classification techniques. Even if there is a standard collection method available, it can generate differences in model results by simply selecting different training and test sets [158]. Moreover, to compare various performance measures used during separate tests, there may be another difficulty related to process evaluation. In general, performance metrics assess attributes of the performance of the classification task and therefore do not necessarily present similar information. Although the underlying mechanics of various measurement metrics vary, it is important to consider precisely what each of these metrics describes and what kind of data they are attempting to express for comparability. Some examples of such performance measures include precision, recall, accuracy, micro-average, macro average, and F-measure. Such calculations depend on a “confusion matrix” composed of true negative, true positive, false negative, and false-positive [159]. Accuracy is considered the fraction of accurate predictions overall predictions. The fraction of known positives accurately estimated is referred to as recall. The fraction of positives accurately estimated for all positives is called precision.

2.1.8. Comparative Analysis

The following section summarizes the benefits and limitations of feature extraction, feature selection methods and supervised, and unsupervised machine and deep learning models used for a text classification task.

2.1.8.1. Comparative Analysis of Standard Text Representation or Feature Extraction methods

The two major feature extraction methods are highlighted: weighted terms and embedding of terms. By taking their frequency and co-occurrence details, term or word embedding approaches learn through sequences of words. These strategies are also unsupervised models to create word vectors. In comparison, the properties of weighted terms depend on the number of words in documents and may be used as a basic word representation ranking scheme. Each approach poses specific constraints. Weighted words explicitly quantify text similarities from the word-count space, which enhances the computational time for large vocabulary. Although counting unique terms offers independent confirmation of similarity, semantic comparisons between words are not taken into consideration [160]. Word embedding techniques solve this challenge but are constrained by the need for a large corpus of text data sets to train [161]. Therefore, researchers tend to use vectors with pre-trained word embedding [162]. Table 2.1 presents the advantages and limitations of each technique of text representation or feature extraction.

Table: 2.1 Benefits and limitations of Text Representation or feature Extraction method

Method	Benefits	Limitations
Bag-of-words	Works well with unseen words, easy to implement as it's based on most frequent terms in a document	Does not cover the syntactic and semantic relation of the words, common words impact on classification
TF-IDF	As like Bag-of-words approach, but common words are excluded due to IDF so doesn't impact the result	Does not cover the syntactic and semantic relation of the words

Word2vec	Covers the syntactic and semantic relation of the words in the text	Does not cover the words' polysemy
Glove	As same of word2vec but performs better, eliminates common words, trained on a large corpus	Does not cover the words' polysemy, does not work well for unseen words
Context-aware representation	Covers the context or meaning of the words in the text	Huge memory is required for storage, does not work well for unseen words

2.1.8.2.Comparative Analysis of Standard Feature Selection methods

For the Support Vector-based feature ranking model, several researchers have favored Fisher's linear discriminant [108][109]. The filter-based technique, according to the author, gives a distinguish feature selector that has further chosen unique features with term properties while eliminating uninformative ones [103]. It provided efficiency by reducing processing time and improving classification accuracy. Another common wrapper approach is the fast forward selection strategy, which is used to choose the best subset of a feature with less computing effort than other techniques [118]. A chaotic genetic algorithm was suggested in the genetic algorithmic approach to simplify the feature selection process and get greater classification accuracy [119][120][163]. Embedded methods were preferred over wrapper in many studies. Researchers mentioned that embedded methods have performed better than wrapper computationally, but these algorithms perform selection features as a sub-part of the learning technique. Further, these were followed by hybrid feature selection approaches where both filter and wrapper methods were combined, and these approaches were proven computationally effective than the performance of a single selection technique. It was observed by the researchers that sometimes original features may not be optimal dimensions for text document representation. They provided text feature extraction methods to solve the problem by creating new feature space or artificial terms [58]. Table 2.2 presents the benefits and limitations of feature selection methods.

Table: 2.2 Benefits and limitations of feature selection methods

Method		Benefits	Limitations
Univariate Filter method	Information Gain	Results into the relevance of an attribute or feature	Biased towards multi-valued attributes, Overfitting
	Chi-Square	Reduce training time, Avoid overfitting	Highly sensitive to sample size
	Fishers' Score	Evaluate features individually to reduce the feature set	Does not handle features redundancy
	Pearson's correlation coefficient	Simplest and Fast, Measures linear correlation between features	It is only sensitive to a linear relationship
	Variance Threshold	Removes features with variance below a certain cut-off	Does not consider the relationship with the target variable
Multivariate Filter method	mRMR (Minimal redundancy maximum relevance)	Measures the non-linear relationship between feature and target variable, Provides Low error accuracies	Features may be mutually as dissimilar to each other as possible
	Multivariate Relative Discriminative Criterion	Best determines the contribution of individual features to the underlying dimensions	Does not fit for small sample size
Linear Multivariate	Recursive Feature Elimination	Considers high-quality top-N features and removes weakest features	Computationally expensive and correlation of features not considered
	Forward/Backward stepwise selection	Computationally efficient, Greedy optimization	Sometimes impossible to find features with no correlation between them

Wrapper method	Genetic Algorithm	Accommodates Data set with a large number of features, knowledge about a problem not required	Stochastic nature, Computationally expensive
Non-linear Multivariate wrapper methods	Non-Linear kernel multiplicative	De-emphasizes the least useful features by multiplying features with scaling factor	The complexity of kernel computation and multiplication
	Relief	Feasible for binary classification, based on nearest neighbor instance pairs, Noise tolerant	Does not evaluate boundaries between redundant features, Not suitable for the low number of training datasets
Embedded methods	LASSO	L1 regularization reduces overfitting, can be applied when features are even more than data	Randomly selection when features are highly correlated
	Ridge Regression	L2 regularization, preferred over L1 when features are highly correlated	Reduction of features is a challenge
	Elastic Net	Better than L1 and L2 for dealing with highly correlated features, Flexible, Solves optimization problem	High Computational cost

2.1.8.3.Comparison of State-of-Art Machine and Deep Learning models for Text Classification

The performance of the classifier depends on the selection of feature selection and extraction method. The Supervised and Unsupervised machine learning techniques have offered a variety of classifiers that performed well in a variety of domain-specific classification problems. The following Table 2.3 presents their pros and cons. Recent studies have focused on deep learning or neural network-based classifiers like CNN, RNN, and RNN with LSTM which have shown better results than conventional algorithms like SVM and KNN in solving a different range of problems. RNN and LSTM have been used in many NLP applications due to their capacity of capturing complex patterns within the text[142]. It has also been used in sequence labeling tasks in POS (part of speech) activity. But it has offered a lot of future scope in resolving complexities involved in the backpropagation technique used in RNN and to make the learning model cheaper and faster [89].

Table: 2.3 Benefits and limitations of the machine and deep learning model

Model	Benefits	Limitations
Naïve Bayes	Need less training data, probabilistic approach handles continuous and discrete data, not sensitive to irrelevant features, Easily updatable	Data Scarcity can chance of loss of accuracy because it is based on assumption that any two features are independent given the output class
SVM	Possible to apply on unstructured data also like text, images, etc., Kernel provides strength to the algorithm, can work for high dimensional data	Needs Long training time on large datasets, Difficult in choosing good kernel function, choosing key parameters varies from problem to problem
KNN	Can be implemented for classification and regression problems, produces best results if large training data is available or even noisy training	Cost is high for computing distance for each instance, finding attributes for distance-based learning is quite a difficult task, Imbalanced data causes problems, no treatment for missing value

	data, Preferred for multi-class problem	
Decision Tree	Reduces ambiguity in decision making, implicitly performs feature selection, easy representation, and interpretation, requires fewer efforts for data preparation	Unstable due to the effect of changes in data requires changes in the whole structure, not suitable for continuous values, Causes overfitting problem
Random Forest	In contrast to other methods, clusters of decision trees are very easy to train, the preparation and pre-processing of the input data does not require	More trees in random forests increase the time complexity in the prediction stage, high chances of overfitting to occur
CNN	Fast predictions, best suited for a large volume of data, no human efforts required for feature design	Computationally expensive requires a large dataset for training
RNN	Implements feedback model so considers best for time series problems, makes accurate predictions than other ANN models	Training of model is difficult and takes a long time to find non-linearity in data, gradient vanishing problem occurs
LSTM, Bi-LSTM	Adds short-term and long-term memory components into RNN so it considers best for applications that has a sequence, uses for solving NLP problems like text classification and text generation, computing speed is high. Bi-LSTM solves the	Expensive and complex due to the backpropagation model increases the dimensionality of the problem and makes it harder to find the optimal solution. Since Bi-LSTM has double LSTM cells so it is expensive to implement.

	issue of predicting fixed sequence to sequence.	
Transformer with an attention mechanism	The issue with RNNs and CNNs is that when sentences are too long, they are not able to keep up with context and content. By paying attention to the word that is currently being operated on this limitation was resolved, the Attention Strategy is an effort to selectively concentrate on a few important items while avoiding those in deep neural networks to execute the same operation, enabling much more parallelization than RNNs and thus reduces training times	At inference time, it is strongly compute-intensive

2.1.9. Hybrid Approach

Both knowledge-driven and data-driven techniques have their own set of drawbacks. Previously, moving an established set of extraction rules to a new domain required a lot of high-quality expert manual effort and was complex. The latter demands a big corpus annotation and a similar level of manual work, and their results are more difficult to interpret. Research [59] investigates a hybrid text classification technique that combines a machine learning algorithm that generates a basic model trained on a labeled corpus, as well as a rule-based expert system for filtering false positives and dealing with false negatives to improve the previous classifier's results. The authors also look into a hybrid method that combines framing rules with machine learning [164]. However, the number of contributions is still limited, and it does not address all of the issues.

2.1.10. Techniques for dealing with Imbalanced data in Multi-class Text classification

We will illustrate in this section the various techniques which have been experienced so far by researchers for training a model to perform well against highly imbalanced data sets. The authors mentioned that where it comes to text classification, the normal distribution of textual data is often unbalanced. To better differentiate documents into minor categories, they used a basic probability-based word weighting scheme to solve the problem [165]. Many real-world text classification tasks, according to the authors, require unbalanced training instances. However, in the text domain, the methods introduced to resolve the imbalanced description have not been consistently tested. They conducted a survey based on the taxonomy of strategies suggested for imbalanced classification, such as resampling and instance weighting, among others [166]. The following sub-sections cover the literature of various techniques used so far to deal the text classification with imbalanced data sets.

2.1.10.1. Data Level technique

Dealing with imbalanced data sets requires techniques such as enhancing classification algorithms or balancing the training data classes until the machine learning algorithm provides the data as input. The main objective of class balancing is to increase the frequency of the minority class while decreasing the frequency of the majority class. This is provided for all classes to get equal instances.

Under-sampling aids in optimizing class allocation by randomly eliminating instances of majority classes. This is achieved when the majority and minority class cases are balanced completely. Evolutionary undersampling outperforms the non-evolutionary models by increasing the degree of imbalance [167]. They describe a performance function that incorporates two values: the classification factor aligned with both the sub-set of training instances and the percentage of reduction associated with the training set of the same sub-set of instances. A novel under-sampling technique is implemented, called cluster-based instance selection, which incorporates clustering analysis with instance selection [168]. The clustering analysis framework groups identical data samples of the majority class dataset into subclasses, while the instance selection framework extracts out unaccountable data samples from each subclass. It is also proven that under-sampling with KNN is the most powerful approach [169].

Over-sampling raises the amount of minority class instances by arbitrarily replicating them to make the minority class more represented in the study. The author suggests a Random Walk Over-Sampling method by generating synthetic samples by randomly walking from real data to match different class samples [170]. This sampling method is designed to address the imbalanced grouping of data by producing some samples of a synthetic minority class. The synthetic samples, that properly follow the initial minority training set and extend the minority class boundaries, are coupled with the actual samples to make a more efficient full dataset, and the entire is used to build unbiased classifiers. Unfortunately, traditional over-sampling approaches have shown their respective shortcomings, such as causing serious over-generalization or not effectively improving the class imbalance in data space, while facing the more challenging problem as opposed to the binary class imbalance scenario. The author proposes a synthetic minority oversampling algorithm based on k-nearest neighbors (k-NN), called SMOM, for handling multi-class imbalance problems [171]. SMOM is a method to prevent over-generalization since safer neighboring directions are more likely to be chosen to produce synthetic instances. It is also suggested that combine sampling be rendered by combining the techniques of SMOTE and Tomek with SVM as the method of binary classification [172]. SMOTE is a useful over-sampling technique for increasing the number of positive classes incorporating sample drawing methods by replicating the data randomly so that the number of positive classes is equal to that of the negative class. [173] has performed multiclass classification with equal distribution of the data among various classes using SMOTE, owing to the introduction of synthetic instances which increased the number of training samples to distribute the data equally among 10 different labels. Tomek links method is under-sampling, which works by decreasing negative class numbers. However, in some extreme cases mixing sampling methods are no stronger than utilizing Tomek link methods.

2.1.10.2. Algorithms-based decomposition techniques

The technique must first use decomposition strategies to transform the original multi-class data into binary subsets.

One-vs-all is a strategy that requires training N independent binary classifiers, each programmed to identify a specific class. All those N classifiers are collectively used to classify multiple classes. With multi-class imbalanced data, an algorithm called One-vs-All with Data Balancing (OAA-DB) is built to enhance the classification performance [174]. It is mentioned that the OAA-DB algorithm can boost classification efficiency for imbalanced multi-class data

without decreasing the overall classification accuracy. In other words, for every class, One-vs-All trains a single classifier, treating the existing class as the minority one and the remaining classes as a majority.

One-vs-One trains a binary classifier for each potential pair of classes, ignoring examples that are not part of the pair classes. To resolve the multi-class imbalance classification problems, an exhaustive empirical study is proposed to investigate the possibility of improving the one-vs-one scheme through the application of binary ensemble learning approaches [175].

One-Against-Higher-Order (OAHO) is an explicitly designed decomposition process for unequaled sets of data. OAHO first divides class by decreasing the number of samples (Bi and Zhang 2018). OAHO sequentially marks the current class as 'positive class' and all the remaining classes with lower ranks as 'negative classes,' then trains a binary classifier.

All-in-One uses One-vs-All along with One-vs-One, it first uses One-vs-All sub-classifiers to find the top two most probable categories for each test case, and then uses the corresponding One-Vs-One sub-classified to decide the final result [176].

2.1.10.3. Algorithms-based Ensemble techniques

The main purpose of the ensemble methodology is to improve single classifier efficiency. The method involves constructing from the original data numerous two-stage classifiers and then aggregating their predictions.

Boosting-based techniques

One strategy which can be used to increase classification efficiency is boosting. Although several data sampling techniques are explicitly developed to fix the issue of class imbalance, boosting is a technique that can increase the efficiency of any weak classifier. Ada Boost iteratively constructs a model ensemble, which is an adaptive boosting strategy that combines many weak and inaccurate rules to build a predictive rule that is highly effective. During each iteration, case weights are changed to properly classify the instances in the next iteration that were wrongly classified during the current iteration. Upon completion, all models were developed to take part in a weighted vote to identify unlabeled cases. Such a strategy is especially useful when grappling with class inequality as in successive implementations the minority class instances are more likely to be misclassified and thus assigned larger weights. In other words, it's a binary classification algorithm that combines many weak classifiers to

create a stronger classifier [177]. Boosting can be achieved either by "reweighing" or "resampling". At each step, the changed example weights are transferred directly to the base learner while boosting by reweighing. Not all learning algorithms are designed to integrate example weights into their decision-making systems, however. This is a class that uses the AdaBoost ML method to boost a nominal classifier which can only address nominal class problems.

$$f(x) = \text{sign}(\sum_{m=1}^M \theta_m f_m(x)) \quad (2.21)$$

Here, $f(x)$ represents m^{th} weak classifier and θ_m is the corresponding weight.

This often improves the performance dramatically but sometimes overfits [178]. Gradient boosting is an approach that generates a set of weak regression trees by introducing iteratively a new one which further strengthens the learning goal by optimizing an arbitrary differentiable loss function [179]. Gradient Boosting builds the first learner to predict the samples on the training dataset and calculates the loss. And use that loss in the second stage to build an improved learner. The recent implementation of this boosting method called XGBoost combines the principles of computational efficiency. The paper presents a scalable end-to-end tree boosting system XGBoost that is widely accessed by researchers to perform "state-of-the-art" machine learning outcomes [179].

Bagging-based techniques

Bootstrap aggregation, also known as bagging, is an ensemble meta-algorithm for machine learning that aims to enhance the stability and accuracy of classification algorithms. The standard algorithm requires the development of specific bootstrap training items, 'n' with substitution. Then train the algorithm on each bootstrapped algorithm separately, then aggregate the forecasts at the end. The authors present online bagging and boosting versions that require only one pass through the training data [180]. Random Forests is an ensemble classifier composed of several decision trees and generating the class which is the class output mode for individual trees. In this way, an RF ensemble classifier works more than a single tree from the classification results perspective [181]. The authors suggested ensemble classifiers focused on actual principles like learning cluster boundaries by the base classifiers and mapping cluster confidences to a class decision using a fusion classification [182]. So, a classified dataset is divided into several clusters and is put into several distinctive base classifiers. Cluster boundaries are identified to base classifiers and cluster confidence vectors

are built. A second-stage fusion classifier blends class decisions with confidences and maps of the clusters. This ensemble classifier restructured the learning environment for the base classifiers and promoted successful learning.

Other Techniques

Despite their effectiveness, however, sampling methods add complexity and the selection of required parameters. To address these problems, the author suggests a modern decision tree strategy named Hellinger Distance Decision Trees (HDDT), which allows the use of distance from Hellinger as the criteria for splitting. For probability and statistics, the Hellinger distance is used to measure the correlation of two distributions of probabilities. The authors use a Hellinger weighted ensemble of HDDTs to combat definition drift and improve the accuracy of single classifiers [183].

Error Correcting Output codes, ECOC is a common multi-class learning tool that works by breaking down the multiclass into a set of binary class subtasks (dichotomies) and creating a binary classifier from each dichotomy. Both the dichotomy classifiers evaluate a test instance and then assign it to the closet class in code space. A suitable code matrix, an effective learning strategy, and a decoding strategy highlighting minority classes are needed to enable ECOC to tackle multi-class imbalances. The authors propose the imECOC approach that operates on dichotomies to deal with both the imbalance between class and the imbalance within a class [184]. ImECOC assigns dichotomy weights and uses weighted decoding distances where optimum dichotomy weights are derived through reducing weighted loss in terms of minority classes.

The authors suggest merging weighted One-vs-One voting with a Winnow dynamic combiner customized to the program for the data stream. This will allow weights for classifiers to be dynamically modified, boosting the power of those competent in the current state of the stream [185]. DOVO simply adjusts the weights for classified objects returned via an active learning approach that enables even more consistent weights and lower processing costs. From those in the perspective of operation recognition, each action shall be taken over a given period. The proposed weighting procedure thereby enables to rapidly increase the significance of qualified classifiers to identify this particular behavior immediately after it has been identified by the active learning methodology and to sustain the significant importance of these related classifiers throughout its length.

2.1.10.4. Existing solutions or software for classification with imbalanced datasets

A program, KEEL [186], provides a customized algorithm for the problem of classification with class imbalances. Multi-IM draws its basis from the probabilistic relational methodology (PRMsIM), developed to learn from imbalanced data for the problem of two categories [187]. Imbalanced-learn; A Python toolbox for resolving imbalanced results [188].

2.2.Text Sentiment Classification

Text sentiment classification is a rapidly expanding application of natural language processing (NLP) that is now widely used in various commercial domains, covering social media, customer experience, algorithmic trading, and human resource management. The research examines the Central Bank of Nigeria's communication policy using text analysis methodologies [189]. Their findings show that the Bank's policy aims, as well as the intensity of economic shocks per period, influenced the problems impacting the linguistic content of the communications.

According to a review of the literature, many of the earlier works dealt with pieces of business news covering stock market-related events and corporate organizations. Economists acknowledge that financial markets can display unusual behavior in reaction to socio-political events, acts of terrorism, or catastrophic events, among other things. [190].

Views and sentiments in the news domain can be examined due to the exponential growth of newsgroups. Authors employ sentiment classification algorithms in the field of political news from columns on many Turkish news websites in comparable research [191]. The authors propose that the inclusion and intensity of emotive words be used to define the sentiment of stock market news reports [192]. While the authors believe that stock or financial markets can react to the sentiment of financial news, no systematic attempt has been made to regulate and use unique banking events for automated sentiment prediction. To the best of our knowledge, no systematic effort is made to examine the sentiments of financial events that are reported on numerous internet news wires. This might be due to the inability to compile a list of all probable banking-related events that could affect stock markets.

2.3.Named Entities Recognition

Many breakthroughs in Natural Language Processing have occurred in recent years in the NLP field, particularly the transition to transfer learning. Researchers have been able to obtain state-of-the-art results on numerous benchmarks due to models like ELMo, ULMFiT, Transformer, and OpenAI's GPT, which have given the community huge pre-trained models with excellent performance. Rather than training new networks from start, lower layers of deep learning networks with millions of parameters learned on a single task may be reused and fine-tuned for different tasks, marking a change in NLP.

The introduction of Google's BERT [57], which has been regarded as the start of a new era in NLP, is one of the most significant milestones in the advancement of NLP recently. The HuggingFace transformers library in python is used to fine-tune a pre-trained BERT model for identifying named entities in banking news events. We just need to fine-tune BERT in 2-4 epochs, although it is quite huge, sophisticated, and has millions of parameters. Because BERT was trained on a large quantity of data and has already encoded a lot of information about our language, this outcome is possible. BERT is considered the most powerful NLP model accessible right now, because of its remarkable performance obtained in less time with a little quantity of data.

The CoNLL Dataset [193] is used to demonstrate Named Entity Recognition. The input text files for our datasets are in the CoNLL format. In CoNLL datasets, there are 9 classes of NER tags:

O, Outside of a named entity

B-MIS, Beginning of a miscellaneous entity right after another miscellaneous entity

I-MIS, Miscellaneous entity

B-PER, Beginning of a person's name right after another person's name

I-PER, Person's name

B-ORG, Beginning of an organization right after another organization

I-ORG, Organisation

B-LOC, Beginning of a location right after another location

I-LOC, Location

2.4.Event Study

In the stock market, certain scheduled and unscheduled events may influence the company's financial performance. The literature on fraud-related events shows that fraud harms a company's profitability and shareholder value [13]. The objective of another research is to show that bank corporate governance news influences market returns [194]. However, the authors mentioned that bidders in stock mergers generate much more news articles after the commencement of merger discussions but before the public announcement. They also show how companies' attempts to affect their stock price can skew the timing and substance of financial media reporting [195]. They believe that companies create and broadcast news to the media to affect the price of their shares during major business events. So, the researchers need to be more proactive pre and post events news publications or announcements. While the study of the press in political economics models has been ignored in the past, there is currently a growing collection of work on the subject. The model evaluates its empirical relevance and reconciles it with what is known about the media's role in promoting political accountability [196].

