FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

IMPROVING SENTIMENT ANALYSIS BASED ON DEEP LEARNING BY USING FEATURE SELECTION

Mohammed Hussein ABDALLA

Master's Thesis

DEPARTMENT OF SOFTWARE ENGINEERING

Program of Software Engineering

JUNE 2021

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

Department of Software Engineering
Software Engineering

Master's Thesis

IMPROVING SENTIMENT ANALYSIS BASED ON DEEP LEARNING BY USING FEATURE SELECTION

Author
Mohammed Hussein ABDALLA

Supervisor
Associate Prof. Dr. FATIH ÖZYURT

JUNE 2021
ELAZIG

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

TÜRKİYE

Department of Software Engineering
Software Engineering

Master's Thesis

Title:	Improving Sentiment Analysis Based on Deep Learning by Using Feature Selection
Author:	Mohammed Hussein ABDALLA
Submission Date:	26 April 2021
Defense Date:	01 June 2021

THESIS APPROVAL

This thesis, which was prepared according to the thesis writing rules of the Graduate School of Natural and Applied Sciences, Fırat University, was evaluated by the committee members who have signed the following signatures and was unanimously approved after the defense exam made open to the academic audience.

Signature

Supervisor:	Associate Prof. Dr. FATIH ÖZYURT Firat University, Faculty of Science	Approved
Chair:	Prof. Dr. Engin AVCI Firat University, Faculty of Science	Approved
Member:	Associate Prof. Dr. Eser SERT Turgut Özal University, Faculty of Engineering and Natural	Approved

This thesis was approved by the Administrative Board of the Graduate School on

..... / ..... / 20 .....

Signature

Assoc. Dr. Kursat ESAT ALYAMAÇ
Director of the Graduate School

DECLARATION

I hereby declare that I wrote this Master's Thesis titled “Improving Sentiment Analysis Based on Deep Learning by Using Feature Selection” in consistent with the thesis writing guide of the Graduate School of Natural and Applied Sciences, Firat University. I also declare that all information in it is correct, that I acted according to scientific ethics in producing and presenting the findings, cited all the references I used, express all institutions or organizations or persons who supported the thesis financially. I have never used the data and information I provide here in order to get a degree in any way.

01 June 2021

Mohammed Hussein ABDALLA

PREFACE

First of all, Praise to Allah, Lord of the Worlds, for his grace and mercy throughout my work. Thank you for the guidance, strength, protection, and skills and for giving me a healthy life to be able to complete this Thesis. I would really like to explicit gratitude to my supervisor (Associate Prof. Dr. Fatih ÖZYURT) for his precious advice and encouragement during the research procedure.

Additionally, I would like to thank all the lecturers for their support, opportunities and facilities for carrying out this work. Also, I want to express appreciation to Firat University for giving me this extraordinary opportunity to study master degree.

Additionally, Special thanks to my parents and wife for their help and endless support during my life. Finally, thanks to all who supported me greatly or sharing with me their knowledge at some stage in my research project.

Mohammed Hussein ABDALLA
ELAZIG, 2021

TABLE OF CONTENTS

Page

ABSTRACT

Improving Sentiment Analysis Based on Deep Learning by Using Feature Selection

Mohammed Hussein ABDALLA

Master's Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences

Department of Software Engineering
Software Engineering
June 2021, Page: xii + 51

In recent years, sentiment analysis has gained a great deal of research due to the dramatic growth in online social media use. For learning-based methods, many technically advanced strategies have been used to improve the performance of the forms. The sentiment analysis system uses natural language processing techniques and a sentimental vocabulary network, and one of the applications of deep learning in NLP is sentiment analysis.