Algorithmic trading has developed as a major trend in financial portfolio management in recent years, because of advancements in processing power and the capacity of computers to analyze enormous quantities of such news events [197]. This, in combination with developments in the machine or deep learning and natural language processing (NLP), has propelled text data as a source of information for investing strategies. So far, sentiment analysis has been the most influential area of NLP in the stock market analysis [50]. They examined the relationship between the tone of news articles and market prices using sentiment analysis, indicating that media negativity can influence both trade volume and market prices

Many techniques have been proposed in the literature based on feature selection for stock market analysis, according to news articles. The authors created a method to assess sentiment in financial news articles by extracting features from noun phrases and scoring them on a negative/positive subjectivity measure [198]. The authors improved accuracy by using a selection process based on a contextual entropy model that discovered comparable emotion words and their accompanying strengths from online stock market news events to extend a collection of seed words [192]. Using Chi-square feature selection with n-gram features, the authors also got good results [199]. An evolving algorithm was utilized to integrate trading rules utilizing technical indicators and events derived from the news with expert-defined

impact scores to incorporate news into stock trading strategies [200]. They derived events from news messages that are given in plain text format with no annotations. They made use of the ViewerPro system based on a domain-specific knowledge repository, which is an ontology that contains attributes and lexical representations of financial entities (firms) and news events.

Financial market analysis has also been approached using deep learning approaches. They evaluated that the deep learning model was more accurate than the SVM model. Following that, they utilized a neural tensor network to learn word and structured event embeddings as inputs to their stock market analysis models [201]. Following this research, the authors coupled recurrent and convolution layers with pre-trained word vectors to forecast changes in the S&P 500 index [202]. Following deep learning approaches for natural language processing (NLP), the authors of a recent paper propose using bidirectional encoder representations from BERT transformers to do sentiment analysis of news items and give useful information for stock market decision-making. A self-learning task is used to pre-train this model on a large number of general-domain texts. They manually categorized stock news items as positive, neutral, or negative to fine-tune their model on sentiment analysis for the stock market [203].

2.5.Summary

This chapter covers event extraction and classification, sentiment classification, NER, and event study using “state-of-the-art” models. The multi-class text classification with data imbalances is the central focus of this chapter. The literature is divided into four sections, which include work on event extraction and classification systems, sentiment classification systems, named entity recognition using various Rules-based, machine learning, deep learning, and hybrid models, and event study using statistical techniques. In addition, studies on the basics of popular text classification models are presented. The chapter also mentions the challenges of multiclass text classification with data imbalances and details the solution for each challenge. Moreover, the literature on event-driven stock market analysis with machine learning and statistical techniques is also discussed.

Chapter-3

Research Methodology

This chapter presents the proposed hybrid model for ‘event extraction and classification’ from the banking and other correlated news. Initially, the banking and other most associated news are isolated from the pool of financial news articles scraped from various web news portals. The events are further classified into negative, positive, and neutral sentiments followed by the event study to analyze the influence on stock market returns. And the econometric model is used to explore the correlation between the events concerning sentiments and the individual banks’ stock, identified or recognized from these news events, or the banking index returns against the benchmark index in the Indian stock market.

The research topic is divided into the following phases:

1. In phase-1, we’re interested in isolating news on the banking sector and its most associated domains from the pool of financial news articles. We feel that ‘banking news’ of any nation is most correlated with their ‘governmental news events’ that covers news on government initiatives for good governance, state or national elections, change or new development of governmental policies, and ‘global’ financial news that covers global trade, changes in currency-commodities prices, and global sentiments. So, we have a 4-class classification problem of a set of news articles to extract banking, and its most correlated news i.e. Government, and Global from entire financial news articles. As per objective-1, news-text documents on the ‘banking sector’ become the in-domain data, and the news-text documents on the ‘governmental’, and ‘global’ become the out-domain data.
2. In phase-2, we present a hybrid model in which a rule-based method is combined with a machine learning algorithm for extracting and classifying financial news from the Indian banking domain into appropriate pre-defined seven-event categories.
3. In phase-3, the study comprises news events for sentiment classification on Results, Fraud, Merger or Acquisition, RBI Policies, Global, Governmental and Ratings Agencies, or Expert’s View.
4. In phase-4, we analyze the impact of news events concerning sentiments or linguistics in the price movement of the respective bank’s stock or banking index, identified or recognized from these news events, against benchmark index Bank Nifty and the banking index against benchmark index Nifty50 for the short to long term. For the short

term, we have analyzed the movement of banking stock or index to benchmark index in terms of CARs (cumulative abnormal returns) surrounding the publication day of the news event in the event windows of $(-1, D)$, $(D, 1)$, $(-1, 1)$, $(D, 5)$, $(-5, -1)$, and $(-5, 5)$. For the long term, we have analyzed the movement of banking stock or index to benchmark index in the event windows of $(D, 30)$, $(-30, -1)$, $(-30, 30)$, $(D, 60)$, $(-60, -1)$ and $(-60, 60)$.

3.1. Phase-1: Banking News Extraction by Multiclass Classification with Imbalanced Datasets of Financial News: Challenges and Solutions

Text classification is crucial for information extraction and summarization, text retrieval, and question-answering in general. Using machine learning algorithms, the authors demonstrated the text classification process [204]. Following the approach, we developed a structure shown in Fig. 3.1 to distinguish the banking and other related sector-oriented news items from financial news posts. However, when faced with imbalanced multi-class results, we can drop output on one class quickly when attempting to get output on another class. A clearer analysis of the essence of the issue of class imbalance is required, as one should recognize in what realms class imbalance most impedes the output of traditional multi-class classifiers while developing a system suitable to this topic. Although most of the problems addressed in the preceded section can be applied to these multi-class concerns, the banking and other related news extraction from the financial news domain.

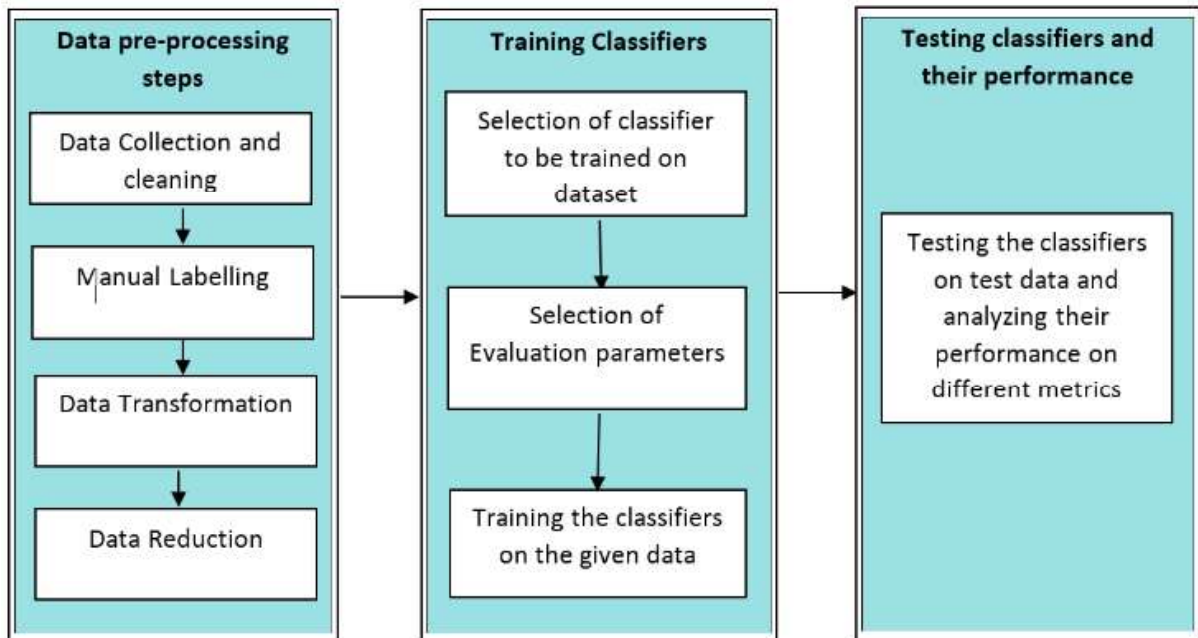


Fig.3.1: Multiclass classification of Financial News

3.1.1. Data pre-processing

Data preprocessing is a method used to transform the raw data into an effective and functional format. Effective pre-processing of text data is critical to achieving an appropriate output and better text classification quality [96].

Challenge-A: The task of preprocessing data here may be much more critical than in the case of binary issues. Possible difficulties can be easily identified: class overlap can occur in more than two classes, class label noise can influence the issue, and class boundaries may not be specific. Therefore, effective data cleaning and sampling techniques must be implemented to take into consideration the various characteristics of the classes and the balanced performance of all of them [205].

Solution-1: The problem of noise present in the data in the case of imbalanced distributions is incredibly difficult. Distortions may dramatically deteriorate classifier efficiency, particularly in the case of minority examples. New data cleaning methods need to be used to manage the existence of overlapping and chaotic samples which can also lead to worsening efficiency of the classifier. We might conceive projections into different spaces where the overlap is alleviated or basic examples are eliminated. However, measures are needed to assess whether a provided overlapping example can be excluded without discrimination to one class. A study of the effect on the real imbalance between classes is quite important in the case of label noise. Measures are therefore required to determine whether a given overlapping example can be discarded without compromising one of the classes. False labeling may lead to increasing the imbalance or disguise actual disproportions. This situation is handled with sustained methods for sensing and filtering noise, as well as handling and relabeling strategies for such examples.

Solution-2: Analysis of the kind of examples found in each class and their connections with other classes is interesting. Measuring each sample's difficulty here isn't straightforward, as it may adjust to various classes. For instance, for classes Banking and Governmental, news related to a collective decision on negative GDP outlook and modification on repo rate by RBI may be of borderline type while at the same time being a safe example when considering the remaining classes. Therefore, we have preferred a more flexible classification i.e. SMOTE. SMOTE functions by choosing similar examples in the vector space, drawing a line through the examples in the vector space, and drawing a new example at a point in the line.

Solution-3: New sampling approaches are needed for issues of multiple classes. Simple re-balancing is not a proper approach towards the largest or smallest class. We need to establish precise methods for adapting the sampling procedures to both the individual class property and

their mutual relationships. [206] has provided the ensemble methods to deal with class imbalance classification, ADASYNBagging, and RSYNBagging. The ADASYN and RSYN were based on over-sampling and under-sampling techniques respectively. These were combined with a bagging algorithm to integrate the advantages of both algorithms. Another paper has provided a hybrid model to get a random sample from an unknown population. When compared with a random sample, a non-random sample could not provide better representative inferential statistics. Hence, to overcome this problem, Snoran Sampling Method was developed by [207]. We have not implemented these techniques in our paper. What sampling strategies would function best with the learning of the ensemble to boost class inequality, however, is highly dependent on problem domains.

3.1.1.1. Data collection

To continue this, using python-written code, we scraped news from public news sources such as Bloomberg, Financial Express, Money Control, and Times of India. Web crawler programmes are frequently employed to solve the problem of automatic online news monitoring. These programmes make use of the Web’s graph structure by beginning at a seed page and then traversing the hyperlinks to subsequent pages. This procedure is repeated until a certain number of pages are retrieved or a higher level objective is met. As a result, from 2017 to 2020, we have gathered almost 10,000 instances of financial news articles. The news items are from different sectors or market areas. These are then pre-processed so that machine learning algorithms may learn from the training dataset and adapt to the testing data collection reasonably. As a result, these are pre-processed for machine learning models to be investigated from the training sample and applied to the test data set acceptably.

3.1.1.2. Labeling

The first step in the pre-processing phase is to *label* the news from 4 classes to which they belong to the specific sector.

Table 3.1: Sample of news articles from different sources and classes

Source	News article	Class
Source1 ¹	“After floating India’s largest bank IPO earlier this month, the Kolkata-based private sector lender Bandhan Bank exceeded the	Banking

¹ www.financialexpress.com

	market capitalization of all listed PSU banks except State Bank of India on its blockbuster stock market debut on Tuesday.”	
Source2 ²	“The current account deficit in India is within acceptable bounds, even though it has expanded, and GDP growth is expected to reach 7.5-7.7%.”	Governmental
Source3 ³	“The Federal Reserve of the United States has lowered its benchmark interest rate by a half-point, the first time it has done so outside of scheduled meetings since the financial crisis of 2008.”	Global
Source4 ⁴	“For the sixth day in a row, the Nifty50 has produced a bearish candle, and analysts believe it would be difficult for the index to quickly surpass the 200-DEMA.”	Non-Banking

4-classes are named Banking, Global, Governmental, and Non-Banking. We prefer manual labeling [208] of the news articles with the help of experts of the financial domain where overlapping examples were preferred to discard without damaging one of the classes. Table 3.1 mentions the instance of each class as mentioned above.

3.1.1.3. Data Cleaning

They are then *cleaned* because the data can have several sections that are insignificant and missing. Data cleaning is done to handle that portion. It includes absent managing data, noisy data, etc. It helps the machine learning algorithms to efficiently grasp and operate on them.

3.1.1.4. Data Transformation

The next step, *data transformation*, is taken to turn the data into appropriate forms suited to the mining process, and the text of news articles therein is converted into measures with quantitative values by constructing a vector *set of features*. Since data mining is a technique used for managing enormous quantities of data. In these instances, research became harder when operating with a huge volume of data. To get rid of that, we use the strategy of *data reduction*. This seeks to increase the capacity of storage and reduce the expense of data

² www.moneycontrol.com

³ www.bloombergquint.com

⁴ www.moneycontrol.com

collection and analysis. In other words, in the last step of this stage, the feature vector is normalized and scaled to prevent an *unbalanced dataset*.

3.1.2. Training classifiers

Training is the practice of having text that is considered to belong to the defined classes and creating a classifier based on that known text. The basic concept is that the classifier accepts a collection of *training data* describing established instances of classes and uses the information obtained from the training data to determine the classes other unknown content belongs to, by conducting statistical analysis of training data. We can also use the classifier to derive information on your data based on the statistical analysis carried during the training process. First, we identify the classes on a collection of training data, and then the classifier uses these classes to evaluate and decide the classification of other data. When the classifier assesses the data, it uses two often contradictory metrics to help decide if the content found in the new data belongs in or outside a class. *Precision*, the likelihood that what has been labeled as being is actually in that class. High precision may come at the cost of missing certain results whose terms match those of other outcomes in other groups. *Recall*, the likelihood that an object is listed as being in that class in fact in a class. High recall may come at the cost of integrating outcomes from other classes whose terms match those of target class results. We need to find the right balance with high precision and high recall while we are tuning our classifier. The balance focuses on what our priorities and criteria are for implementation. We need to train the classifier with *sample data* that describes members of all the classes to find the best thresholds for our data. Finding good training samples is very critical because the nature of the training can directly influence the quality of the classification. The samples should be statistically valid for each class and should include samples that include both solid class examples and samples near the class boundary.

Challenge-B: The strong potential resides in the complexity of multi-class, distort-insensitive classifiers. They will permit multi-class complications to be handled without referring to strategies for resampling when algorithm-level approaches are used to counter class imbalances. So one may wonder if other prominent classifiers can be adapted in this case [205].

Solution-1: Certain issues should be addressed when constructing multi-level classifiers in the case of class imbalances. A broader study is required of how numerous unbalanced data sets influence decision boundaries in classifiers. Based on [209] Hellinger distance has proved useful in cases of class imbalance. Since accuracy may offer a distorted picture of success on

unbalanced data, current stream classifiers are focused on accuracy that is hampered by minority class output on unbalanced streams, resulting in low recall levels of minority classes. A split based on Hellinger Distance will give a high score to a split separating the classes in the best way relative to the parent population. When utilizing Hellinger, it is possible to obtain a statistically relevant change in the recall level on imbalanced data sources, with a reasonable rise in the false positive rate.

Solution-2: Other solutions with potential robustness to the imbalance, such as methods based on density, need to be explored. [210] have provided a more thorough review of the cluster oversampling based on density and in terms of density-dependent clustering under-sampling techniques. Their findings suggest the strategy will boost the classifier's predictive efficiency. It also yields the best in the precision average.

Solution-3: While modern methods of learning with imbalances are suggested to tackle the question of data imbalances, they have certain limitations; under-sampling methods lose essential details, and cost-sensitive methods are prone to outliers and noise. [211] has provided an efficient hybrid ensemble classifier architecture incorporating density-based under-sampling and cost-effective approaches by investigating state-of-the-art solutions using an algorithm for multi-objective optimization. First, they developed a density-based under-sampling method to select informative samples with probability-based data transformation from the original training data, which enables multiple subsets to be obtained following a balanced class-wide distribution. Second, they have used the cost-sensitive approach of classification to address the problem of information incompleteness by modifying weights in minority groups misclassified, rather than majority samples. Finally, they implemented a multi-objective optimization method and used sample-to-sample relations to auto-modify the classification outcome utilizing an ensemble classification system.

3.1.3. Testing classifiers and their performance

We run the trained classifier on unknown news articles to check a classifier to decide which classes each news article belongs to. The goal of this stage is to check the performance of the classifiers on the training set and to see if they detect the training correctly. The classifiers considered will be graded according to their effectiveness in detecting the appropriate class. In the later section, we will test various classifiers on the unseen news articles and compare the performance of each.

3.2. Phase-2: Banking news-events classification with a novel hybrid model using DistilBERT and rule-based features

The process of recognizing and extracting structured information about events in any text and assigning the right label is known as event extraction and classification. Because of the thoroughness of the collected data and its applicability to a variety of real-world situations, it has attracted a large number of researchers and businesses. Furthermore, a growing number of ways to enhance the performance of event extraction systems and the quality of derived information from rule-based approaches to machine learning techniques have been presented. It is stated that the machine learning approach usually produces errors while extracting de-verbal entities that signify events [164]. This discovery motivates researchers to use a variety of approaches in conjunction with machine learning. We suggest a hybrid approach in this research that combines these two approaches to get superior performance by utilizing each. The suggested hybrid system's architecture for event extraction and categorization from news articles, with a focus on banking news events, is in Fig. 3. 2.

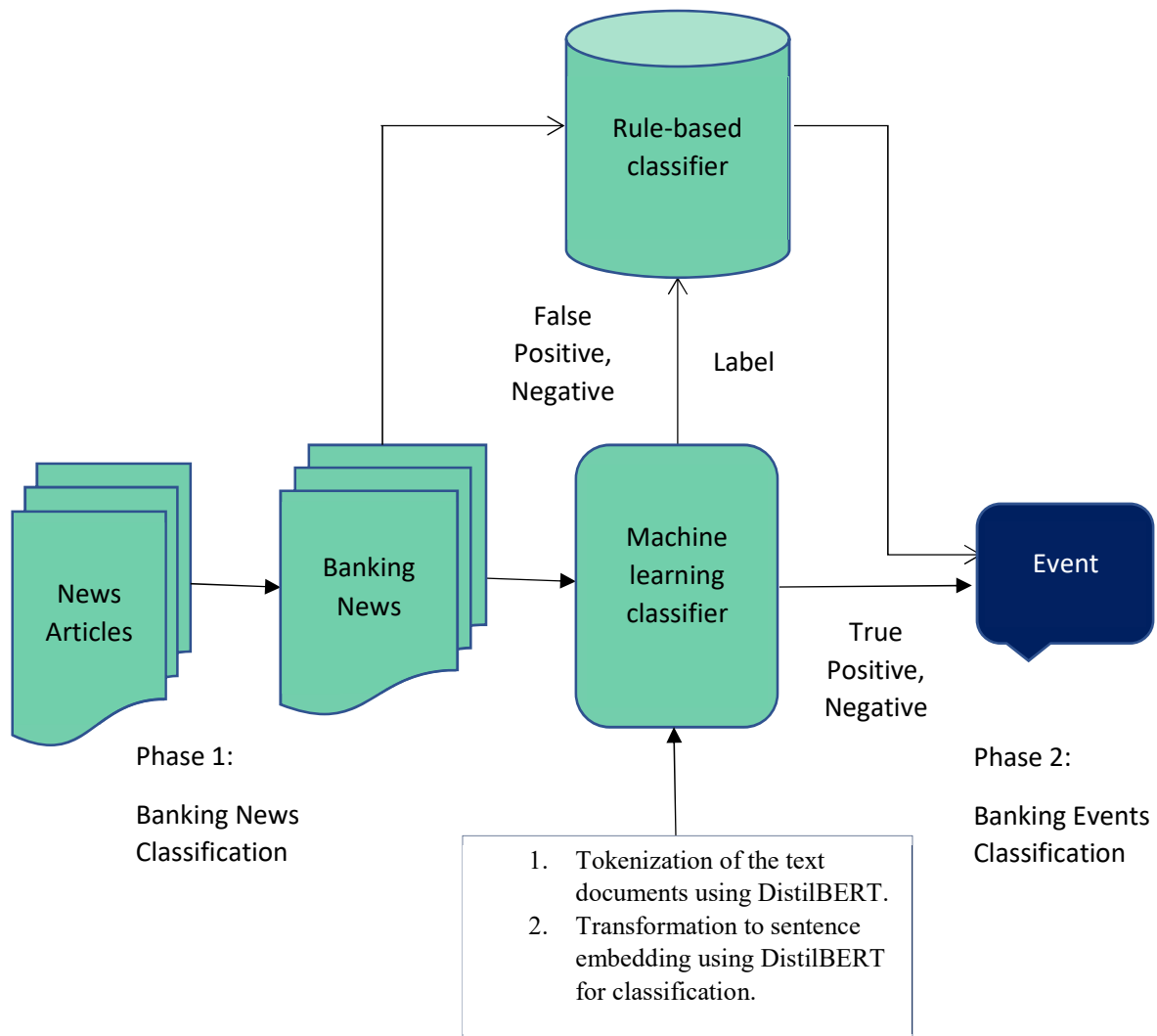


Fig. 3.2: Steps for News-Events representation and classification using a hybrid approach of Rule-based and machine learning model based on DistilBERT

Our research involves an experimental assessment of Indian banking news for event extraction and classification utilizing a hybrid technique that uses a rule-based approach in conjunction with a machine-learning algorithm to discover event extent and event triggers. A python script was used to monitor and scrape unstructured web news, and a set of news articles was chosen to classify documents suited for the banking domain, such as documents reporting financial events. We're interested in dealing with seven events that occur sometimes in the Indian banking system.