The most popular and successful type of RNN is the LSTM network. There is much research that uses the LSTM ability to analyze sentiment. However, large data volumes reduce the accuracy of LSTM network results in test data; in other words, over-fitting occurs. This problem occurs when there is a high correlation between independent variables. The model may not have high validity despite the high value of the correlation coefficient between the independent and dependent variables. In other words, although the model looks good, it does not have significant independent variables. Combining the LSTM network with feature selection methods can increase sentiment analysis accuracy to select effective features and solve them. In this study, we used the three benchmark datasets, namely (YELP, US Airline, and IMDB), then proposed a deep learning model (LSTM, Bi-LSTM, and GRU) to improve classification accuracy through using the feature selection method, namely, filter-based feature selection method Chi-square and Principle component analysis method PCA to select an optimal subset of features, and the performance of each measured and compared in terms of accuracy, precision, recall, and F1 score.

After comparing the (LSTM, Bi-LSTM, and GRU) models with (Chi-2 and PCA) selected features, and also with the original feature set, the results show that feature selection methods significantly increase classification accuracy in all cases. . In the Yelp dataset, the maximum attained an accuracy of Bi-LSTM is 100% using chi-square. In the US Airline dataset, the maximum achieved accuracy of GRU-LSTM is 97.9% using chi-square. In the IMDB dataset, the maximum achieved accuracy of Bi-LSTM is 99.9% using chi-square.

Keywords: Sentiment Analysis, Deep Learning, LSTM network, Feature Selection.

ÖZET

Özellik Seçimini Kullanarak Derin Öğrenmeye Dayalı Duygu Analizi İyileştirme

Mohammed Hussein ABDALLA

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Yazılım Mühendisliği Anabilim Dalı

Haziran 2021, Sayfa: xii + 51

Son yıllarda, çevrimiçi sosyal medya kullanımındaki çarpıcı artış nedeniyle duyarlılık analizi çok fazla araştırma kazandı. Öğrenmeye dayalı yöntemler için, formların performansını iyileştirmek için birçok teknik olarak gelişmiş strateji kullanılmıştır. Duygu analizi sistemi, doğal dil işleme tekniklerini ve duygusal bir kelime ağını kullanır ve NLP'deki derin öğrenmenin uygulamalarından biri duygu analizidir.

En popüler ve başarılı RNN türü LSTM ağıdır. Duyguları analiz etmek için LSTM yeteneğini kullanan çok fazla araştırma var. Ancak, büyük veri hacimleri, test verilerinde LSTM ağ sonuçlarının doğruluğunu azaltır; başka bir deyişle, aşırı uydurma meydana gelir. Bu sorun, bağımsız değişkenler arasında yüksek bir korelasyon olduğunda ortaya çıkar. Bağımsız ve bağımlı değişkenler arasındaki korelasyon katsayısının yüksek değerine rağmen model yüksek geçerliliğe sahip olmayabilir. Diğer bir deyişle, model iyi görünmesine rağmen, önemli bağımsız değişkenlere sahip değildir. LSTM ağını özellik seçim yöntemleriyle birleştirmek, etkili özellikleri seçmek ve çözmek için duyarlılık analizi doğruluğunu artırabilir. Bu çalışmada, üç karşılaştırma veri setini (YELP, US Airline ve IMDB) kullandık, ardından özellik seçme yöntemini kullanarak sınıflandırma doğruluğunu iyileştirmek için bir derin öğrenme modeli (LSTM, Bi-LSTM ve GRU) önerdik. , filtre tabanlı özellik seçim yöntemi Ki-kare ve Prensip bileşen analiz yöntemi PCA, özelliklerin optimum bir alt kümesini seçmek ve her birinin performansı doğruluk, kesinlik, geri çağırma ve F1 skoru açısından ölçülür ve karşılaştırılır.

(LSTM, Bi-LSTM ve GRU) modellerini (Chi-2 ve PCA) seçilen özelliklerle ve ayrıca orijinal özellik setiyle karşılaştırdıktan sonra, sonuçlar, özellik seçim yöntemlerinin tüm durumlarda sınıflandırma doğruluğunu önemli ölçüde artırdığını göstermektedir. . Yelp veri setinde Bi-LSTM'nin elde edilen maksimum doğruluğu, ki-kare kullanılarak% 100'dür. ABD Havayolu veri kümesinde, GRU-LSTM'nin elde edilen maksimum doğruluğu ki-kare kullanılarak% 97,9'dur. IMDB veri kümesinde, Bi-LSTM'nin elde edilen maksimum doğruluğu ki-kare kullanılarak% 99,9'dur

Anahtar Kelimeler: Duygu Analizi, Derin Öğrenme, LSTM ağı, Özellik Seçimi.