Our method for events classification involves three steps: (1) Representation of news articles using DistilBert; (2) Fine Tuning DistilBert along with machine learning models for classification; (3) Validating the result of step 2 with Rules for false positives and false negatives.

3.2.1. DistilBERT for banking news-events representation

Because of the enormous unlabeled corpus, embedding is one of the most promising applications of both unsupervised and transfer learning. There was a great deal of demand for transfer learning. There were some advantages of the transformer-based model. The benefits are twofold: (1) Instead of producing an input sequence token by token, these models accept the entire sequence as input all at once, which is a significant improvement over RNN-based models, and the GPUs will now help the model run faster. We don't require labeled data to pre-train these models. This means that all we need to train a transformer-based model is a huge volume of unlabeled text data. This transfer learning model was utilized for the NLP problem of News-Events classification.

Overfitting occurs when a BERT model is trained from scratch on a short dataset. As a starting point, a pre-trained BERT model trained on a big dataset is preferable. We may next refine the method using our comparably smaller dataset, a process known as model fine-tuning. However, while transfer learning via large-scale pre-trained systems is becoming increasingly popular in Natural Language Processing, running these big models with limited CPU training or inference budgets remains a challenge. The authors provide a method for pre-training DistilBERT, a general-purpose language representation model that may be fine-tuned for a range of purposes [56]. However, the tendency toward larger models in terms of parameter

count raises certain concerns as shown in Fig. 3.3. The first is the environmental cost of expanding these models' computational needs. Second, while real-time execution of these on-device models has the potential to enable new and interesting language processing applications, the increased CPU and memory needs of these models may limit their widespread use [212].

The findings showed that a 40% smaller Transformer pre-trained by distillation under the supervision of a larger Transformer (based on BERT) language model may achieve comparable efficiency while being 60% quicker at inference time on a range of downstream tasks.

3.2.2. Machine learning classifiers

Different classifiers are used to compare the classification efficiency in detail. Logistic regression is one of the most essential and widely used machine learning techniques. When learning predictive modeling, one of the first few subjects that people select is logistic regression. Logistic regression is a model for probabilistic classification, not a regression technique. The authors apply the model to a range of document classification tasks and show that it produces compact prediction models that are at least as accurate as those generated using SVM and feature selection [112]. *Linear SVC* is another machine learning method that may be used to solve the NLP problem. A Linear SVC seeks to match the data we have and returns a “best fit” subspace that divides or categorizes it. Support Vector Classifier was used to create the text classification application, which has a high level of accuracy [111].



Fig. 3.3: Several pre-trained models with parameters count [56]

The *Decision Tree* classifier separates data depending on the state of data points using a hierarchical decomposition of training data. Separation generally necessitates the inclusion or

exclusion of words. Every non-leaf node has an attribute feature connected with it, and every leaf node has a positive or negative classification score [213]. The decision tree is frequently used in research, and the rules it generates are simple to understand. The rules that were developed, on the other hand, are skewed against definite features. In a cost-sensitive case, there exist still various small branches in the tree, and the overfitting issue is severe. Furthermore, because the decision tree consists of only one root, the generated rules are typically necessary locally, however, useless globally. To find efficient rules, the author constructs a decision forest that comprises several powerful decision trees. On unbalanced data, this performs superior to standard decision tree approaches in building a prediction algorithm [84]. *Random Forest* is a powerful and easy-to-understand machine learning method that uses random sampling and ensemble techniques to provide precise predictions for a variety of datasets. The interpretability of the model and the performance of Random Forest are quite unusual among machine learning methods in terms of prediction accuracy. Random Forest, on the other hand, is less prone to outliers in training outcomes, trimming the trees is not required because the bootstrapping and ensemble approach efficiently solve the concerns of overfitting. Random Forest covers all the benefits of decision trees and, as a result of the use of bagging on samples, random subsets of variables, and voting systems, it often provides superior results [5]. The performance of each classifier is discussed in the upcoming section.

3.2.3. Fine-tuning with rules

A rule-based technique is generally made up of a series of if-then rules that are represented in such a way that different methods of information processing may be found in the area of artificial intelligence. The most well-known is probably in the form of if-then rules, which are described as IF condition (set of rules) THEN outcome (pattern set). The first step in annotating an ‘Event mention’ in the text is to determine its scope. An Event summary’s scope is always the whole term detailing the Incident. While classifying the scope of the case is not an assessed function, it will be essential in the annotation process. It specifies whether or not the text’s Values and Entities can be utilized as arguments in surrounding Events. Within the scope of an event, only Entities and Values are allowed parameters. A term or phrase that accurately describes its occurrence. Nouns, verbs, and adjectives such as “scam” or “bankrupt” can all be used as event triggers. Rule-based systems may be thought of as data that can be understood in various situations as a collection of rules that determine what to do or infer. Entities and values

that are not appropriately included in the context of an event are known as event attributes as mentioned in Table 3.2 for each class.

Table 3.2: List of Events labels and Attributes

Event Label	Attributes (Supporting terms)
Results	quarter, quarterly, q1, q2, q3, q4, q o q, yearly, year-on-year, y o y, profit, net profit, loss, net loss, estimates, NPA, fiscal, flat, report, reported, provision, provisioning.
RBI_Policies	issue, issued, interest rate, consumer price index, cpi, consumer price index, GDP, gross domestic product, circular, slump, slumped, economists, economy, complaint, complaints.
Merger_Or_Acquisition	amalgamation, amalgamated, subsidiary, merger, merged, merging, integration, disinvest, disinvestment, acquisition, acquire, reduce.
RatingsAgencies_Experts_View	rating, ratings, outlook, maintained, revised, bb-, bb+, cc-, ccc, ccc+, bbb+, bbb-, a+, b+, cc+,a1+, faa, stable, asset, quality.
Fraud	scam, fraud, probe, unfold, misgovernance, scandal, inquiry, case, register, registered, booked, loan default.
Global	trade war, fed, global, globally, Asian, dollar, trade data, import, export, world bank, barrel, Brent crude oil, geo-political, OPEC.
Governmental	union budget, govt borrowing, Indian government bond, GST, election, ltcg, tax, state, center, minister, Lok sabha, NDA, parliament.

3.3. Phase3: Sentiments classification

This phase seeks to use the benefits of DistilBERT's transfer learning for sentiment classification with fine-tuning on Indian banking, financial, and other relevant news on seven events. To the best of our knowledge, this is the first study to try to extract sentiments from banking financial news events. The study covers news events for sentiment classification on

Fraud, Global, Governmental, Merger or Acquisition, RBI Policies, Ratings Agencies, or Expert's View, and Results. Because most research utilizes context-independent methods, the purpose behind this is to leverage the context or semantic connection of the words to identify attitudes behind news articles i.e. only words and their polarities to perform classification [214].

Contextual language representations are thought to have the capacity to significantly decrease the amount of training data required to build optimum techniques by task-specific fine-tuning on relatively little data. As a result, positive, negative, and neutral labels were assigned to news items on Fraud, Global, Governmental, Merger or Acquisition, RBI Policies, Ratings Agencies or Expert's Views, and Results. Even for domain experts, the classification of financial news events is imprecise. The complexity of financial news is to blame for a lot of the confusion. The interpretation of a banking financial event will be significantly influenced by announcements of projected quarterly results, analyst terminology used during ratings, and perceptions of RBI policies, among other things. The annotator's knowledge is required for labeling or annotation of news events into the appropriate polarity. It is carried out with a comprehensive grasp of the features of each news item and their importance in the development of the Indian stock market movement.

3.4. Phase-4: Named entities recognition and analysis of banking stocks and the banking index in Indian Stock market based on past and current linguistic or sentiments of news events

The HuggingFace transformers library in python is used to fine-tune a pre-trained BERT model for identifying named entities in banking news events. We just need to fine-tune BERT in 2-4 epochs, although it is quite huge, sophisticated, and has millions of parameters. Because BERT was trained on a large quantity of data and has already encoded a lot of information about our language, this outcome is possible. BERT is considered the greatest influential NLP model accessible right now, because of its influential performance obtained in less time with a little quantity of data. The CoNLL Dataset [193] is used to demonstrate Named Entity Recognition.

Finally, we analyze how stock markets respond to banking news events, and if so, how these reactions differ depending on if the news is negative, positive, or neutral. The influence of six banking and governmental events on Indian bank stock or index returns is analyzed using the event study technique. This method calculates abnormal returns after a banking news event is introduced to the public on the day 't'. The release of the banking news event explains

variations in the stock prices of the respective bank or index (Bank Nifty). The difference between the actual returns of a bank's stock or index reported on the day 't' and the predicted returns that the bank's stock or index would have shown in the absence of the event is computed as abnormal returns. The three-factor model is used to calculate expected returns [215]. The model includes a US factor, a global financial factor, and a domestic market factor, $F'_t = [R^U_t, R^G_t, R^D_t]$. Because these factors are all value-weighted market indexes, the model can theoretically integrate distinct CAPMs as special cases: when the β 's on the first two factors are zero, the model converts a domestic CAPM; if the β 's on the domestic factor is zero, the model becomes a world CAPM.

$$R_{i,t} = \alpha_i + \beta'_t F_t + \varepsilon_{i,t}, \quad (3.1)$$

Where $R_{i,t}$ refers to stock return, α_i refers to intercept, and β'_t refers to the three-dimensional vector of factor sensitivities for bank i on day t . The idiosyncratic component is $\varepsilon_{i,t}$. Banks might be more or less international depending on their business models. We use the model [215] to estimate abnormal returns. As a result, all of the factors in the factor vector F_t drive the systematic component of returns.

For day t , the abnormal return $AR_{i,t}$ due to any of six banking news events of bank i is computed as follows.

$$AR_{i,t} = R_{i,t} - (\hat{\alpha}_i + \hat{\beta}'_t F_t) \quad (3.2)$$

On day t , the average abnormal return ($\overline{AR}_t$) is calculated as the average abnormal stock return for all n bank shares.

$$\overline{AR}_t = \frac{1}{n} \sum_{i=1}^n AR_{i,t} \quad (3.3)$$

The CAR is calculated as the sum of the average abnormal returns for all days t in the event window for each stock i , $CAR_i(T_1, T_2)$

$$CAR_i(T_1, T_2) = \sum_{t=T_1}^{T_2} \overline{AR}_t \quad (3.4)$$

By calculating the average CARs(T_1, T_2) for all n shares, we may estimate the mean CARs in the event windows ($CAR(1, 2)$).

$$\overline{CAR}_i(T_1, T_2) = \frac{1}{n} \sum_{i=1}^n CAR_i(T_1, T_2) \quad (3.5)$$

The statistical significance of the CARs estimations is determined using the Standardised Cross-Sectional t-test method [216].

Chapter-4

Results and Discussions

The findings and discussions following the implementation of the proposed technique are summarized in this chapter. Isolating banking news from the pool of financial news, news-events representation and classification, NER, sentiment classification, and an event-study are among the results and discussion. This chapter goes through text classification using a hybrid model technique in great depth.

4.1. Banking News Extraction by Multiclass Classification with Imbalanced Datasets of Financial News

We have carried out some experiments on our pre-processed data collection utilizing conventional machine learning models described in the section preceding. The key purpose of the experiments is to determine the right classifier that gives the best performance. Every classifier's output concerning classification is calculated using the metrics Precision, Recall, and F₁-score. The accuracies are obtained with both train-test split and 5-fold cross-validation for all classifiers.

4.1.1. Results of multiclass classification with data-imbalances

Table 4.1 lists the results of each of the classifiers where data is being vectorized using the N-gram TF-IDF feature with data imbalances across classes. This is also shown in fig. 1. From the different classifiers Decision Tree {criterion='gini' to measure quality of split, splitter='best', max_depth=2 for maximum depth of tree, random_state=1 is the seed for random number generator}, Linear SVC {C=1 regularization parameter, multi_class='y'}, Logistic Regression {C=1 regularization parameter, random_state=0}, Multinomial Naïve Bayes {alpha=1.0 smoothing parameter}, Random Forest {n_estimators=100 for number of trees, random_state=1 will always produce same results with same parameters and training data, max_depth=3 for maximum depth of the tree}, and Multilayer Perceptron {solver='lbfgs' for weight optimization, alpha=0.0001 L2 penalty, learning_rate='constant', hidden_layer_sizes=(5,2), random_state=1}, Random Forest performed best with accuracy 88% as shown in Table 4.2. The Random Forest achieved the F₁-score 0.96, 0.71, 0.92, 0.38 for classes Banking, Global, Non-Banking, and Governmental respectively. The comparison of all the mentioned classifiers for 4-different classes is visualized in Fig 4.1-4.4. Table 4.2 shows that the accuracy comes out to be 78%-88% range for all machine learning algorithms

with train-test split and cross-validation. However, Table 4.1 shows that the recall of the minority classes is very less. The Logistic Regression and Multinomial NB have shown 0% precision and recall for the minority class i.e. Governmental. This is visualized in fig. 5. At the same time, the precision and recall for the other classes have shown high precision and recall. It shows that machine learning models are more biased towards the majority class. So, we need to apply imbalanced data handling techniques.

TABLE 4.1 : RESULTS FOR THE CLASSIFIERS FOR DIFFERENT CLASSES WITH DATA IMBALANCES

Classifier	N-Gram	Banking			Global			Non-Banking			Governmental		
		P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
Decision Tree	1, 2, 3	0.93	0.96	0.94	0.76	0.68	0.72	0.90	0.95	0.92	0.80	0.31	0.44
Linear SVC	1, 2, 3	1.00	0.77	0.87	0.86	0.61	0.71	0.86	0.98	0.91	1.00	0.38	0.56
Logistic Regression	1, 2, 3	1.00	0.31	0.47	0.88	0.34	0.49	0.76	0.99	0.86	0.00	0.00	0.00
Multinomial NB	1, 2, 3	0.86	0.23	0.36	0.73	0.54	0.62	0.79	0.97	0.87	0.00	0.00	0.00
Random Forest	1, 2, 3	0.93	1.00	0.96	0.89	0.59	0.71	0.88	0.98	0.92	1.00	0.23	0.38
Multi-layer Perceptron	1, 2, 3	1.00	0.69	0.82	0.78	0.68	0.73	0.86	0.96	0.91	1.00	0.31	0.47

TABLE 4.3 : RESULTS FOR THE CLASSIFIERS FOR DIFFERENT CLASSES WITH BALANCED DATA USING UP-SAMPLING

Classifier	N-Gram	Banking			Global			Non-Banking			Governmental		
		P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
Decision Tree	1, 2, 3	0.99	1.00	0.99	0.95	0.99	0.97	0.99	0.94	0.97	0.99	1.00	1.00
Linear SVC	1, 2, 3	0.98	1.00	0.99	0.98	0.99	0.98	0.99	0.97	0.98	0.99	1.00	1.00
Logistic Regression	1, 2, 3	0.98	1.00	0.99	0.93	1.00	0.96	1.00	0.91	0.95	0.99	1.00	1.00
Multinomial NB	1, 2, 3	0.97	0.96	0.97	0.93	0.97	0.95	0.93	0.83	0.88	0.94	1.00	0.97
Random Forest	1, 2, 3	0.99	1.00	1.00	0.98	0.99	0.98	0.99	0.98	0.98	1.00	1.00	1.00
Multi-layer Perceptron	1, 2, 3	0.99	1.00	0.99	0.94	0.99	0.97	0.99	0.93	0.96	1.00	1.00	1.00

TABLE 4.2: ACCURACY OF THE CLASSIFIERS WITH IMBALANCED DATA

Classifier	Accuracy(Train/Test) Cross-Validation	
Decision Tree	0.87	0.87
Linear SVC	0.87	0.87
Logistic Regression	0.78	0.78
Multinomial NB	0.78	0.78
Random Forest	0.88	0.88
Multi-layer Perceptron	0.87	0.87

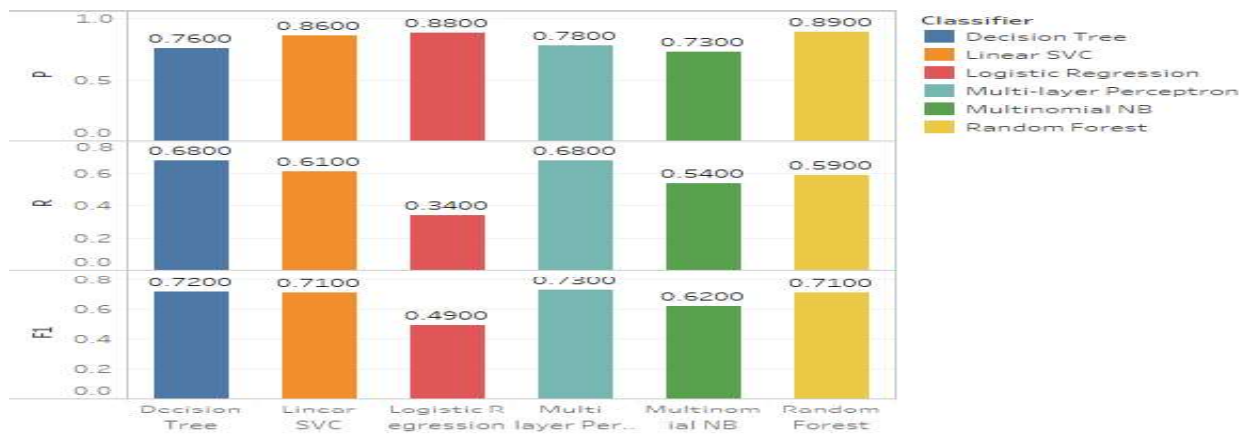


Fig 4.1: Performance metrics P, R, F-1 for various classifiers for Banking Class

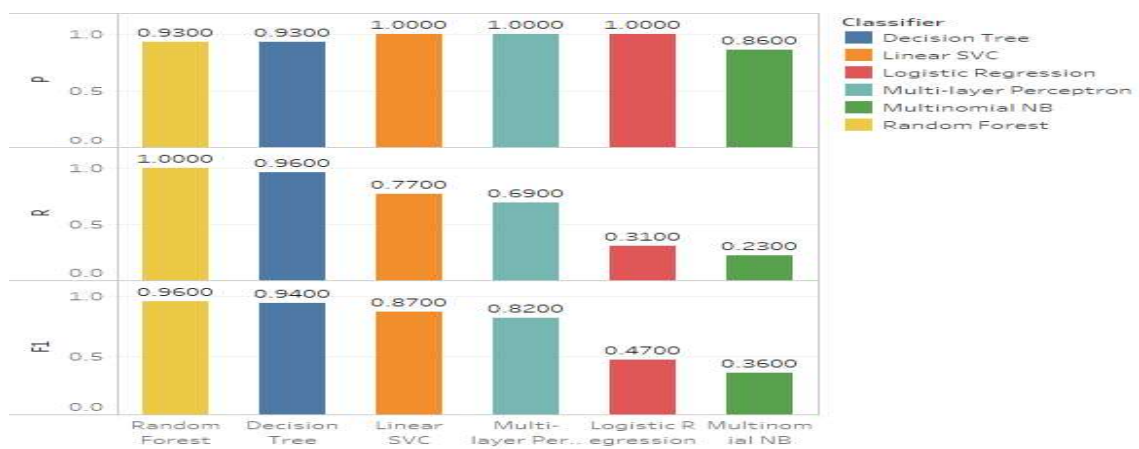


Fig 4.2: Performance metrics P, R, F-1 for various classifiers for Global Class

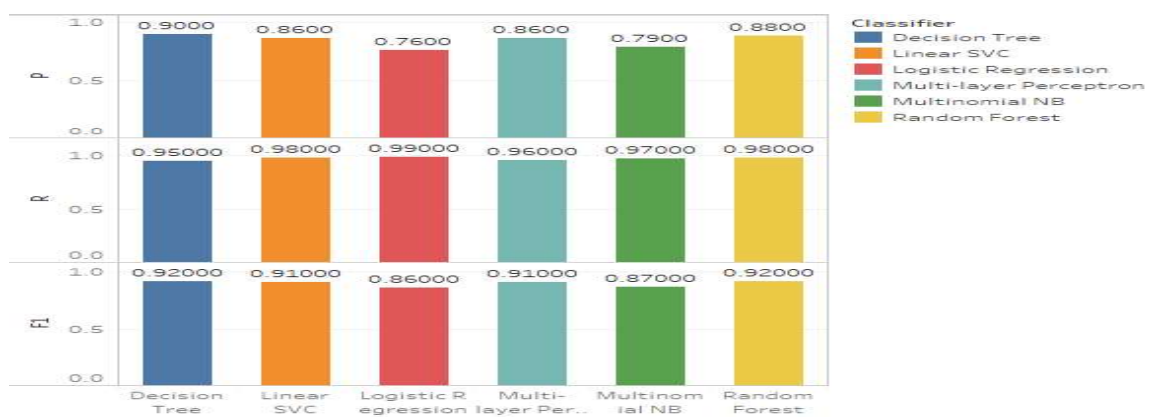


Fig 4.3: Performance metrics P, R, F-1 for various classifiers for Non-Banking Class

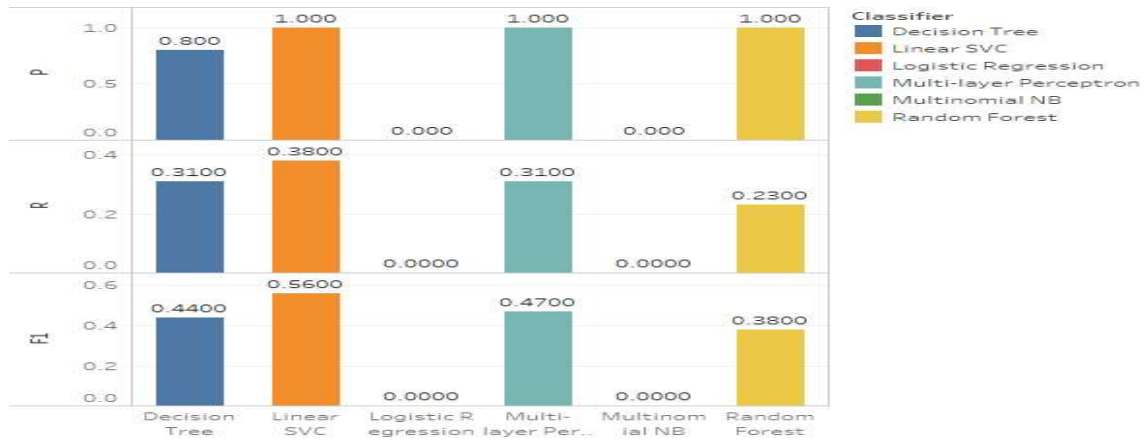


Fig 4.4: Performance metrics P, R, F-1 for various classifiers for Governmental Class

4.1.2. Results of multiclass classification with data balance using Data-level technique: Random Over-Sampling with replacement

The Resampling takes place with the exclusion of the minority class, increasing the sample number to equal that of the majority class. Tables 4.3 and 4.4 lists the results of each of the classifiers where data is being vectorized using the N-gram TF-IDF feature with data balanced using random over-sampling technique across classes.

TABLE 4.4: ACCURACY OF THE CLASSIFIERS WITH BALANCED DATA USING UP-SAMPLING

Classifier	Accuracy(Train/Test)	Cross-Validation
Decision Tree	0.98	0.983
Linear SVC	0.98	0.982
Logistic Regression	0.98	0.975
Multinomial NB	0.94	0.948
Random Forest	0.99	0.996
Multi-layer Perceptron	0.98	0.985

From the different classifiers Decision Tree, Linear SVC, Logistic Regression, Multinomial Naïve Bayes, Random Forest, and Multilayer Perceptron with accuracy for all classes with balanced datasets using up-sampling, the Random Forest again performed best with accuracy 99% as shown in Table 5. The Random Forest achieved the F₁-score 1.00, 0.98, 0.98, 1.00 for classes Banking, Global, Non-Banking, and Governmental respectively. The comparison of all the mentioned classifiers for 4-different classes is shown in Tables 4.3 and 4.4. It is observed that with data balances the precision and recall have also improved for every classifier. And the accuracy of the classifiers varies between 94% to 100% and it is visualized in fig 4.5.

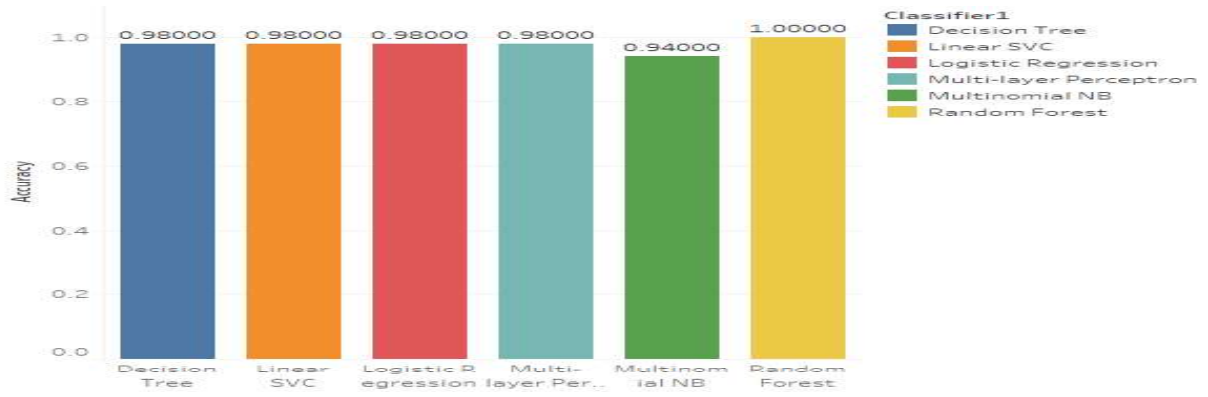


Fig 4.5: Accuracy for various classifiers with balanced classes using Up-Sampling

4.1.3. Results of multiclass classification with data balance using Data-level technique: Random Down-Sampling without replacement.

This is done by resampling the majority class without replacement, setting the number of samples corresponding to that of the minority class. Table 4.5, 4.6 lists the results of each of the classifiers where data is being vectorized using the N-gram TF-IDF feature with data balanced using down-sampling technique across classes.

TABLE 4.6: ACCURACY OF THE CLASSIFIERS WITH BALANCED DATA USING DOWN-SAMPLING

Classifier	Accuracy(Train/Test) Cross-Validation	
Decision Tree	0.69	0.695
Linear SVC	0.76	0.764
Logistic Regression	0.71	0.720
Multinomial NB	0.67	0.750
Random Forest	0.80	0.803
Multi-layer Perceptron	0.69	0.692

From the different classifiers Decision Tree, Linear SVC, Logistic Regression, Multinomial Naïve Bayes, Random Forest, and Multilayer Perceptron with accuracy for all classes with balanced datasets using down-sampling, the Random Forest again performed best with an accuracy of 80% as shown in Table 4.6.

The Random Forest achieved the F₁-score 0.95, 0.83, 0.73, 0.70 for classes Banking, Global, Non-Banking, and Governmental respectively. The comparison of all the mentioned classifiers for 4-different classes is shown in Tables 4.5 and 4.6. The accuracy of the classifiers has degraded with data balances using down-sampling as compared to up-sampling. And the accuracy of the classifiers varies between 67% to 80% and it is visualized in fig 4.6.

TABLE 4.5: RESULTS FOR THE CLASSIFIERS FOR DIFFERENT CLASSES WITH BALANCED DATA USING DOWN-SAMPLING

Classifier	N-Gram	Banking			Global			Non-Banking			Governmental		
		P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
Decision Tree	1, 2, 3	1.00	0.91	0.95	0.56	0.83	0.67	0.71	0.38	0.50	0.60	0.67	0.63
Linear SVC	1, 2, 3	1.00	0.91	0.95	0.75	0.80	0.77	0.78	0.67	0.72	0.67	0.67	0.67
Logistic Regression	1, 2, 3	1.00	0.82	0.90	0.69	0.75	0.72	0.70	0.54	0.61	0.54	0.78	0.64
Multinomial NB	1, 2, 3	0.82	0.82	0.82	0.69	0.75	0.72	0.67	0.31	0.42	0.53	0.89	0.67
Random Forest	1, 2, 3	1.00	0.91	0.95	0.83	0.83	0.83	0.89	0.62	0.73	0.57	0.89	0.70
Multi-layer Perceptron	1, 2, 3	0.88	0.64	0.74	0.71	0.83	0.77	0.62	0.62	0.62	0.60	0.67	0.63

TABLE 4.7: RESULTS FOR THE CLASSIFIERS FOR DIFFERENT CLASSES WITH BALANCED DATA USING SMOTE

Classifier	N-Gram	Banking			Global			Non-Banking			Governmental		
		P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
Decision Tree	1, 2, 3	0.99	0.93	0.96	0.99	1.00	1.00	0.95	0.99	0.97	0.98	1.00	0.99
Linear SVC	1, 2, 3	0.99	0.92	0.96	0.98	1.00	0.99	0.94	0.99	0.97	0.99	1.00	1.00
Logistic Regression	1, 2, 3	1.00	0.91	0.95	0.98	1.00	0.99	0.93	1.00	0.96	0.99	1.00	1.00
Multinomial NB	1, 2, 3	0.92	0.85	0.88	0.97	0.96	0.97	0.94	0.96	0.95	0.94	1.00	0.97
Random Forest	1, 2, 3	0.99	0.99	0.99	0.99	1.00	1.00	0.99	0.99	0.99	1.00	1.00	1.00
Multi-layer Perceptron	1, 2, 3	0.99	0.93	0.96	0.99	1.00	0.99	0.94	0.99	0.97	1.00	1.00	1.00

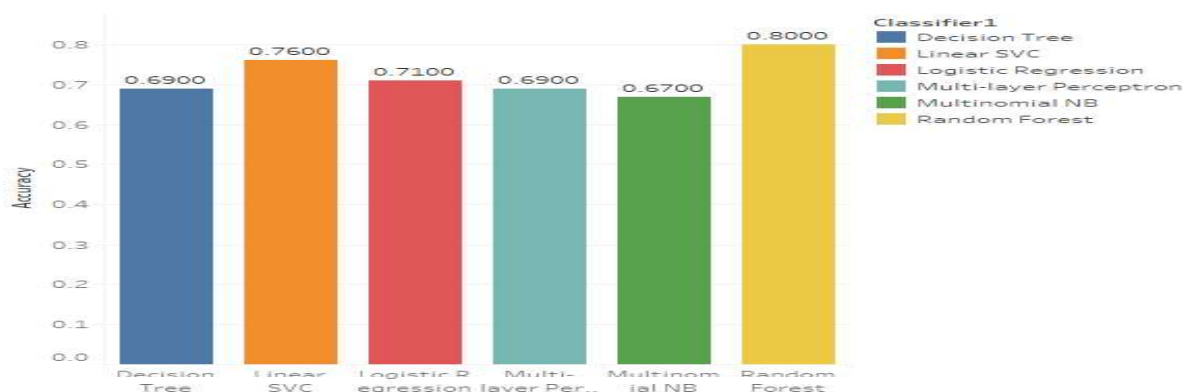


Fig 4.6: Accuracy for various classifiers with balanced classes using Down-Sampling

4.1.4. Results of multiclass classification with data balance using Data-level technique: hybrid over-sampling technique SMOTE

SMOTE helps to balance the representation of the classes by replicating randomly through minority class examples. SMOTE synthesizes new instances within existing instances of minority classes. This produces the virtual train records by linear interpolation for the minority class. For each case, these synthetic training records are created by a random selection of one or more k-nearest neighbors in the minority class. The data is reconstructed after the oversampling process, and the classification models are implemented for the processing data. Table 4.7 lists the results of each of the classifiers where data is being vectorized using the N-gram TF-IDF feature with data balanced using the over-sampling technique SMOTE across classes.

TABLE 4.8: ACCURACY OF THE CLASSIFIERS WITH BALANCED DATA USING SMOTE UP-SAMPLING

Classifier	Accuracy(Train/Test) Cross-Validation	
Decision Tree	0.98	0.981
Linear SVC	0.98	0.948
Logistic Regression	0.98	0.972
Multinomial NB	0.94	0.948
Random Forest	1.00	0.995
Multi-layer Perceptron	0.98	0.986

From the different classifiers Decision Tree, Linear SVC, Logistic Regression, Multinomial Naïve Bayes, Random Forest, and Multilayer Perceptron with accuracy for all classes with balanced datasets using up-sampling, the Random Forest again performed best with accuracy 100% as shown in Table 4.8. The Random Forest achieved the F₁-score 0.99, 1.00, 0.99, 1.00 for classes Banking, Global, Non-Banking, and Governmental respectively. The comparison of all the mentioned classifiers for 4-different classes is shown in Tables 4.7 and 4.8. It is observed that with data balances using SMOTE the precision and recall have also improved for every classifier. And the accuracy of the classifiers varies between 94% to 100% as visualized in fig 4.7.

TABLE 4.9: RESULTS FOR THE CLASSIFIERS FOR DIFFERENT CLASSES WITH BALANCED DATA USING ADASYN

Classifier	N-Gram	Banking			Global			Non-Banking			Governmental		
		P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
Decision Tree	1, 2, 3	0.89	0.96	0.93	0.82	0.66	0.73	0.91	0.94	0.92	0.42	0.38	0.40
Linear SVC	1, 2, 3	0.96	0.85	0.90	0.76	0.71	0.73	0.89	0.95	0.92	0.62	0.38	0.48
Logistic Regression	1, 2, 3	0.96	0.85	0.90	0.79	0.76	0.73	0.90	0.95	0.92	0.60	0.46	0.52
Multinomial NB	1, 2, 3	0.50	0.85	0.63	0.57	0.93	0.70	0.99	0.65	0.78	0.29	0.77	0.43
Random Forest	1, 2, 3	0.93	0.96	0.94	0.91	0.71	0.79	0.91	0.98	0.94	0.88	0.38	0.53
Multi-layer Perceptron	1, 2, 3	0.94	0.65	0.77	0.73	0.73	0.73	0.88	0.95	0.92	0.88	0.38	0.53

TABLE 4.11: RESULTS FOR THE CLASSIFIERS FOR DIFFERENT CLASSES WITH BALANCED DATA USING NEAR-MISS

Classifier	N-Gram	Banking			Global			Non-Banking			Governmental		
		P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
Decision Tree	1, 2, 3	0.68	1.00	0.81	0.23	0.24	0.24	0.82	0.70	0.75	0.25	0.54	0.34
Linear SVC	1, 2, 3	0.41	0.96	0.57	0.30	0.56	0.39	0.88	0.45	0.59	0.27	0.69	0.39
Logistic Regression	1, 2, 3	0.43	0.96	0.60	0.29	0.56	0.38	0.94	0.17	0.28	0.12	0.92	0.22
Multinomial NB	1, 2, 3	0.32	0.88	0.47	0.30	0.59	0.40	0.91	0.32	0.47	0.20	0.77	0.32
Random Forest	1, 2, 3	0.91	0.77	0.83	0.67	0.20	0.30	0.82	0.97	0.89	0.60	0.46	0.52
Multi-layer Perceptron	1, 2, 3	0.46	0.81	0.58	0.57	0.61	0.59	0.91	0.72	0.80	0.28	0.62	0.38

TABLE 4.10: ACCURACY OF THE CLASSIFIERS WITH BALANCED DATA USING ADASYN UP-SAMPLING

Classifier	Accuracy(Train/Test)	Cross-Validation
Decision Tree	0.87	0.872
Linear SVC	0.87	0.865
Logistic Regression	0.88	0.881
Multinomial NB	0.72	0.725
Random Forest	0.91	0.914
Multi-layer Perceptron	0.86	0.863

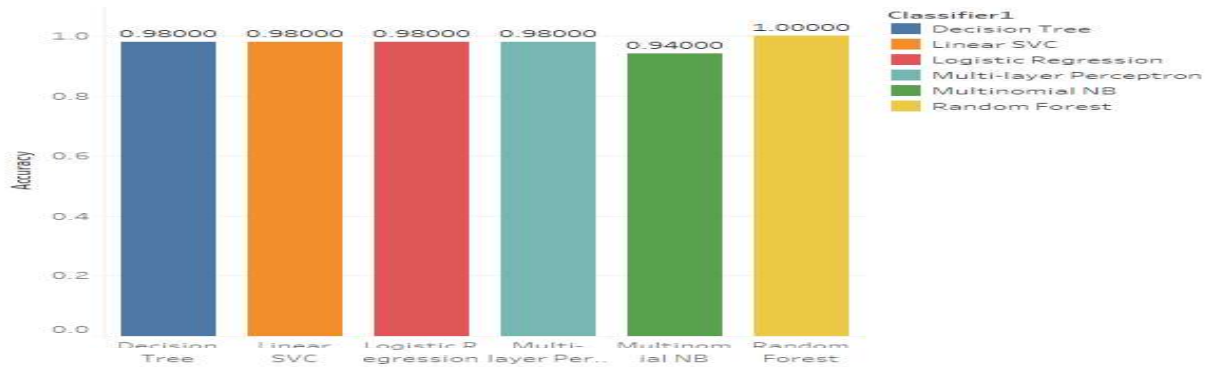


Fig 4.7: Accuracy for various classifiers with balanced classes using SMOTE Up-Sampling

4.1.5. Results of multiclass classification with data balance using Data-level technique: over-sampling technique ADASYN

ADASYN (Adaptive synthetic sampling approach) algorithm builds on the methodology of SMOTE. This uses a weighted distribution for specific examples of minority classes due to their degree of learning capacity, whereas more sophisticated data is produced for examples of minority classes. The main objective of the ADASYN algorithm is to apply a density distribution as a parameter to calculate the number of synthetic samples that each minority data example requires to be generated. The data is reconstructed after the oversampling process, and the classification models are implemented for the processing data. Table 4.9 lists the results of each of the classifiers where data is being vectorized using the N-gram TF-IDF feature with data balanced using From the different classifiers Linear SVC, Decision Tree, Multinomial Naïve Bayes, Logistic Regression, Random Forest, and Multilayer Perceptron with accuracy for all classes with balanced datasets using up-sampling, the Random Forest again performed best with accuracy 91% as given in Table 4.10. The Random Forest achieved the F_1 -score 0.94, 0.79, 0.94, 0.53 for classes Banking, Global, Non-Banking, and Governmental respectively. The comparison of all the mentioned classifiers for 4-different classes is shown in Tables 4.9

and 4.10. It is observed that with data balances using ADASYN the precision and recall have also downgrade as compared to SMOTE-based up-sampling for every classifier. And the accuracy of the classifiers varies between 72% to 91% as visualized in fig 4.8.

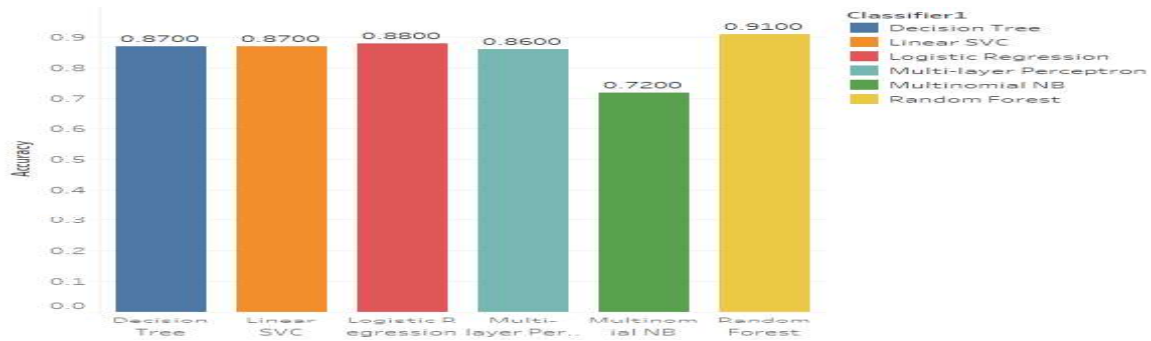


Fig 4.8: Accuracy for various classifiers with balanced classes using ADASYN Up-Sampling

4.1.6. Results of multiclass classification with data balances using Data-level technique: down-sampling technique Near-Miss

The NearMiss Algorithm under-sampled the majority class's instances and made them equivalent to the minority class. The majority classes, here, were reduced to the minimum number as of the minority class so that all classes would have the same number of records. The data is reconstructed after the down-sampling process using the Near-Miss method, and the classification models are implemented for the processing data. Table 4.11 shows the outcomes of each of the classifiers where data is being vectorized using the N-gram TF-IDF feature with data balanced using the down-sampling technique Near-Miss across classes.

From the different classifiers Linear SVC, Decision Tree, Multinomial Naïve Bayes, Logistic Regression, Random Forest, and Multilayer Perceptron with accuracy for all classes with balanced datasets using down-sampling, the Random Forest again performed best with an accuracy of 81% as shown in Table 4.12. The Random Forest achieved the F_1 -score 0.83, 0.30, 0.89, 0.52 for classes Banking, Global, Non-Banking, and Governmental respectively. The comparison of all the mentioned classifiers for 4-different classes is shown in Tables 4.11 and 4.12. The accuracy of the classifiers has degraded with data balances using down-sampling with the Near-Miss approach as compared to all other up-sampling approaches as visualized in fig 4.9.

Table 4.12: Accuracy of the classifiers with balanced data using Near-Miss down-sampling

Classifier	Accuracy(Train/Test)	Cross-Validation
Decision Tree	0.65	0.652
Linear SVC	0.53	0.526
Logistic Regression	0.34	0.344
Multinomial NB	0.43	0.431
Random Forest	0.81	0.814
Multi-layer Perceptron	0.70	0.704

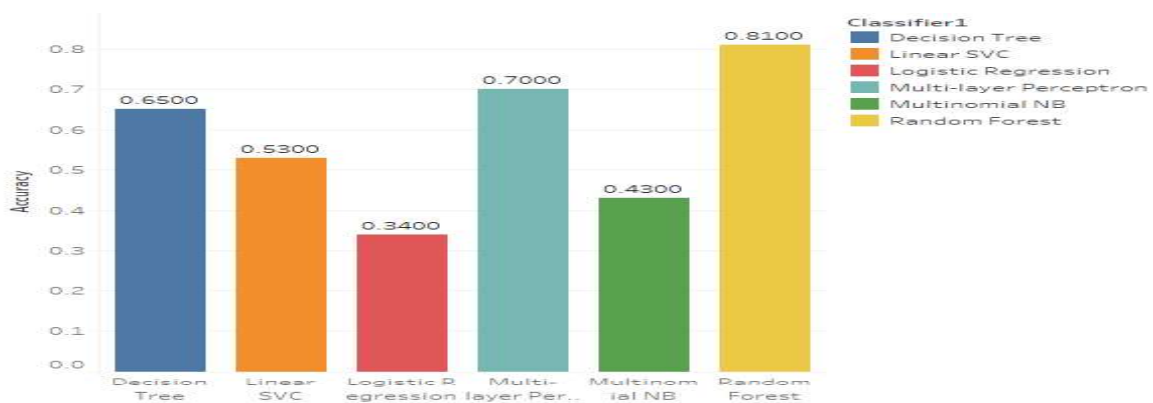


Fig 4.9: Accuracy for various classifiers with balanced classes using Near-Miss Down-Sampling

4.1.7. Results of multiclass classification with data balance using Ensemble classifiers

Ensemble models are meta-algorithms that combine many machine learning techniques into a single predictive model to reduce variance (bagging), bias (boosting) or reinforce predictions (stacking). Bagging methods build multiple estimators on various randomly chosen subsets of data in ensemble classifiers. The classifier is called `BaggingClassifier` in scikit-learn. This classifier, however, does not require a balancing of the data sub-set. So, this classifier would support the plurality groups when training on imbalanced data set. ***BalancedBaggingClassifier*** requires each subset of data to be resampled until any of the ensemble estimators are equipped. In brief, the performance of an `EasyEnsemble` sampler is paired with an ensemble of classifiers (i.e., `BaggingClassifier`). Hence the `BalancedBaggingClassifier` requires the same parameters as the `BaggingClassifier` scikit-learn. Additionally, there are two additional parameters to monitor the actions of the random under-sampler, sampling strategy, and substitution. ***BalancedRandomForestClassifier*** is another ensemble method that provides a balanced bootstrap sample for each tree in the forest. ***RUSBoostClassifier*** sub-sample the data collection randomly before executing a boosting iteration. A particular method in the bagging classifier which uses AdaBoost as learners is named `EasyEnsemble`. The ***EasyEnsembleClassifier***

allows AdaBoost learners to be trained on appropriate samples of bootstrap. Table 4.13 lists the results of each of these ensemble classifiers for the various classes. And Table 4.14 lists the accuracy of these ensemble classifiers.

TABLE 4.13: RESULTS FOR THE ENSEMBLE CLASSIFIERS FOR DIFFERENT CLASSES

Classifier	N-Gram	Banking			Global			Non-Banking			Governmental		
		P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
BalancedBaggingClassifier	1, 2, 3	0.98	0.97	0.97	0.99	1.00	1.00	0.98	0.98	0.98	0.99	1.00	1.00
BalancedRandomForestClassifier	1, 2, 3	0.86	0.64	0.73	0.99	0.90	0.94	0.66	0.88	0.76	0.84	0.87	0.86
RUSBoostClassifier	1, 2, 3	0.33	1.00	0.50	0.94	0.34	0.50	0.08	0.05	0.06	0.00	0.00	0.00
EasyEnsembleClassifier	1, 2, 3	0.87	0.93	0.90	0.57	0.44	0.49	0.26	0.85	0.40	0.92	0.83	0.87

Table 4.14: Accuracy of the ensemble classifiers

Classifier	Accuracy(Train/Test)	Cross-Validation
BalancedBaggingClassifier	0.99	0.991
BalancedRandomForestClassifier	0.82	0.823
RUSBoostClassifier	0.34	0.344
EasyEnsembleClassifier	0.78	0.781

From the different ensemble classifiers BalancedBaggingClassifier, BalancedRandomForestClassifier, RUSBoostClassifier, EasyEnsembleClassifier with accuracy for all classes, the BalancedBaggingClassifier performed best with an accuracy of 99% as shown in Table 4.14. The BalancedBaggingClassifier achieved the F₁-score 0.97, 1.00, 0.98, 1.00 for classes Banking, Global, Non-Banking and Governmental respectively. The comparison of all the mentioned ensemble classifiers for 4-different classes is shown in Table 4.13 and 4.14. The accuracy of the BalancedBaggingClassifier has resulted in 99% which is quite similar to the result of multiclassification using Random-Forest Classifier with SMOTE sampling i.e. 100%. The accuracy of the Random Forest classifier with a random up-sampling approach for data balances is also 99%. The comparison of the accuracies of classifiers across all approaches is visualized in fig 4.10. The accuracy of classifiers with down-sampling using the Near-Miss approach is worst amongst all.

The accuracy, precision, recall, and F-1 of Random-Forest Classifier with SMOTE sampling is very good in terms of multiclass news classification. However, under Governmental and Banking classes (minor classes in original), the precision of Random Forest with SOMTE overlapped with the precision of Random Forest with a random up-sampling

approach. The comparison of the Precision of classifiers with each approach across all mentioned classes is visualized in fig 4.11. One of the key explanations for the low performance of some of the classifiers, including Linear SVC and Multinomial naïve Bayes, is that a huge number of features don't fit well for them. Earlier it has been stated that Multinomial Naive Bayes' output is very weak when the dataset faces class imbalance problems. The result has shown that the efficiency of the RUSBoostClassifier ensemble algorithm is very poor when it comes to the multi-class classification of text with noisy data and class imbalance.

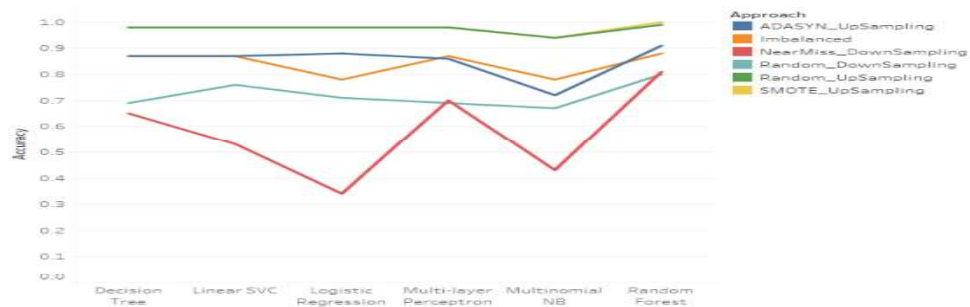


Fig 4.10: Comparison of accuracies with Classifiers across different approaches

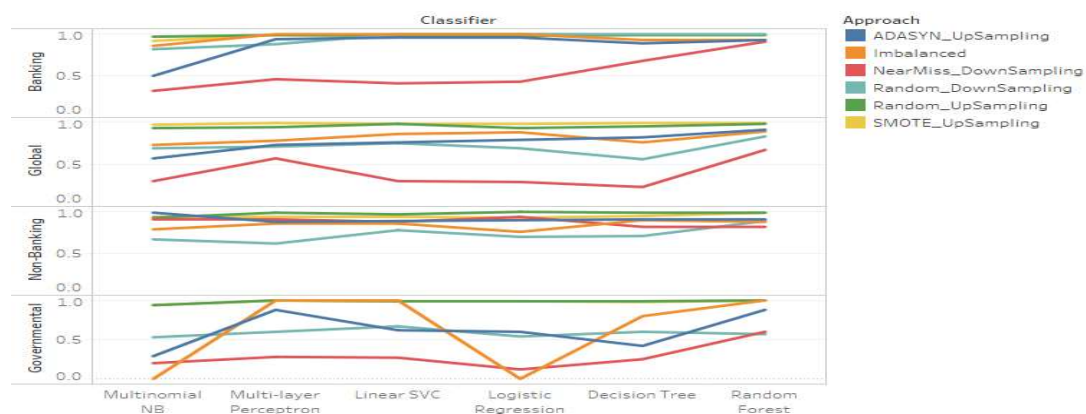


Fig 4.11: Comparison of Precision with Classifiers under each class across different approaches

It is clear from fig 4.12, the recall of the classifier Random Forest with data balanced across classes using random up-sampling and SMOTE is increased as compared to down-sampling techniques random down-sampling and Near-Miss. The comparison of recall across all approaches with different classifiers under each class is visualized in fig 4.12.

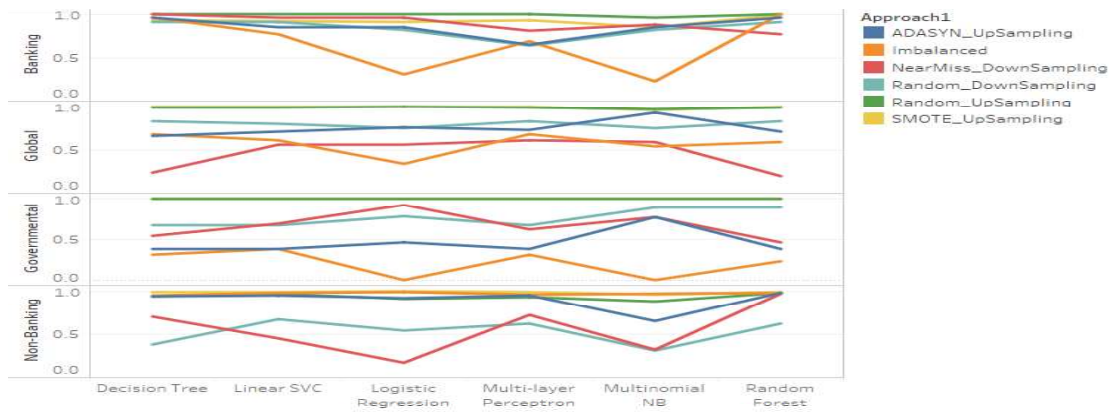


Fig 4.12: Comparison of Recall with Classifiers under each class across different approaches

The accuracy of the ensemble classifiers is compared with Random Forest with SMOTE and it is visualized in fig 4.13. The accuracy of multi-class financial news classification using Random Forest with data balanced using SMOTE is higher as compared to all other ensemble classifiers discussed in the previous section. It is slightly greater than BalancedBaggingClassifier. The precision and recall of Random Forest with data balanced using SMOTE across all classes are higher as compared to all other ensemble classifiers and it is visualized in fig 4.14 and 4.15 respectively.

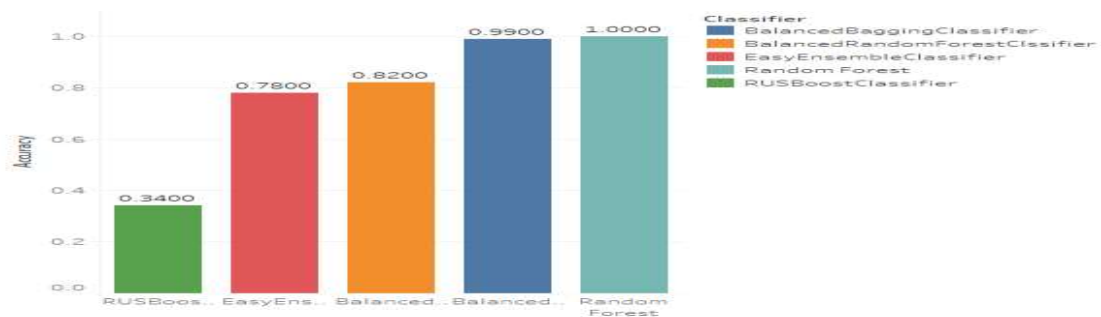


Fig 4.13: Ensemble classifiers vs Random Forest (SMOTE)

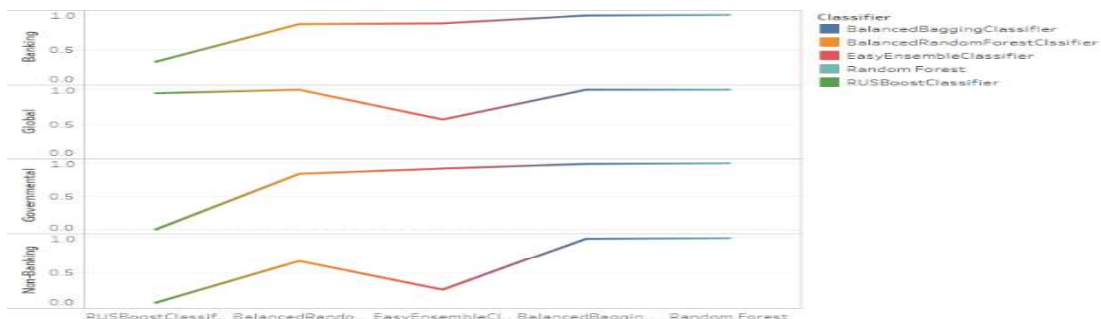


Fig 4.14: Precision of ensemble classifiers vs Random Forest (SMOTE)

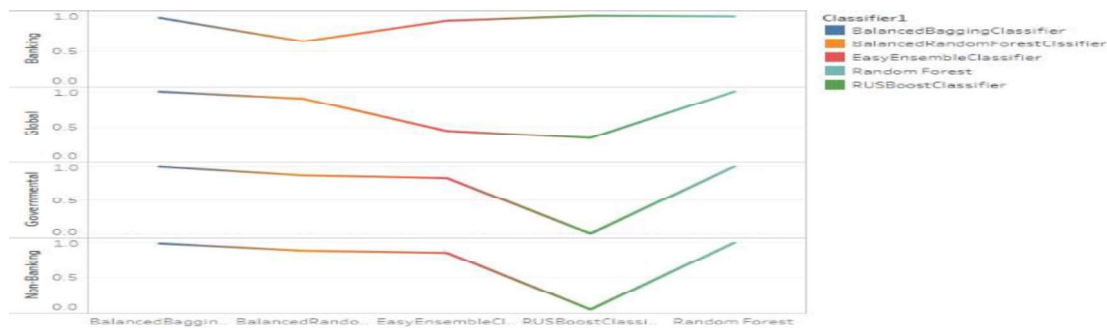


Fig 4.15: Recall of ensemble classifiers vs Random Forest (SMOTE)

4.2. Banking news-events representation and classification with a novel hybrid model using DistilBERT and rule-based features

An experimental assessment of Indian banking news for event extraction and classification utilizing a hybrid technique by discovering event extent and event triggers using a rule-based approach in combination with a machine-learning algorithm is part of our research.

4.2.1. Data sets

Using python-written code, we scraped news from public news sources such as Bloomberg, Financial Express, Money Control, and Times of India for our research. As a result, from 2017 to 2020, we have gathered almost 10,000 instances of financial news articles. The news articles are related to a variety of events. We'd like to categorize banking and other relevant news articles into seven categories as mentioned in Table 4.15. The data was gathered by executing a python script many times every day.

The news data is then pre-processed such that machine learning or deep learning models may learn from the training dataset and adapt to the testing dataset reasonably. As a result, these are pre-processed for machine learning models to be developed from the training sample and applied to the test data set in an appropriate format.

Table 4.15: List of Events labels and news samples for classification

Event Label	News Samples	Date and Time
Results	"For the quarter ended June, private lender HDFC Bank NSE -1.84 % an 18.17 % year-on-year (YoY) increase in net profit of Rs 4,601.44 crore."	Jul 21, 2018, 05:34 PM
RBI_Policies	"The Reserve Bank of India has lowered interest rates to their lowest level in almost two decades, predicting that the economy will shrink this fiscal year."	MAY 22, 2020, 12:45 PM

Merger_Or_Acquisition	“Punjab National Bank will be formed through the merger of Oriental Bank of Commerce (OBC) and United Bank of India (PNB). Following the merger, these two banks would create India's second-largest public sector bank, behind State Bank of India (SBI).”	Apr 01, 2020, 08:18 AM
RatingsAgencies_Experts_View	“IDBI Bank's Viability Rating (VR) was improved by one notch from "ccc" to "ccc+" by Fitch Ratings. The enhanced core capitalization and excellent loan-loss coverage are the key reasons for the VR upgrade.”	December 04, 2020, 12:53 P.M
Fraud	“The main indexes Sensex and Nifty rose just little in early trade on Monday, owing to a drop in PSU bank shares. In today's trading, shares of the PNB, which has been plagued by fraud, fell more than 4%.”	February 20, 2018, 10:35 AM
Global	“Trump issued two proclamations imposing a 25% tariff on steel imports and a 10% tax on aluminum imports from all nations except Canada and Mexico.”	March 09, 2018, 07:21 PM
Governmental	“In comparison to last year, the LTCG tax has dulled the edge of stocks; but, from a growth standpoint, equities should be able to outperform other asset classes, ensuring that inflows into equity continue.”	March 15, 2018, 11:01 AM

4.2.2. Data transformation and classification

The following stage, data transformation, is to convert the data into forms that are acceptable for the classification process, and *DistilBert* is fine-tuned on the news articles in conjunction with the machine learning classifier by producing a vector collection of features. Using the up-sampling approach, the feature vector is normalized and scaled to evade an imbalanced dataset. This stage also defines the classes (events) on a set of training data, which are subsequently used by the machine learning classifier to conduct data classification. *Logistic Regression*, *Linear SVC*, *Decision Tree*, and *Random Forest* were the four most popular machine learning classifiers evaluated for classification tasks. For dealing with Python's unbalanced dataset, machine learning methods were built on data for categorization using Python's libraries *scikit-learn* and *imblearn*. *DistilBERT* was created using Python's *transformer* library. The model was developed using *google-colab*.

By using micro-averaging, we were able to assess *Precision* and *Recall*. Precision refers to the likelihood that what has been labeled as being in that class is what has been labeled as

being. The likelihood that an object in a class is now recognized as belonging to that class is called recall. Concerning seven Banking News-Events, we compared the performance of the above described classifiers mentioned in Table 4.15. Among all machine learning methods, the Random Forest classifier has the highest accuracy.

4.2.3. Applying rules

Our Rule-based section based on simple or complicated logical expressions of Natural Language processing fine-tuned the foundation model further. In this section of our model, we have certain rule descriptions that can put such requirements on the terms that occur in news articles or not. For our implementation, however, a simplified rule language has been created to make rule creation easier. Each event label may be associated with one or more rules. Each rule is compared to the input news article to approve or reject a category, based on whether the input news piece meets the rule's criteria. Rejecting a label reduces the number of false positives provided by the classifier, which improves accuracy. By adding a new label, all false negatives are eliminated, and recall improves. The advantage of our approach is that by fine-tuning the classifier by creating unique rules for each label, we were able to increase the accuracy of our model by 1.0% over the baseline model. When compared to the expert's efforts necessary to train the classifier on a training set containing a collection of documents for each label with stated rules, writing rules for each label did not require a significant expert workload. The following elements are evaluated for validating categories for each category:

- Collection of attributes for i^{th} event $E_i = \langle e_{i1}, e_{i2}, \dots, e_{in} \rangle$

The news-events must consist at-least one of these attributes;

if $\langle e_{i1} \text{ or } e_{i2} \text{ or } \dots \text{ or } e_{in} \rangle$ then

label the category (i)

else

pass to the next article

- Collection of non-supporting attributes for i^{th} event $NE_i = \langle ne_{i1}, ne_{i2}, \dots, ne_{in} \rangle$

The news-events must not consist any of these attributes;

if $\langle ne_{i1} \text{ or } ne_{i2} \text{ or } \dots \text{ or } ne_{in} \rangle$ then

pass to the next article

else

label the category (i)

- The decision of labeling the news article will depend on the count of supporting attributes resulting in each news article in favor of a particular category. The total number of attributes for each category is kept the same.

4.2.4. Results of Banking News-Events classification with DistilBERT

Using state-of-the-art context-dependent representation and sentence embedding techniques, we conducted numerous tests on our pre-processed news articles. The embeddings are then given to traditional machine learning methods using DistilBert. The main goal of these tests is to find the best classifier for each event. Precision, Recall, and F1-score are used to compute each classifier's output for classification. The results of the top classifier, 'Random Forest', are fine-tuned further with the help of a Rules-based expert system.

Table 4.16 shows the precision, recall, and F-1 score of DistilBERT fine-tuned using several machine learning classifiers on banking news events. The accuracy of the models is given in Table 4.17.

Table 4.16. Results of the machine learning classifiers fine-tuned with DistilBERT

Classifier	Logistic Regression			Random Forest			Decision Tree			Linear SVC		
	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
Fraud	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Global	1.00	0.88	0.93	1.00	1.00	1.00	0.97	1.00	0.98	1.00	1.00	1.00
Governmental	1.00	1.00	1.00	1.00	1.00	1.00	0.96	1.00	0.98	1.00	1.00	1.00
Merger_Or_Acquisition	0.96	1.00	0.98	0.97	1.00	0.98	1.00	1.00	1.00	0.97	1.00	0.98
RBI_Policies	0.94	1.00	0.97	1.00	1.00	1.00	0.91	1.00	0.95	1.00	1.00	1.00
RatingsAgencies_Experts_View	1.00	1.00	1.00	1.00	0.96	0.98	1.00	0.82	0.90	1.00	0.89	0.94
Results	1.00	1.00	1.00	0.96	1.00	0.98	1.00	1.00	1.00	0.93	1.00	0.96

Table 4.17: Accuracy of the classifiers with DistilBERT

Classifier	Accuracy
Logistic Regression	0.98
Random Forest	0.99
Decision Tree	0.97

Linear SVC	0.98
------------	------

As demonstrated in Table 4.17, the Random Forest performed the best out of the four classifiers (Decision Tree, Linear SVC, Logistic Regression, and Random Forest). For the classifications Fraud, Global, Governmental, Merger Or Acquisition, RBI Policies, RatingsAgencies Experts View, and Results, the Random Forest got F1 scores of 1.00, 1.00, 1.00, 0.98, 1.00, 0.98, and 0.98, respectively. Table 4.16 compares all of the aforementioned classifiers in terms of Precision, Recall, and F-1 score for seven distinct classes of Banking News Events. The classifiers' accuracy ranges from 97% to 99%.

4.2.5. Results of banking news-events classification with TFIDF

Table 4.18 shows the accuracy, recall, and F-1 score of TFIDF language representation with various machine learning classifiers on banking news-events. The accuracy of the models is given in Table 4.19.

Table 4.18. Results of the machine learning classifiers with TFIDF

Classifier	Logistic Regression			Random Forest			Decision Tree			Linear SVC		
	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
Fraud	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Global	1.00	0.87	0.93	1.00	0.89	0.94	1.00	0.76	0.87	1.00	0.89	0.94
Governmental	1.00	1.00	1.00	1.00	1.00	1.00	0.85	1.00	0.92	1.00	1.00	1.00
Merger_Or_Acquisition	0.96	1.00	0.98	0.97	1.00	0.98	0.97	1.00	0.99	0.97	1.00	0.98
RBI_Policies	0.94	1.00	0.97	1.00	1.00	1.00	0.98	0.93	0.96	1.00	1.00	1.00
RatingsAgencies_Experts_View	1.00	0.96	0.98	1.00	0.96	0.98	1.00	1.00	1.00	1.00	0.89	0.94
Results	0.93	1.00	0.96	0.96	1.00	0.98	0.91	1.00	0.95	0.93	1.00	0.96

Table 4.19. Accuracy of the classifiers with TFIDF

Classifier	Accuracy
Logistic Regression	0.97
Random Forest	0.98
Decision Tree	0.95
Linear SVC	0.97

The Random Forest outperformed the other classifiers (Decision Tree, Linear SVC, Logistic Regression, and Random Forest) with an accuracy of 98%, as shown in Table 4.19. For the classifications Fraud, Global, Governmental, Merger Or Acquisition, RBI Policies, RatingsAgencies Experts View, and Results, the Random Forest achieved F1-scores of 1.00, 0.94, 1.00, 0.98, 1.00, 0.98, and 1.00, respectively. Table 4.18 compares all of the aforementioned classifiers in terms of Precision, Recall, and F-1 score for seven distinct classes of Banking News Events. Furthermore, the classifiers' accuracy ranges from 95% to 98%.

4.2.6. Results of banking news-events classification with a hybrid approach

Table 4.20 shows the accuracy, recall, and F-1 score of the proposed hybrid model and DistilBERT fine-tuned using Random Forest classifier on banking news-events. The accuracy results of both models are listed in Table 4.21.

Table 4.20. Results of the DistilBERT fine-tuned with Random Forest classifier and proposed Hybrid model

Approach	(Hybrid) DistilBERT+RandomForest+ Rules			DistilBERT+Rando mForest		
	P	R	F ₁	P	R	F ₁
Fraud	1.00	1.00	1.00	1.00	1.00	1.00
Global	1.00	1.00	1.00	1.00	1.00	1.00
Governmental	1.00	1.00	1.00	1.00	1.00	1.00
Merger_Or_Acquisition	0.98	1.00	0.99	0.97	1.00	0.98
RBI_Policies	1.00	1.00	1.00	1.00	1.00	1.00
RatingsAgencies_Experts_ View	1.00	0.98	0.99	1.00	0.96	0.98

Results	1.00	1.00	1.00	0.96	1.00	0.98
---------	------	------	------	------	------	------

Table 4.21. Accuracy of the DistilBERT fine-tuned with Random Forest classifier and proposed Hybrid model

Classifier	Accuracy
(Hybrid) DistilBERT+RandomForest+Rules	1.00
DistilBERT+RandomForest	0.99

Table 4.21 shows that the suggested Hybrid model outperformed the DistilBERT fine-tuned with Random Forest classifier. For the classifications Fraud, Global, Governmental, Merger Or Acquisition, RBI Policies, RatingsAgencies Experts View, and Results, the Hybrid model achieved F1-scores of 1.00, 1.00, 1.00, 0.99, 1.00, 0.99, and 1.00 respectively. Table 4.20 compares the precision, recall, and F-1 score of both methods for seven different classes of Banking News Events.

4.3. Sentiment Classification of Banking Financial News

In this part, DistilBERT and TF-IDF are compared on the sentiment classification of banking news events using different classifiers.

4.3.1. DistilBERT fine-tuned on banking news-events sentiments for classification with machine learning classifiers

Table 4.22 shows the precision, recall, and F-1 score of DistilBERT fine-tuned with several machine learning classifiers on banking news-events sentiments. The accuracy of the models is given in Table 4.23.

Table 4.22. Results of the DistilBERT fine-tuned on banking news-events sentiments with machine learning classifiers

Classifier	Logistic Regression			Random Forest			Decision Tree			Linear SVC		
	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
Negative	0.73	0.59	0.65	0.57	0.72	0.63	0.52	0.62	0.56	0.68	0.65	0.67
Neutral	0.85	0.92	0.88	0.88	0.88	0.88	0.78	0.84	0.81	0.82	0.69	0.75
Positive	0.68	0.77	0.72	0.88	0.74	0.81	0.88	0.74	0.81	0.70	0.82	0.75

The Random Forest performed best with F-1 score among the different classifiers Logistic Regression, Random Forest, Decision Tree, and Linear SVC, as shown in Table 4.23. Except for the ‘Negative’ class, the Random Forest has produced strong precision and recall for the ‘Neutral’ and ‘Positive’ classes, resulting in a higher F-1 score, which is even better for the ‘Negative’ class.

Table 4.23. Accuracy of the classifiers with DistilBERT

Classifier	Accuracy
Logistic Regression	0.76
Random Forest	0.78
Decision Tree	0.74
Linear SVC	0.73

In addition, the Random Forest outperformed the other classifiers (Logistic Regression, Random Forest, Decision Tree, and Linear SVC) with an accuracy of 78%, as shown in Table 4.24. Table 4.23 compares the Precision, Recall, and F-1 scores of all the stated classifiers for three distinct classes of Banking News-Events sentiments. The classifiers' accuracy ranges from 73% to 78%.

4.3.2. Results of Banking News-Events sentiments classification using machine learning classifiers with TF-IDF

Table 4.24 shows the precision, recall, and F-1 score of several machine learning classifiers using TFIDF language representation of banking news-events sentiments for classification. The accuracy of the models is listed in Table 4.25.

Table 4.24. Results of the machine learning classifiers with TF-IDF

Classifier	Logistic Regression			Random Forest			Decision Tree			Linear SVC		
	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
Negative	0.45	0.38	0.42	0.48	0.58	0.53	0.52	0.65	0.58	0.43	0.38	0.41
Neutral	0.76	0.67	0.71	0.76	0.76	0.76	0.77	0.70	0.73	0.82	0.70	0.75
Positive	0.70	0.84	0.76	0.85	0.74	0.79	0.79	0.71	0.75	0.65	0.79	0.71

The Random Forest performed best with F-1 score among the different classifiers Logistic Regression, Random Forest, Decision Tree, and Linear SVC, as shown in Table 4.24. Though with the Decision Tree on the ‘Negative’ class, precision and recall are on the upper side. Random Forest, on the other hand, has a better total F-1 score in both the ‘Positive and ‘Neutral’ classes.

Table 4.25. Accuracy of the classifiers with TF-IDF

Classifier	Accuracy
Logistic Regression	0.66
Random Forest	0.70
Decision Tree	0.69
Linear SVC	0.65

In addition, the Random Forest outperformed the other classifiers (Logistic Regression, Random Forest, Decision Tree, and Linear SVC) with an accuracy of 70%, as shown in Table 4.25. Table 4.25 compares the Precision, Recall, and F-1 scores of all the stated classifiers for three distinct classes of Banking News-Events sentiments. The classifiers' accuracy ranges from 65% to 70%.

4.4. Named entities recognition and analysis of banking stocks and the banking index in Indian Stock market based on past and current linguistic or sentiments of news events

The introduction of Google’s BERT [57], which has been regarded as the start of a new era in NLP, is one of the most significant milestones in the advancement of NLP recently. The HuggingFace transformers library in python is used to fine-tune a pre-trained BERT model for identifying named entities in banking news events. We just need to fine-tune BERT in 2-4 epochs, although it is quite huge, sophisticated, and has millions of parameters. Because BERT was trained on a large quantity of data and has already encoded a lot of information about our

language, this outcome is possible. BERT is one of the most powerful NLP models accessible right now, because of its impressive performance obtained in a short length of time with a little quantity of data.

The CoNLL Dataset [193] is used to demonstrate Named Entity Recognition. The input text files for our datasets are in the CoNLL format. We need to make an optimizer to fine-tune our Bert Classifier. The hyper-parameters are used: batch size 16, learning rate $4e-5$, and epochs 2. We achieve 90.61% accuracy with a precision of 90% and recall 91.21%.

The dataset in the event study covers the news events on the banks or banking stock index comprises both private and public sectors banks listed in the Indian equity markets; AU Small Finance, Axis Bank, Bank of Baroda, Bank of India, Canara Bank, HDFC Bank, ICICI Bank, IDBI, Indian Bank, IndusInd Bank, Indian Overseas Bank, Punjab National Bank, RBL Bank, State Bank of India, South Indian Bank, UCO Bank, Union Bank, and Yes Bank. We split our overall dataset into three subsamples to see if different news events with a negative polarity, a positive polarity, or a neutral polarity influence market reactions to banking news events.

Table 4.26: The impact of banking news events with different polarities or tones of communication on Bank stock or Bank Nifty returns

Event window	Negative news events		Positive news events		Neutral news events	
	Mean CAR	t-value	Mean CAR	t-value	Mean CAR	t-value
(-60,60)	-0.0513	-1.565	0.0356	0.516	0.0180	0.685
(-30,30)	-0.0517**	-2.353	-0.0356	-1.499	0.0267	1.298
(-5,5)	-0.0004	-0.017	-0.0105	-0.846	0.0343*	1.678
(-1,1)	-0.0148***	-3.615	-0.0027	-0.522	0.0321	1.451
(-60,-1)	-0.0264	-1.213	0.0566	0.796	0.0101	0.495
(-30,-1)	-0.0273*	-1.670	-0.0011	-0.062	0.0125	1.048
(-5,-1)	-0.0269	0.544	0.0058	0.778	0.0130**	2.019
(-1,D)	0.0061***	-3.046	0.0023	0.542	0.0272	1.314
(D,1)	-0.0088***	-2.650	-0.0048	-1.379	0.0050	1.341
(D,5)	-0.0111	-1.381	-0.0172***	-2.792	-0.0026	-0.427
(D,30)	-0.0248**	-2.141	-0.0287*	-1.752	-0.0128	-0.930
(D,60)	-0.0291	-1.612	-0.0047	-0.191	-0.0126	-0.642

Note: *, **, *** denote the statistical significance at 10%, 5% and 1%, respectively

In Table 4.26, we find highly statistically significant mean CARs of 0.6%, -0.8%, and -1.48% in the event windows (-1, D), (D, 1), and (-1, -1), respectively, surrounding the publication day of the banking news event. Following the immediate announcement of the negative banking news event, negative CARs were found in the event window (D, 1), and (-1, -1). It means the investors have reacted in the same direction as the tone of news events. We also find a statistically significant mean CAR of -2.48% in the event window of (D,30). It means that the effect of the negative banking news events sustained for one month period after the publication of news events. The significant mean CAR of -5.13% is shown in the event window of (-30,30). Therefore, it is interpreted that the negative banking news events can make a negative effect on the banking stock index for a short to medium time.

Banking stock indexes, on the other hand, do not appear to counter to positive and neutral banking news events, as seen by mean CARs that are not statistically significant in all event windows except we get statistically significant negative CARs of -1.72% and -2.87% in the event window of (D,5) and (D,30), respectively, even after the publication of positive banking news events. However, the mean CARs are positive before the publication of banking news events with a positive tone of communication in all the event windows (-1, D), (-5,-1), (-30,-1), and (-60,-1). Moreover, the mean CARs convert to negative for all the event windows after the publication of positive tone of the banking news event (D,1), (D,5), (D,30), (D,60), but not the statistically significant. We can still take this as evidence that the banking index can predict news events with a positive tone before they occur but significant. The findings in Table 4.26 show that the banking news events' polarity has a significant influence on banking index returns; negative news, as compared to positive and neutral news events, seems to be informational.

To examine the possible influence of tone, we construct three different subgroups of each news event: negative (Fraud, Global, Governmental, Merger or Acquisition, RBI Policies, Ratings Agencies or Expert's View, and Results), positive (Fraud, Global, Governmental, Merger or Acquisition, RBI Policies, Ratings Agencies or Expert's View, and Results), and neutral (Fraud, Global, Governmental, Merger or Acquisition, RBI Policies, Ratings Agencies or Expert's View, and Results). Due to the lack of such a category of news items in the data set, data for one or more news events is missing in each subsample. Due to the lack of news items with particular polarity in the data set, data for one or more news events are missing in each subsample.

Table 4.27: The impact of Negative polarity or tone of online banking news events publication on Bank stock or Bank Nifty returns

Event window	Mean CARs				
	Results	RBI_Policies	Fraud	RatingsAgencies_Experts_View	Governmental
(-60,60)	0.0737	0.1626***	-0.2378***	-0.0418	0.0082***
(-30,30)	-0.0847	0.0532***	-0.1857***	0.0198	0.0026
(-5,5)	-0.0380**	0.0114	-0.0317	0.0369*	-0.0016
(-1,1)	-0.0033	-0.0101**	-0.0254***	-0.0128*	-0.0069***
(-60,-1)	0.0230	0.1449***	-0.1438***	-0.0267	-0.0056
(-30,-1)	-0.1032**	0.0718**	-0.1238***	0.0328*	-0.0039
(-5,-1)	-0.0195	0.0035	0.0238	0.0255**	-0.0016
(-1,D)	-0.0050	-0.0102***	-0.0027	-0.0065**	-0.0100
(D,1)	0.0009	0.0001	-0.0231***	-0.0061	0.0031
(D,5)	-0.0114	0.0193	-0.0607***	0.0158*	0.0100
(D,30)	0.0298	-0.0065	-0.0796***	-0.0101	0.0170*
(D,60)	0.0747	0.0157	-0.1180***	-0.0150	0.0254***

Note: *, **, *** denote the statistical significance at 10%, 5% and 1%, respectively

The findings in Table 4.27 support and expand on earlier findings, indicating that the tone of news events has a significant influence on stock market returns. The greater influence of negative ‘Fraud’ news event on bank stock or the bank index returns tends to be in almost all the event windows (-60,60), (-30,30), (-1,1), (-60,-1), (-30,-1), (-1,D), (D,1), (D,5), (D,30), (D,60) that exhibit statistically significant mean CARs equal to -23.78%, -18.57%, -2.54%, -14.38%, -12.38%, -0.27%, -2.31%, -6.07%, -7.96%, -11.80% respectively. It indicates that the impact of negative ‘Fraud’ news events lasted for a long time after the news had been published.

The significant outcomes before the publication of any of the news events are interpreted as confirmation that the stock market may predict news with negative sentiments. Moreover, the negative CARs are observed in almost all the event windows before the date of publication of all negative news events shown in Table 4.27. Though it is statistically significant results seen before the publication of news articles on RBI policies, however, the negative CARs are seen only around the date of publication with CAR -1.02%, -1.01% in the

event window (-1, D), and (-1,1) respectively. It means the negative tone of news articles on RBI policies may impact returns for 1-2 trading sessions around the news publication day. It may not impact the returns much at any time. Similarly, the negative CARs are seen before the publication of negative news on Governmental policies but with only significant CAR -0.69% in the event window (-1, 1). The impact of the negative news on the Results event is seen in banking stock returns for a short time. The negative significant CARs are found in the symmetric event window (-5,5), and another event window (-30,-1) with CARs -3.8% and -10.32%, respectively. It gives a piece of evidence that the stock market may anticipate the negative quarterly or annual results of the banking stocks and it can impact the returns for a short time. It is also shown that the impact of news publication on negative views of Rating agencies and Experts on the banking stocks is not statistically significant.

Table 4.28: The impact of Positive polarity or tone of online banking news events publication on Bank stock or Bank Nifty returns

Event window	Mean CARs				
	Results	RBI_Policies	Merger_Or_Acquisition	RatingsAgencies_Experts_View	Governmental
(-60,60)	0.1044	0.2360***	-0.0570	-0.1198	0.0061***
(-30,30)	-0.0463	0.0753	-0.1142**	-0.0229	-0.0019
(-5,5)	-0.0356*	0.0121	-0.0348**	0.0498	0.0048**
(-1,1)	-0.0033	0.0116***	-0.01087	0.0022	-0.0051
(-60,-1)	0.1286	0.0932	-0.0692	0.0295	-0.0172***
(-30,-1)	-0.0149	0.0146	-0.0354	0.0593	-0.0044
(-5,-1)	0.0017	-0.0024	-0.0273	0.0324***	0.0033
(-1,D)	0.0040	0.0097	0.0100	0.0021	-0.0108*
(D,1)	-0.0071	0.0020	-0.0199***	0.0000	0.0058
(D,5)	-0.0374***	0.0047	-0.0365***	0.0121*	0.0123
(D,30)	-0.02299	0.0462	-0.0768***	-0.0774*	0.0135
(D,60)	-0.00419	0.1032**	0.0376	-0.1260*	0.0379***

Note: *, **, *** denote the statistical significance at 10%, 5% and 1%, respectively

As Table 4.26 shows that banking stock indexes do not appear to counter to positive and neutral banking news events, as evidenced by mean CARs that are not statistically significant in all

event windows, except for event windows (D,5) and (D,30) with statistically significant negative CARs of -1.72% and -2.87%, even after the publication of positive banking news events. It means stock markets anticipate the positive news and investors book profits after the online publication of positive news and positive reaction does not sustain after the publication of positive tone of news events. This is also supported with the evidence of publication of positive tone of news event on 'Results', as indicated in Table 4.28. The statistically significant negative CAR -3.74% is shown in the event window of (D,5).

It is also indicated, in Table 4.28, that even the positive tone of news published on event 'Merger or Acquisition', the statistically significant negative CARs are seen in the event windows (D,1), (D,5), (D,30) after the date of news publication with CAR -1.99%, -3.65%, -7.68%, respectively. The negative impact of even positive tone of news event 'Merger or Acquisition' increases from a day of an online news publication or announcement and sustains for a month. The negative CARs are also seen before the publication of news on event 'Merger or Acquisition', moreover, statistically significant in the symmetric event windows (-5,5) and (-30, 30) with CAR -3.48%, and -11.42%, respectively. It means stock markets react negatively to pre or post-announcement of news events on 'Merger or Acquisition' in banks for the short to medium term. However, the CAR turns positive with 3.76% in the event window (D,60) i.e. 2 months post-announcement of banks' mergers, also suggested by [217] in the post-announcement analysis of bank mergers. Another paper demonstrates on European banks' mergers that target bank shares have significant positive abnormal returns, but bidding bank shareholders have no significant abnormal returns [218].

Whenever, there are positive announcements on banks' by Ratings agencies or Experts, the banking stock index tends to appear statistically significant but with negative CARs -7.74%, and -12.60% in the event window (D,30), and (D,60), respectively. Before the announcements or ratings by the agencies or experts on the banking stocks or index has made published online, banking stocks or banking index has shown positive CARs that were too statistically significant in the event window (-5,-1) with CAR 3.24% and other event windows too but not statistically significant. It means when banking stock or index has been shown a positive trend, the rating agencies or experts give their positive announcements. However, stock or index tends to give negative returns post-announcements by Market experts even with positive sentiments.

It is noticeable that announcements or news publications on events 'RBI_Policies' and 'Governmental' with a positive tone or sentiment have a significant positive impact on banking

stocks or banking index. Table 4.28 has shown positive CARs in all the event windows after the online publication of news articles with a positive tone or sentiments on RBI policies and Governmental policies, moreover, statistically significant in the event window (D,60) with CARs 10.32%, and 3.79%, respectively. It shows that banking stocks or the banking index perform aligned with positive tone or sentiments of news events on RBI_Policies and national Government.

Table 4.29: The impact of Neutral polarity or tone of online banking news events publication on Bank stock or Bank Nifty returns

Event window	Mean CARs			
	RBI_Policies	Merger_Or_Acquisition	RatingsAgencies_Experts_View	Governmental
(-60,60)	0.1634***	-0.1139***	0.1659	0.0141***
(-30,30)	0.0877***	-0.0005	0.0333	0.0107**
(-5,5)	0.0213**	0.0803	-0.0198	0.0067**
(-1,1)	0.0037	0.0811	0.0059	0.0016
(-60,-1)	0.1414***	-0.0842***	0.0746	0.0021
(-30,-1)	0.0835***	-0.0277	0.0132	0.0079*
(-5,-1)	0.0101*	0.0283*	-0.0184	0.0082***
(-1,D)	0.0004	0.0735	-0.0046**	0.0013
(D,1)	0.0032*	0.0081	0.0104	0.0003
(D,5)	0.0104**	-0.0123	0.0021	-0.0025
(D,30)	-0.0022	-0.0412	0.0227	0.0012
(D,60)	0.0039	-0.0802**	0.1116	0.0122**

Note: *, **, *** denote the statistical significance at 10%, 5% and 1%, respectively

The results shown in Table 4.29 are interesting to discuss for all the news events mentioned in the table. Though Table 4.26 shows that the banking stock indexes do not appear to react to neutral and positive banking news events collectively. However, Table 4.29 shows that banking stocks or the banking index perform positively with positive CARs pre and post-announcements with even neutral sentiments or tone of communication in almost all the event windows, especially statistically significant in the event windows of the pre-online publication of news as well as symmetric event windows for RBI_Policies and Governmental news events.

However, banking stocks do not show abnormal returns during online publication of neutral views or announcements by Market Experts and Rating Agencies on banking stocks except one event window pre-publication date (-1, D) with -0.46% CAR. But, even neutral sentiments or tone of communication on banks' mergers or acquisitions strongly impact the markets. It has shown negative CARs post-announcements in all the event windows, especially statistically significant in the event window (D,60) with CAR -8.02%. Though only the news articles with positive and neutral sentiments or tone of communication available for news event 'Merger_Or_Acquisition' in the data set, the banking stocks or the banking index has reacted negatively with statistically significant abnormal returns, as shown in Table 4.28 and Table 4.29.

Chapter-5

Conclusion and future scope

This chapter presents the proposed research's results and future scope. The following is how this chapter is organized: The conclusions are presented in Section 5.1. The thesis' future work is described in Section 5.2.

5.1.Conclusions

Economists have performed several studies to see whether there are any links between news events and stock returns. According to economists, banking stocks can display abnormal behavior in response to socio-political events, bank risks, and other news events. Every day, hundreds of events are published as articles on web news portals or corporate websites, necessitating the need to automatically arrange and interpret them. Natural Language Processing has been studied in various ways as machine learning algorithms and their problem-solving capabilities have advanced. Event Extraction and Classification, a method of gathering data on recurring events included in documents and automatically extracting information about what happened and when is one of its most popular uses. However, due to the unstructured nature of the original data, data transformation is mind-numbing.

We begin with phase-1 to the extraction of banking news from a pool of financial news articles. This multi-class Financial News classification facilitates the retrieval of banking-related news. For the Indian stock market, developing a system for acquiring banking news and other important areas is a big and untested challenge. We're looking for news from Indian banks, the Indian government, and the rest of the world. We use a systematic technique to split the news into domains of our choice, classifying the articles into four categories. To fulfill the aim, news articles are gathered from a variety of online news sources and tagged to extract banking and other relevant news. To classify the news articles into four categories, five standard machine learning classifiers, one neural network classifier, and four ensemble classifiers are utilized to automate the classification process (Banking, Governmental, Global, and Non-Banking). We utilized a variety of approaches to align the data set between classes since our data set has a class imbalance, and the classifier output is assessed using the original unbalanced and balanced data set. The classification models were evaluated using precision, recall, F-1, and accuracy metrics. The findings show that Random Forest with balanced data

utilizing SMOTE has the greatest classification accuracy of 100%, while other models have lower classification accuracy of even 34%.

Further, we present phase-2 with a unique hybrid approach for classifying banking news articles into 7 events (Fraud, Global, Governmental, Merger Or Acquisition, RBI Policies, RatingsAgencies Experts View, and Results) to automate the classification process. The suggested hybrid model using the DistilBERT language representation model fine-tuned with Random Forest classifier and Rules attained the best accuracy of 100%, as seen by the results.

The phase-3 aims to classify the sentiments of banking news events into three categories: positive, negative, and neutral. On this sentiment classification job, DistilBERT with fine-tuning is compared to conventional TF-IDF. The main goal was to see how much the deep contextual pre-trained language representation might help with this task. DistilBERT outperformed TF-IDF across the field with all four machine learning classifiers. DistilBERT with Random Forest has an accuracy of 78 percent, which is 7% higher than Random Forest with TF-IDF. Random Forest with DistilBERT was also shown to outperform other classifiers such as Logistic Regression, Decision Tree, and Linear SVC. When compared to TFIDF, DistilBERT had better precision and recall for all classes. In conclusion, when a model's transfer capability is a priority, contextual neural models are preferable to conventional methods, despite significantly longer training periods and memory requirements.

Because news events or online news articles can contribute to the building of investor sentiment, therefore online media has a considerable influence on financial markets. This phase-4 explores the connections between online news events and stock market returns, concentrating on the influence of a selection of banking and government news events reported in major online news portals between 2018 and 2021 on abnormal stock returns. The study takes into account the potential that the type of news event and the polarity or tone of communication impact the returns of banking stocks.

The results indicate that the effect of the negative banking news events sustained for one month period after the publication of news events. The significant mean CAR of -5.13% is shown in the symmetric event window of (-30,30). It also provides evidence that the stock market may expect unfavorable quarterly or annual results from banking firms and that this might have a short-term influence on returns. Significant outcomes before the event date are interpreted as confirmation that stock markets may predict negative sentiment news events. This confirms earlier studies that show stock markets behave differently before, during, and

after the announcement of negative news. We also found evidence that banking stocks or the banking index react more negatively to the negative tone or sentiments of the banking news events.

Our findings have some significance. To begin with, banking index overreactions to negative news suggest that the market is concerned about good and effective government and banking policies. News publications with a positive or even neutral tone or sentiment on the events ‘RBI Policies’ and ‘Governmental’ have a significant positive impact on banking stocks or banking indexes, which is statistically significant in the event window (D,60) with CARs of 10.32% and 3.79%, respectively in case of positive news events. However, investors dislike merger-related news for even thriving banks, according to the study, therefore they prefer to sell their stocks. Furthermore, even with positive sentiments, stock or index returns tend to be negative following market experts’ statements.

5.2.Future scope

Based on our results, our trained classification model can be used to classify the news into other specific domains by training the model on data sets of those domains. The labeling of the dataset is done manually at the current stage of our study, with the help of domain experts. In our future research, including those listed in this thesis, we may also use certain recently introduced methods and frameworks for classifying data with a larger volume.

More training and test data may be acquired in the future to generalize the sentiment classification findings in the banking news domain. Different tests might be used to compare the impacts more easily. Furthermore, dictionary-based rules for each class of positive, negative, and neutral may be developed to let DistilBERT or other pre-trained transformer-based models classify data. We can also perform the event-study of private and public sector banks separately.

References

1. Atkins A, Niranjana M, Gerding E (2018) Financial news predicts stock market volatility better than close price. *J Financ Data Sci* 4:120–137. <https://doi.org/10.1016/j.jfds.2018.02.002>
2. Zhang Y, Dang Y, Chen H, et al (2009) Automatic online news monitoring and classification for syndromic surveillance. *Decis Support Syst* 47:508–517. <https://doi.org/10.1016/j.dss.2009.04.016>
3. Robinson J, Glean A, Moore W (2018) How does news impact on the stock prices of green firms in emerging markets? *Res Int Bus Financ* 45:446–453. <https://doi.org/10.1016/j.ribaf.2017.07.176>
4. Löffler G, Norden L, Rieber A (2016) Salient News and the Stock Market Impact of Tone in Rating Reports. *Ssrn* 1–41. <https://doi.org/10.2139/ssrn.2802570>
5. Elagamy MN, Stanier C, Sharp B (2018) Stock market random forest-text mining system mining critical indicators of stock market movements. *2nd Int Conf Nat Lang Speech Process ICNLSP 2018* 1–8. <https://doi.org/10.1109/ICNLSP.2018.8374370>
6. West JC, Clarke DE, Duffy FF, et al (2016) Availability of Mental Health Services Prior to Health Care Reform Insurance Expansions. *Psychiatr Serv* 67:983–989. <https://doi.org/10.1176/appi.ps.201500423>
7. Ali M, Khalid S, Rana MI, Azhar F (2018) A probabilistic framework for short text classification. *2018 IEEE 8th Annu Comput Commun Work Conf CCWC 2018* 2018-Janua:742–747. <https://doi.org/10.1109/CCWC.2018.8301712>
8. Romero C, Ventura S (2007) Educational data mining: A survey from 1995 to 2005. *Expert Syst Appl* 33:135–146. <https://doi.org/10.1016/j.eswa.2006.04.005>
9. Ye Q, Zhang Z, Law R (2009) Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. *Expert Syst Appl* 36:6527–6535. <https://doi.org/10.1016/j.eswa.2008.07.035>
10. Thorleuchter D, Van Den Poel D (2012) Predicting e-commerce company success by mining the text of its publicly-accessible website. *Expert Syst Appl* 39:13026–13034. <https://doi.org/10.1016/j.eswa.2012.05.096>

11. Goel T, Lewrick U, Tarashev N (2020) Bank capital allocation under multiple constraints. *J Financ Intermediation* 44:. <https://doi.org/10.1016/j.jfi.2019.100844>
12. Ho KY, Shi Y, Zhang Z (2020) News and return volatility of Chinese bank stocks. *Int Rev Econ Financ* 69:1095–1105. <https://doi.org/10.1016/j.iref.2018.12.003>
13. Sharma D, Verma R (2020) Reaction of Stock Price to Frauds' Announcements: Evidence from Indian Banking Sector. *Asia-Pacific J Manag Res Innov* 16:157–166. <https://doi.org/10.1177/2319510x20930879>
14. Park J, Ha S, Chang FK (2009) Monitoring impact events using a system-identification method. *AIAA J* 47:2011–2021. <https://doi.org/10.2514/1.34895>
15. Flohrer T, Krag H, Klinkrad H (2009) ESA's process for the identification and assessment of high-risk conjunction events. *Adv Sp Res* 44:355–363. <https://doi.org/10.1016/j.asr.2009.04.012>
16. Dimpfl T (2011) The impact of US news on the German stock market-An event study analysis. *Q Rev Econ Financ* 51:389–398. <https://doi.org/10.1016/j.qref.2011.07.005>
17. Erickson M, Wang SW (1999) Earnings management by acquiring firms in stock for stock mergers. *J Account Econ* 27:149–176. [https://doi.org/10.1016/S0165-4101\(99\)00008-7](https://doi.org/10.1016/S0165-4101(99)00008-7)
18. Lam W, Ho KS (2001) FIDS: An intelligent financial web news articles digest system. *IEEE Trans Syst Man, Cybern Part A Systems Humans* 31:753–762. <https://doi.org/10.1109/3468.983433>
19. Hogenboom F (2014) Automated Detection of Financial Events in News Text
20. Chen J, Huang H, Tian S, Qu Y (2009) Feature selection for text classification with Naïve Bayes. *Expert Syst Appl* 36:5432–5435. <https://doi.org/10.1016/j.eswa.2008.06.054>
21. Armanfard N, Reilly JP, Komeili M (2016) Local Feature Selection for Data Classification. *IEEE Trans Pattern Anal Mach Intell* 38:1217–1227. <https://doi.org/10.1109/TPAMI.2015.2478471>
22. Abbasi A, France S, Zhang Z, Chen H (2011) Selecting attributes for sentiment classification using feature relation networks. *IEEE Trans Knowl Data Eng* 23:447–

462. <https://doi.org/10.1109/TKDE.2010.110>
23. Majumder N, Politénico I, Soujanya Poria N, et al (2017) AFFECTIVE COMPUTING AND SENTIMENT ANALYSIS Deep Learning-Based Document Modeling for Personality Detection from Text. *IEEE Intell Syst* (Vol 32 , Issue 2 , Mar-Apr 2017) 32:
 24. Mazyad A, Teytaud F, Fonlupt C (2018) A comparative study on term weighting schemes for text classification. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer Verlag, pp 100–108
 25. Gomez JC, Moens M (2012) PCA document reconstruction for email classification. *Comput Stat Data Anal* 56:741–751. <https://doi.org/10.1016/j.csda.2011.09.023>
 26. Makazhanov A, Rafiei D (2013) Predicting political preference of Twitter users. *Proc 2013 IEEE/ACM Int Conf Adv Soc Networks Anal Min - ASONAM '13* 298–305. <https://doi.org/10.1145/2492517.2492527>
 27. Phu VN, Tran VTN, Chau VTN, et al (2017) A decision tree using ID3 algorithm for English semantic analysis. *Int J Speech Technol* 20:593–613. <https://doi.org/10.1007/s10772-017-9429-x>
 28. Gandomi A, Haider M (2015) Beyond the hype: Big data concepts, methods, and analytics. *Int J Inf Manage* 35:137–144. <https://doi.org/10.1016/j.ijinfomgt.2014.10.007>
 29. SUN Y, WONG AKC, KAMEL MS (2009) Classification of Imbalanced Data: a Review. *Int J Pattern Recognit Artif Intell* 23:687–719. <https://doi.org/10.1142/S0218001409007326>
 30. Manne S, Fatima SS (2011) A Novel Approach for Text Categorization of Unorganized data based with Information Extraction. *Int J Comput Sci Eng* 3:2846–2854
 31. Harish BS, Guru DS, Manjunath S (2010) Representation and classification of text documents: A brief review. *IJCA, Spec Issue Recent Trends Image Process Pattern Recognit* 110–119

32. Liang Y, Liu Y, Kwong CK, Lee WB (2012) Learning the “whys”: Discovering design rationale using text mining - An algorithm perspective. *CAD Comput Aided Des* 44:916–930. <https://doi.org/10.1016/j.cad.2011.08.002>
33. Liu B, Li X, Lee WS, Yu PS (2004) Text classification by labeling words. *Artif Intell* 34:425–430
34. Zhou DY, Bousquet O, Lal TN, et al (2004) Learning with local and global consistency. *Adv Neural Inf Process Syst* 16 16:321–328
35. Bengio Y, Ducharme R, Vincent P (2001) A neural probabilistic language model. *Adv Neural Inf Process Syst* 1–11
36. Hochreiter S, Schmidhuber J (1997) Long Short-Term Memory. *Neural Comput* 9:1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
37. Lecun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521:436–444. <https://doi.org/10.1038/nature14539>
38. Harish BS, Revanasiddappa MB (2018) A New Approach of Categorizing Text Documents through Consensus Clustering. 87–91. <https://doi.org/10.1145/3233347.3233371>
39. Man Lan, Chew Lim Tan, Jian Su, Yue Lu (2009) Supervised and Traditional Term Weighting Methods for Automatic Text Categorization. *IEEE Trans Pattern Anal Mach Intell* 31:721–735. <https://doi.org/10.1109/TPAMI.2008.110>
40. Billal B, Fonseca A, Sadat F (2016) Named Entity Recognition and Hashtag Decomposition to Improve the Classification of Tweets. 64–73
41. Schumaker RP, Chen H (2009) A quantitative stock prediction system based on financial news. *Inf Process Manag* 45:571–583. <https://doi.org/10.1016/j.ipm.2009.05.001>
42. Chen Y, Liqiang L, X N, et al (2012) Supervised Bayesian Network Model for Microblog Topic Classification. 1:561–576
43. Xia R, Zong C, Hu X, Cambria E (2015) Feature ensemble plus sample selection: Domain adaptation for sentiment classification. *IJCAI Int Jt Conf Artif Intell 2015-Janua*:4229–4233. <https://doi.org/10.1109/MIS.2013.27>

44. Cambria E (2016) Affective Computing and Sentiment Analysis. *IEEE Intell Syst* 31:102–107. <https://doi.org/10.1109/MIS.2016.31>
45. Haibo He, Member, IEEE and EAG (2019) Learning from imbalanced data. *IEEE Trans Knowl Data Eng* 21:. https://doi.org/10.1007/978-3-030-04663-7_4
46. Moreo A, Esuli A, Sebastiani F (2016) Distributional random oversampling for imbalanced text classification. *SIGIR 2016 - Proc 39th Int ACM SIGIR Conf Res Dev Inf Retr* 805–808. <https://doi.org/10.1145/2911451.2914722>
47. Cao P, Zhao D, Zaiane O (2013) An optimized cost-sensitive SVM for imbalanced data learning. *Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics)* 7819 LNAI:280–292. https://doi.org/10.1007/978-3-642-37456-2_24
48. Kaur H, Pannu HS, Malhi AK (2019) A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Comput Surv* 52:. <https://doi.org/10.1145/3343440>
49. Tayyar Madabushi H, Kochkina E, Castelle M (2019) Cost-Sensitive BERT for Generalisable Sentence Classification on Imbalanced Data. 125–134. <https://doi.org/10.18653/v1/d19-5018>
50. Tetlock PC (2007) Giving content to investor sentiment: The role of media in the stock market. *J Finance* 62:1139–1168. <https://doi.org/10.1111/j.1540-6261.2007.01232.x>
51. Xing FZ, Cambria E, Welsch RE (2018) Natural language based financial forecasting: a survey. *Artif Intell Rev* 50:49–73. <https://doi.org/10.1007/s10462-017-9588-9>
52. Li X, Xie H, Chen L, et al (2014) News impact on stock price return via sentiment analysis. *Knowledge-Based Syst* 69:14–23. <https://doi.org/10.1016/j.knosys.2014.04.022>
53. Shapiro AH, Wilson DJ (2019) Taking the Fed at its Word: A New Approach to Estimating Central Bank Objectives Using Text Analysis. *Fed Reserv Bank San Fr Work Pap Ser* 01–74. <https://doi.org/10.24148/wp2019-02>
54. Shapiro AH, Sudhof M, Wilson D, Shapiro AH (2018) FEDERAL RESERVE BANK OF SAN FRANCISCO Measuring News Sentiment Measuring News Sentiment *.

55. MacKinlay AC (1997) Event Studies in Economics and Finance. *J Econ Lit* 35:13–39
56. Sanh V, Debut L, Chaumond J, Wolf T (2019) DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. 2–6
57. Kenton MC, Kristina L, Devlin J (2018) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
58. Sebastiani F (2002) Machine Learning in Automated Text Categorization. 34:1–47
59. Villena-Román J, Collada-Pérez S, Lana-Serrano S, González-Cristóbal JC (2011) Hybrid approach combining machine learning and a rule-based expert system for text categorization. *Proc 24th Int Florida Artif Intell Res Soc FLAIRS* - 24 323–328
60. Solovyev V, Ivanov V (2016) Knowledge-Driven Event Extraction in Russian: Corpus-Based Linguistic Resources. *Comput Intell Neurosci* 2016:.
<https://doi.org/10.1155/2016/4183760>
61. Hirsch L, Saeedi M, Hirsch R (2005) Evolving rules for document classification. *Lect Notes Comput Sci* 3447:85–95. https://doi.org/10.1007/978-3-540-31989-4_8
62. Aubaid AM, Mishra A (2020) A rule-based approach to embedding techniques for text document classification. *Appl Sci* 10:. <https://doi.org/10.3390/app10114009>
63. Zhang Y, Jin R, Zhou ZH (2010) Understanding bag-of-words model: A statistical framework. *Int J Mach Learn Cybern* 1:43–52. <https://doi.org/10.1007/s13042-010-0001-0>
64. Joseph Turian, Lev Ratinov YB (2010) Word representations: A simple and general method for semi-supervised learning. *Proc 48th Annu Meet Assoc Comput Linguist Assoc Comput Linguist* 384–394
65. Sivic, Zisserman (2003) Video Google: a text retrieval approach to object matching in videos. *Proc Ninth IEEE Int Conf Comput Vis* 1470–1477 vol.2.
<https://doi.org/10.1109/ICCV.2003.1238663>
66. Li-Ping Jing, Hou-Kuan Huang, Hong-Bo Shi (2003) Improved feature selection approach TFIDF in text mining. 944–946. <https://doi.org/10.1109/icmlc.2002.1174522>

67. Montañés E, Díaz I, Ranilla J, et al (2005) Scoring and Selecting Terms for Text Categorization. *IEEE Intell Syst* 20:40–47. <https://doi.org/10.1109/MIS.2005.49>
68. Bengio, Yoshua, Réjean Ducharme, Pascal Vincent CJ (2003) A Neural Probabilistic Language Model. *J Mach Learn Res* 1137–1155
69. Xiong Y, Chen S, Qin H, et al (2020) Distributed representation and one-hot representation fusion with gated network for clinical semantic textual similarity. *BMC Med Inform Decis Mak* 20:1–7. <https://doi.org/10.1186/s12911-020-1045-z>
70. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. *1st Int Conf Learn Represent ICLR 2013 - Work Track Proc* 1–12
71. Liu B (2020) Text sentiment analysis based on CBOW model and deep learning in big data environment. *J Ambient Intell Humaniz Comput* 11:451–458. <https://doi.org/10.1007/s12652-018-1095-6>
72. Schwenk H (2007) Continuous space language models. *Comput Speech Lang* 21:492–518. <https://doi.org/10.1016/j.csl.2006.09.003>
73. Mikolov T, Sutskever I, Chen K, et al (2013) Distributed Representations of Words and Phrases and their Compositionality. In: Burges CJC, Bottou L, Welling M, et al (eds) *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., pp 3111–3119
74. Bojanowski P, Grave E, Joulin A, Mikolov T (2016) Enriching Word Vectors with Subword Information. 5:135–146
75. Jeffrey Pennington, Richard Socher CDM (2014) GloVe: Global Vectors for Word Representation. *Proc 2014 Conf Empir Methods Nat Lang Process (EMNLP), Assoc Comput Linguist* 1532–1543
76. Wei CP, Lee YH (2004) Event detection from online news documents for supporting environmental scanning. *Decis Support Syst* 36:385–401. [https://doi.org/10.1016/S0167-9236\(03\)00028-9](https://doi.org/10.1016/S0167-9236(03)00028-9)
77. Zhu Q, Li X, Conesa A, Pereira C (2018) GRAM-CNN: A deep learning approach with local context for named entity recognition in biomedical text. *Bioinformatics*

- 34:1547–1554. <https://doi.org/10.1093/bioinformatics/btx815>
78. Prokhorov V, Pilehvar MT, Kartsaklis D, et al (2018) Unseen Word Representation by Aligning Heterogeneous Lexical Semantic Spaces
 79. Bosco C, Patti V, Bolioli A (2013) Developing Corpora for Sentiment Analysis: The Case of Irony and Senti-TUT. *Ieee Intell Syst* 55–63
 80. Sutskever I, Vinyals O, Le Q V. (2014) Sequence to sequence learning with neural networks. *Adv Neural Inf Process Syst* 4:3104–3112
 81. Minaee S (2020) Deep Learning Based Text Classification: A Comprehensive Review. 1:1–42
 82. X. Zhu, P. Sobihani, and H. Guo (2015) “Long short-term memory over recursive structures.” *Int Conf Mach Learn* 1604–1612
 83. P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao and BX (2016) “Text classification improved by integrating bidirectional lstm with two-dimensional max pooling.” *arXiv*
 84. Wei W, Li J, Cao L, et al (2013) Effective detection of sophisticated online banking fraud on extremely imbalanced data. *World Wide Web* 16:449–475. <https://doi.org/10.1007/s11280-012-0178-0>
 85. Saleh I. Alfuraih, Nien T. Sui DM (2004) Using Trusted Email to Prevent Credit Card Frauds in Multimedia Products. 245–256. <https://doi.org/10.1023/A:1020940830716>
 86. Peters ME, Neumann M, Iyyer M, et al (2018) Deep contextualized word representations. *NAACL HLT 2018 - 2018 Conf North Am Chapter Assoc Comput Linguist Hum Lang Technol - Proc Conf* 1:2227–2237. <https://doi.org/10.18653/v1/n18-1202>
 87. Dai Z, Yang Z, Yang Y, et al (2020) Transformer-XL: Attentive language models beyond a fixed-length context. *ACL 2019 - 57th Annu Meet Assoc Comput Linguist Proc Conf* 2978–2988. <https://doi.org/10.18653/v1/p19-1285>
 88. Howard J Universal Language Model Fine-tuning for Text Classification
 89. Sun Y, Wang S, Li Y, et al ERNIE : Enhanced Representation through Knowledge Integration

90. Ye Z, Guo Q, Gan Q, et al (2019) BP-Transformer: Modelling Long-Range Context via Binary Partitioning
91. Aip R Neural Attentive Bag-of-Entities Model for Text Classification
92. Magnini B, Strapparava C, Pezzulo G, Gliozzo A (2002) The role of domain information in Word Sense Disambiguation. *Nat Lang Eng* 8:359–373.
<https://doi.org/10.1017/S1351324902003029>
93. Bilenko M, Mooney R (2003) Adaptive Name Matching in Information Integration. *IEEE Intell Syst* 18:16–23. <https://doi.org/10.1109/MIS.2003.1234765>
94. Huang SJ, Gao W, Zhou ZH (2018) Fast Multi-Instance Multi-Label Learning. *IEEE Trans Pattern Anal Mach Intell* PP:1. <https://doi.org/10.1109/TPAMI.2018.2861732>
95. Zhang ML, Zhou ZH (2007) ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognit* 40:2038–2048. <https://doi.org/10.1016/j.patcog.2006.12.019>
96. Yan J, Zhang B, Liu N, et al (2006) Effective and efficient dimensionality reduction for large-scale and streaming data preprocessing. *IEEE Trans Knowl Data Eng* 18:320–332. <https://doi.org/10.1109/TKDE.2006.45>
97. Xu X, Liang T, Zhu J, et al (2019) Review of classical dimensionality reduction and sample selection methods for large-scale data processing. *Neurocomputing* 328:5–15. <https://doi.org/10.1016/j.neucom.2018.02.100>
98. Pölsterl S, Conjeti S, Navab N, Katouzian A (2016) Survival analysis for high-dimensional, heterogeneous medical data: Exploring feature extraction as an alternative to feature selection. *Artif Intell Med* 72:1–11.
<https://doi.org/10.1016/j.artmed.2016.07.004>
99. Mladenić D, Grobelnik M (2003) Feature selection on hierarchy of web documents. *Decis Support Syst* 35:45–87. [https://doi.org/10.1016/S0167-9236\(02\)00097-0](https://doi.org/10.1016/S0167-9236(02)00097-0)
100. Haury AC, Gestraud P, Vert JP (2011) The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. *PLoS One* 6:1–14.
<https://doi.org/10.1371/journal.pone.0028210>
101. Yang J, Liu Y, Zhu X, et al (2012) A new feature selection based on comprehensive measurement both in inter-category and intra-category for text categorization. *Inf*

- Process Manag 48:741–754. <https://doi.org/10.1016/j.ipm.2011.12.005>
102. Inza I, Larrañaga P, Blanco R, Cerrolaza AJ (2004) Filter versus wrapper gene selection approaches in DNA microarray domains. *Artif Intell Med* 31:91–103. <https://doi.org/10.1016/j.artmed.2004.01.007>
 103. Uysal AK, Gunal S (2012) A novel probabilistic feature selection method for text classification. *Knowledge-Based Syst* 36:226–235. <https://doi.org/10.1016/j.knosys.2012.06.005>
 104. Roobaert D, Karakoulas G, Chawla N V. (2008) Information Gain, Correlation and Support Vector Machines. *Featur Extr* 470:463–470. https://doi.org/10.1007/978-3-540-35488-8_23
 105. Lei S (2012) A feature selection method based on information gain and genetic algorithm. *Proc - 2012 Int Conf Comput Sci Electron Eng ICCSEE 2012* 2:355–358. <https://doi.org/10.1109/ICCSEE.2012.97>
 106. Jin X, Xu A, Bie R, Guo P (2006) Machine Learning Techniques and Chi-Square Feature Selection for Cancer Classification Using SAGE Gene Expression Profiles. *Data Min Biomed Appl* 106–115
 107. Chen YT, Chen MC (2011) Using chi-square statistics to measure similarities for text categorization. *Expert Syst Appl* 38:3085–3090. <https://doi.org/10.1016/j.eswa.2010.08.100>
 108. Gu Q, Li Z, Han J (2010) Generalized Fisher Score for Feature Selection A Brief Review of Fisher Score. *Ratio*
 109. J. Weston and S. Mukherjee and O. Chapelle and M. Pontil and T. Poggio and V. Vapnik (2000) Feature Selection for SVMs. *Adv Neural Inf Process Syst* 13 668--674
 110. Youn E, Koenig L, Jeong MK, Baek SH (2010) Support vector-based feature selection using Fisher's linear discriminant and Support Vector Machine. *Expert Syst Appl* 37:6148–6156. <https://doi.org/10.1016/j.eswa.2010.02.113>
 111. Yi Wang, Xiao-Jing Wang (2005) A new approach to feature selection in text classification. 3814-3819 Vol. 6. <https://doi.org/10.1109/icmlc.2005.1527604>
 112. Genkin A, Lewis DD, Madigan D (2007) Large-scale bayesian logistic regression for

- text categorization. *Technometrics* 49:291–304.
<https://doi.org/10.1198/004017007000000245>
113. Max-dependency C (2005) Feature Selection Based on Mutual Information : 27:1226–1238
 114. Labani M, Moradi P, Ahmadizar F, Jalili M (2018) Engineering Applications of Artificial Intelligence A novel multivariate filter method for feature selection in text classification problems. *Eng Appl Artif Intell* 70:25–37.
<https://doi.org/10.1016/j.engappai.2017.12.014>
 115. Norman KA, Polyn SM, Detre GJ, Haxby J V (2006) Beyond mind-reading: multi-voxel pattern analysis of fMRI data. *Trends Cogn Sci* 10:424–30.
<https://doi.org/10.1016/j.tics.2006.07.005>
 116. Wang L, Lei Y, Zeng Y, et al (2013) Principal Feature Analysis : A Novel Voxel Selection method for fMRI Data. *Comput Math Methods Med* 2013:3–7
 117. Granitto PM, Furlanello C, Biasioli F, Gasperi F (2006) Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products. *Chemom Intell Lab Syst* 83:83–90. <https://doi.org/10.1016/j.chemolab.2006.01.007>
 118. Mao KZ (2002) Fast orthogonal forward selection algorithm for feature subset selection. *IEEE Trans Neural Networks* 13:1218–1224.
<https://doi.org/10.1109/TNN.2002.1031954>
 119. Chen H, Jiang W, Li C, Li R (2013) A heuristic feature selection approach for text categorization by using chaos optimization and genetic algorithm. *Math Probl Eng* 2013:.. <https://doi.org/10.1155/2013/524017>
 120. Leardi R, Boggia R, Terrile M (1992) GENETIC ALGORITHMS AS A STRATEGY FOR FEATURE SELECTION. 6:267–281
 121. Guyon I, Bitter H-M, Ahmed Z, et al (2003) Multivariate Non-Linear Feature Selection with Kernel Multiplicative Updates and Gram-Schmidt Relief. *BISC FLINT-CIBI 2003 Work* 1–11
 122. Urbanowicz RJ, Meeker M, Cava W La, et al (2018) Relief-Based Feature Selection : Introduction and Review. *J Biomed Inform.* <https://doi.org/10.1016/j.jbi.2018.07.014>

123. Urbanowicz RJ, Olson RS, Schmitt P, et al (2018) Benchmarking Relief-Based Feature Selection Methods for Bioinformatics Data Mining. *J Biomed Inform.*
<https://doi.org/10.1016/j.jbi.2018.07.015>
124. Mimouni N, Yeung TY (2018) Comparing Performance of Text Pre-processing Methods for Predicting a Binary Position by LASSO Experiment with Textual Data of European Union Public Consultation. 18–21
125. Marafino BJ, Boscardin WJ, Dudley RA (2015) Efficient and sparse feature selection for biomedical text classification via the elastic net : Application to ICU risk stratification from nursing notes. *J Biomed Inform* 54:114–120.
<https://doi.org/10.1016/j.jbi.2015.02.003>
126. Taylor P, Hoerl AE, Kennard RW (2012) Technometrics Ridge Regression : Biased Estimation for Nonorthogonal Problems Ridge Regression : Biased Estimation Nonorthogonal Problems. 37–41
127. Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. 301–320
128. Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet Allocation. 3:993–1022
129. Isa D, Lee LH, Kallimani VP, Rajkumar R (2008) Text document preprocessing with the bayes formula for classification using the support vector machine. *IEEE Trans Knowl Data Eng* 20:1264–1272. <https://doi.org/10.1109/TKDE.2008.76>
130. Lee C, Lee GG (2006) Information gain and divergence-based feature selection for machine learning-based text categorization. *Inf Process Manag* 42:155–165.
<https://doi.org/10.1016/j.ipm.2004.08.006>
131. Trstenjak B, Mikac S, Donko D (2014) KNN with TF-IDF based framework for text categorization. *Procedia Eng* 69:1356–1364.
<https://doi.org/10.1016/j.proeng.2014.03.129>
132. Liu J, Jin T, Pan K, et al (2017) An improved KNN text classification algorithm based on Simhash. *Proc 2017 IEEE 16th Int Conf Cogn Informatics Cogn Comput ICCI*CC 2017* 92–95. <https://doi.org/10.1109/ICCI-CC.2017.8109735>
133. Khreisat L (2009) A machine learning approach for Arabic text classification using N-

- gram frequency statistics. *J Informetr* 3:72–77.
<https://doi.org/10.1016/j.joi.2008.11.005>
134. Apt C, Weiss S (1997) Data mining with decision trees and decision rules
 135. Mahender CN (2012) TEXT CLASSIFICATION AND CLASSIFIERS : 3:85–99
 136. Yang Y (1997) An Evaluation of Statistical Approaches To Text Categorization. *Inf Retr Boston* 1:69–90
 137. Wiener E, Pedersen J, Weigend A (1995) A Neural Network Approach to Topic Spotting. *Annu Symp Doc Anal Inf Retr* 317:332
 138. Johnson R, Zhang T (2014) Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. 103–112
 139. Zhang Y, Zhang Z, Miao D, Wang J (2019) Three-way enhanced convolutional neural networks for sentence-level sentiment classification. *Inf Sci (Ny)* 477:55–64.
<https://doi.org/10.1016/j.ins.2018.10.030>
 140. Kalchbrenner N, Grefenstette E, Blunsom P (2014) A convolutional neural network for modelling sentences. 52nd Annu Meet Assoc Comput Linguist ACL 2014 - Proc Conf 1:655–665. <https://doi.org/10.3115/v1/p14-1062>
 141. Elman JL (1990) Finding structure in time. *Cogn Sci* 14:179–211.
[https://doi.org/10.1016/0364-0213\(90\)90002-E](https://doi.org/10.1016/0364-0213(90)90002-E)
 142. Makarencov V, Guy I, Hazon N, et al (2019) Implicit dimension identification in user-generated text with LSTM networks. *Inf Process Manag.*
<https://doi.org/10.1016/j.ipm.2019.02.007>
 143. Sarmiento L, Carvalho P, Silva MJ, de Oliveira E (2009) Automatic creation of a reference corpus for political opinion mining in user-generated content. 29.
<https://doi.org/10.1145/1651461.1651468>
 144. Ruchansky N, Seo S, Liu Y (2017) CSI: A Hybrid Deep Model for Fake News Detection. <https://doi.org/10.1145/3132847.3132877>
 145. Chung J Gated Recurrent Neural Networks on Sequence Modeling arXiv : 1412 . 3555v1 [cs . NE] 11 Dec 2014. 1–9

146. Tang D, Qin B, Liu T (2015) Document modeling with gated recurrent neural network for sentiment classification. Conf Proc - EMNLP 2015 Conf Empir Methods Nat Lang Process 1422–1432. <https://doi.org/10.18653/v1/d15-1167>
147. Kuchaiev O, Ginsburg B (2019) Factorization tricks for LSTM networks. 5th Int Conf Learn Represent ICLR 2017 - Work Track Proc 1–6
148. Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarczyk, Andy Davis, Quoc Le, Geoffrey Hinton and JD (2017) Outrageously large neural networks: The sparsely-gated mixture-of-experts layer
149. Sarthak Jain BCW (2019) Attention is not Explanation. arXiv
150. Serrano S, Smith NA (2020) Is attention interpretable? ACL 2019 - 57th Annu Meet Assoc Comput Linguist Proc Conf 2931–2951. <https://doi.org/10.18653/v1/p19-1282>
151. Vashishth S, Upadhyay S, Tomar GS, Faruqui M (2019) Attention Interpretability Across NLP Tasks. 1–10
152. T. Munkhdalai and H. Yu (2017) “Neural semantic encoders.” Proc Conf Assoc Comput Linguist 1:397
153. Commissariat T (2020) Ask me anything: Dynamic Memory Networks for Natural Language Processing. Phys World 33:53–58. <https://doi.org/10.1088/2058-7058/33/3/35>
154. Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. Adv Neural Inf Process Syst 2017-Decem:5999–6009
155. Sun C, Qiu X, Xu Y, Huang X (2019) How to Fine-Tune BERT for Text Classification? Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 11856 LNAI:194–206. https://doi.org/10.1007/978-3-030-32381-3_16
156. Lan Z, Chen M, Goodman S, et al (2019) ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. 1–17
157. Joshi M, Chen D, Liu Y, et al SpanBERT: Improving Pre-training by Representing and Predicting Spans
158. Yang Y (1999) An Evaluation of Statistical Approaches to Text Categorization. Inf

- Retr Boston 1:69–90. <https://doi.org/10.1023/A:1009982220290>
159. Kowsari K, Meimandi KJ, Heidarysafa M, et al (2019) Text classification algorithms: A survey. *Inf* 10:1–68. <https://doi.org/10.3390/info10040150>
 160. Alami N, Meknassi M, En-nahnahi N (2019) Enhancing unsupervised neural networks based text summarization with word embedding and ensemble learning. *Expert Syst Appl* 123:195–211. <https://doi.org/10.1016/j.eswa.2019.01.037>
 161. Ittoo A, Nguyen LM, Van Den Bosch A (2016) Text analytics in industry: Challenges, desiderata and trends. *Comput Ind* 78:96–107. <https://doi.org/10.1016/j.compind.2015.12.001>
 162. Wiegrefe S, Pinter Y (2020) Attention is not not explanation. *EMNLP-IJCNLP 2019 - 2019 Conf Empir Methods Nat Lang Process 9th Int Jt Conf Nat Lang Process Proc Conf* 11–20. <https://doi.org/10.18653/v1/d19-1002>
 163. Atkinson-Abutridy J, Mellish C, Aitken S (2004) Combining Information Extraction with Genetic Algorithms for Text Mining. *IEEE Intell Syst* 19:22–30. <https://doi.org/10.1109/MIS.2004.4>
 164. Kolya AK, Ekbal A, Bandyopadhyay S (2012) A Hybrid Approach for Event Extraction. *Polibits* 46:55–59. <https://doi.org/10.17562/pb-46-6>
 165. Liu Y, Loh HT, Sun A (2009) Imbalanced text classification: A term weighting approach. *Expert Syst Appl* 36:690–701. <https://doi.org/10.1016/j.eswa.2007.10.042>
 166. Sun A, Lim EP, Liu Y (2009) On strategies for imbalanced text classification using SVM: A comparative study. *Decis Support Syst* 48:191–201. <https://doi.org/10.1016/j.dss.2009.07.011>
 167. García S, Herrera F (2009) Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. *Evol Comput* 17:275–306. <https://doi.org/10.1162/evco.2009.17.3.275>
 168. Tsai CF, Lin WC, Hu YH, Yao GT (2019) Under-sampling class imbalanced datasets by combining clustering analysis and instance selection. *Inf Sci (Ny)* 477:47–54. <https://doi.org/10.1016/j.ins.2018.10.029>
 169. Liang G, Zhang C (2012) A comparative study of sampling methods and algorithms

- for imbalanced time series classification. *Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics)* 7691 LNAI:637–648.
https://doi.org/10.1007/978-3-642-35101-3_54
170. Zhang H, Li M (2014) RWO-Sampling: A random walk over-sampling approach to imbalanced data classification. *Inf Fusion* 20:99–116.
<https://doi.org/10.1016/j.inffus.2013.12.003>
 171. Zhu T, Lin Y, Liu Y (2017) Synthetic minority oversampling technique for multiclass imbalance problems. *Pattern Recognit* 72:327–340.
<https://doi.org/10.1016/j.patcog.2017.07.024>
 172. Sain H, Purnami SW (2015) Combine Sampling Support Vector Machine for Imbalanced Data Classification. *Procedia Comput Sci* 72:59–66.
<https://doi.org/10.1016/j.procs.2015.12.105>
 173. Raza M, Hussain FK, Hussain OK, et al (2019) A comparative analysis of machine learning models for quality pillar assessment of SaaS services by multi-class text classification of users’ reviews. *Futur Gener Comput Syst* 101:341–371.
<https://doi.org/10.1016/j.future.2019.06.022>
 174. Jeatrakul P, Wong KW (2012) Enhancing classification performance of multi-class imbalanced data using the OAA-DB algorithm. *Proc Int Jt Conf Neural Networks* 10–15. <https://doi.org/10.1109/IJCNN.2012.6252450>
 175. Zhang Z, Krawczyk B, Garcia S, et al (2016) Empowering one-vs-one decomposition with ensemble learning for multi-class imbalanced data. *Knowledge-Based Syst* 106:251–263. <https://doi.org/10.1016/j.knosys.2016.05.048>
 176. Zhang C, Bi J, Xu S, et al (2019) Multi-Imbalance: An open-source software for multi-class imbalance learning. *Knowledge-Based Syst* 174:137–143.
<https://doi.org/10.1016/j.knosys.2019.03.001>
 177. Schapire RE, Singer Y, Singhal A (1998) Boosting and Rocchio applied to text filtering. *SIGIR Forum (ACM Spec Interes Gr Inf Retrieval)* 215–223.
<https://doi.org/10.1145/290941.290996>
 178. Mathew K, Issac B (2011) Intelligent spam classification for mobile text message. *Proc 2011 Int Conf Comput Sci Netw Technol ICCSNT 2011* 1:101–105.

<https://doi.org/10.1109/ICCSNT.2011.6181918>

179. Stein RA, Jaques PA, Valiati JF (2019) An analysis of hierarchical text classification using word embeddings. *Inf Sci (Ny)* 471:216–232.
<https://doi.org/10.1016/j.ins.2018.09.001>
180. Oza NC (2005) Online bagging and boosting. *Conf Proc - IEEE Int Conf Syst Man Cybern* 3:2340–2345. <https://doi.org/10.1109/icsmc.2005.1571498>
181. Özçift A (2011) Random forests ensemble classifier trained with data resampling strategy to improve cardiac arrhythmia diagnosis. *Comput Biol Med* 41:265–271.
<https://doi.org/10.1016/j.combiomed.2011.03.001>
182. Verma B, Rahman A (2012) Cluster-oriented ensemble classifier: Impact of multicluster characterization on ensemble classifier learning. *IEEE Trans Knowl Data Eng* 24:605–618. <https://doi.org/10.1109/TKDE.2011.28>
183. Dal Pozzolo A, Johnson R, Caelen O, et al (2014) Using HDDT to avoid instances propagation in unbalanced and evolving data streams. *Proc Int Jt Conf Neural Networks* 588–594. <https://doi.org/10.1109/IJCNN.2014.6889638>
184. Liu X, Li Q, Zhou Z (2013) Learning Imbalanced Multi-class Data with Optimal Dichotomy Weights. 478–487. <https://doi.org/10.1109/ICDM.2013.51>
185. Krawczyk B (2017) PT US CR. *Knowledge-Based Syst.*
<https://doi.org/10.1016/j.knosys.2017.09.032>
186. Alcalá-Fdez J, Sánchez L, García S, et al (2009) KEEL: A software tool to assess evolutionary algorithms for data mining problems. *Soft Comput* 13:307–318.
<https://doi.org/10.1007/s00500-008-0323-y>
187. Ghanem AS, Venkatesh S, West G (2010) Multi-class pattern classification in imbalanced data. *Proc - Int Conf Pattern Recognit* 2881–2884.
<https://doi.org/10.1109/ICPR.2010.706>
188. Lemaître G, Nogueira F, Aridas CK (2017) Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *J Mach Learn Res* 18:1–5
189. Tumala MM, Omotosho BS (2020) A Text Mining Analysis of Central Bank Monetary Policy Communication in Nigeria. *Cent Bank Niger J Appl Stat* 73–107.

<https://doi.org/10.33429/cjas.10219.3/6>

190. Verma I, Dey L, Meisheri H (2017) Detecting, quantifying and accessing impact of news events on indian stock indices. Proc - 2017 IEEE/WIC/ACM Int Conf Web Intell WI 2017 550–557. <https://doi.org/10.1145/3106426.3106482>
191. Kaya M, Fidan G, Toroslu IH (2012) Sentiment analysis of Turkish political news. Proc - 2012 IEEE/WIC/ACM Int Conf Web Intell WI 2012 174–180. <https://doi.org/10.1109/WI-IAT.2012.115>
192. Yu LC, Wu JL, Chang PC, Chu HS (2013) Using a contextual entropy model to expand emotion words and their intensity for the sentiment classification of stock market news. Knowledge-Based Syst 41:89–97. <https://doi.org/10.1016/j.knosys.2013.01.001>
193. Sang EFTK, De Meulder F (2003) Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. <https://doi.org/10.3115/1118853.1118877>
194. Carlini F, Cucinelli D, Previtali D, Soana MG (2020) Don't talk too bad! stock market reactions to bank corporate governance news. J Bank Financ 121:105962. <https://doi.org/10.1016/j.jbankfin.2020.105962>
195. Ahern KR, Sosyura D (2014) Who Writes the News ? Corporate Press Releases during Merger Negotiations. LXIX: <https://doi.org/10.1111/jofi.12109>
196. Besley T, Prat A (2001) Handcuffs for the Grabbing Hand ? The Role of the Media in Political Accountability
197. Ruta D (2014) Automated trading with machine learning on big data. Proc - 2014 IEEE Int Congr Big Data, BigData Congr 2014 824–830. <https://doi.org/10.1109/BigData.Congress.2014.143>
198. Schumaker RP, Zhang Y, Huang CN, Chen H (2012) Evaluating sentiment in financial news articles. Decis Support Syst 53:458–464. <https://doi.org/10.1016/j.dss.2012.03.001>
199. Hagenau M, Liebmann M, Neumann D (2013) Automated news reading: Stock price prediction based on financial news using context-capturing features. Decis Support

- Syst 55:685–697. <https://doi.org/10.1016/j.dss.2013.02.006>
200. Nuij W, Milea V, Hogenboom F, et al (2014) An automated framework for incorporating news into stock trading strategies. *IEEE Trans Knowl Data Eng* 26:823–835. <https://doi.org/10.1109/TKDE.2013.133>
 201. El-Masry AM, Ghaly MF, Khalafallah MA, El-Fayed YA (2002) Deep Learning for Event-Driven Stock Prediction Xiao. *J Sci Ind Res (India)* 61:719–725
 202. IEEE Instrumentation and Measurement Society, IEEE Computational Intelligence Society, Institute of Electrical and Electronics Engineers (2017) CIVEMSA : 2017 IEEE International Conference on Computational Intelligence & Virtual Environments for Measurement Systems and Applications : 2017 conference proceedings : Annecy, France, June 26-28, 2017,
 203. M. G. Sousa, K. Sakiyama, L. d. S. Rodrigues, P. H. Moraes, E. R. Fernandes and E. T. Matsubara, “BERT for Stock Market Sentiment Analysis,” 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), 2019, pp. 1597-1601, doi: 1
 204. Ikonomakis M, Kotsiantis S, Tampakas V (2005) Text classification using machine learning techniques. *WSEAS Trans Comput* 4:966–974. <https://doi.org/10.11499/sicejl1962.38.456>
 205. Krawczyk B (2016) Learning from imbalanced data: open challenges and future directions. *Prog Artif Intell* 5:221–232. <https://doi.org/10.1007/s13748-016-0094-0>
 206. Ahmed S, Mahbub A, Rayhan F, et al (2018) Hybrid Methods for Class Imbalance Learning Employing Bagging with Sampling Techniques. *2nd Int Conf Comput Syst Inf Technol Sustain Solut CSITSS 2017* 1–5. <https://doi.org/10.1109/CSITSS.2017.8447799>
 207. Wasantha Premarathne, KDU SL (2017) Snoran Sampling (Hybrid Sampling). 0–2. <https://doi.org/10.13140/RG.2.2.15181.64488>
 208. Dogra V, Verma S (2019) Challenges and Opportunities in Labeling for Text Classification. *Think India* 22:4390–4400
 209. Lyon RJ, Brooke JM, Knowles JD, Stappers BW (2013) Hellinger Distance Trees for

Imbalanced Streams

210. Verma MK, Xaxa DK, Verma S (2017) DBCS : Density based Cluster Sampling for Solving Imbalanced Classification Problem. 156–161
211. Yang K, Yu Z, Member S, et al (2019) Hybrid Classifier Ensemble for Imbalanced Data. IEEE Trans Neural Networks Learn Syst PP:1–14.
<https://doi.org/10.1109/TNNLS.2019.2920246>
212. Strubell E, Ganesh A, McCallum A (2020) Energy and policy considerations for deep learning in NLP. ACL 2019 - 57th Annu Meet Assoc Comput Linguist Proc Conf 3645–3650. <https://doi.org/10.18653/v1/p19-1355>
213. Bhardwaj A, Narayan Y, Vanraj, et al (2015) Sentiment Analysis for Indian Stock Market Prediction Using Sensex and Nifty. Procedia Comput Sci 70:85–91.
<https://doi.org/10.1016/j.procs.2015.10.043>
214. Shah D, Isah H, Zulkernine F (2019) Predicting the Effects of News Sentiments on the Stock Market. Proc - 2018 IEEE Int Conf Big Data, Big Data 2018 4705–4708.
<https://doi.org/10.1109/BigData.2018.8621884>
215. Bekaert G, Ehrmann M, Fratzscher M, Mehl A (2014) The Global Crisis and Equity Market Contagion. J Finance 69:2597–2649. <https://doi.org/10.1111/jofi.12203>
216. Winter JCF (2013) Using the student's t-test with extremely small sample sizes. Pract Assessment, Res Eval 18:1–12
217. Tan HB, Hooy CW (2004) Bank merger and bank stock volatility: A post - announcement analysis. Manag Financ 30:29–47.
<https://doi.org/10.1108/03074350410769010>
218. Tourani Rad A, Van Beek L (1999) Market valuation of European bank mergers. Eur Manag J 17:532–540. [https://doi.org/10.1016/S0263-2373\(99\)00042-0](https://doi.org/10.1016/S0263-2373(99)00042-0)

Research Publications

1. Dogra, Varun, Aman Singh, Sahil Verma, Abdullah Alharbi, and Wael Alosaimi. "Event Study: Advanced Machine Learning and Statistical Technique for Analyzing Sustainability in Banking Stocks" *Mathematics* 9, no. 24 (2021): 3319.
2. Dogra, Varun. "Banking news-events representation and classification with a novel hybrid model using DistilBERT and rule-based features." *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 12.10 (2021): 3039-3054.
3. Dogra, Varun, et al. "Understanding of Data Preprocessing for Dimensionality Reduction Using Feature Selection Techniques in Text Classification." *Intelligent Computing and Innovation on Data Science*. Springer, Singapore, (2021). 455-464.
4. Dogra, Varun, et al. "Analyzing DistilBERT for Sentiment Classification of Banking Financial News." *Intelligent Computing and Innovation on Data Science*. Springer, Singapore, (2021). 501-510.