LIST OF FIGURES

Page

LIST OF TABLES

Page

SYMBOLS AND ABBREVIATIONS

Abbreviations

RNN : Recursive Neural Network

LSTM : Long Short-Term Memory

Bi-LSTM : Bidirectional Long Short-Term Memory

GRU : Gated Recurrent Unit

PCA : Principal Component Analysis

IG : Information Gain

LSA : Latent Sematic Analysis

PSO : Part of Speech Tagging

SVM : Support Vector Machine

PSO : Particle Swarm Optimization

KNN : K-Nearest Neighbor

AUC : Area-Under-Curve

API : Application Programming Interface

GA : Genetic Algorithm

ML : Machine Learning

SWN : SentiWordNet

Chi-2 : Chi Square

NB : Naïve Bayes

GNB : Gaussian Naïve Bayes

Doc2Vec : Document to Vector

DM : Data Mining

DBoW : Distributed Bag of Word

Word2Vec: Word to Vector

LR : Logistic Regression

TF-IDF : Term Frequency – Inverse Document Frequency

BoW : Bag of Word

ELM : Extreme Learning Machine

NLTK : Natural Language Toolkit

OR : Odds Ratio

RFE : Recursive Feature Elimination

BCA : Binary Coordinate Ascent

SA : Sentiment Analysis

OM : Opinion Mining

NLP : Natural Language Processing

VADER : Valence Aware Dictionary

TF : Term Frequency

CSV : Comma Separated Value

TP : True Positive

TF : True Negative

FP : False Positive

FN : False Negative

1. INTRODUCTION

An important part of the data that is created today is unstructured. Unstructured data involves social media comments, browsing history and customer feedback. To process and analyze such data, text classification is one way to accomplish this task.

Social media has provided a new device for making and sharing ideas with others. Receiving public opinion about social events, marketing activities, and product priorities attracted the scientific community and marketing companies [1]. Today, if one wants to buy a product, he is no longer limited to family and friend opinions because users' comments and discussions about different products are available on websites. However, finding and evaluating opinions is not an easy task due to the expansion and variety of opinions. Each site usually has a large volume of text comments that are not easy to decrypt and summarize such data, so automated sentiment analysis systems are required [2]. Now, the acceptance of the generated content development by users on websites and social networks such as Twitter has led to an increase in the power of social media to express opinions about services, products and events. Today, the number of people using these networks to make decisions is increasing [3]. Using sentiment analysis systems, we can find out what people think about different topics and what opinions they have, and we can find out the reasons for the failure or success of different topics in society from the users' point of view by analyzing these ideas. Figure 1.1, shows the Sentiment Analysis System.

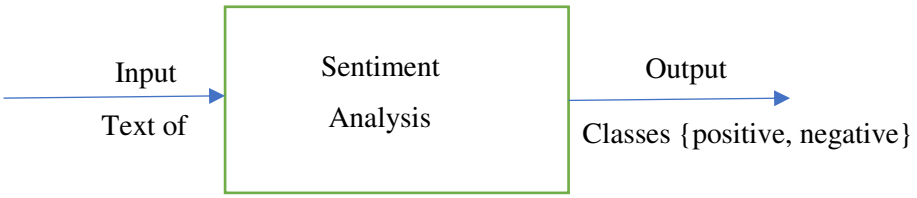


Figure 1.1. Sentiment Analysis System

Deep learning is a specific subfield of machine learning in artificial intelligence (AI) and composed of algorithms that permit software to train itself to perform tasks by exposing multilayered neural networks to vast amounts of data [4].

Analysis of sentiment at the document level is a basic task in sentiment classification. Recently, neural network approaches have been very successful in sentiment analysis. Recursive Neural Network (RNN) is one of the most common architectures among these methods, due to the ability to manage variable-length texts [5]. The recursive neural network needs information from the beginning of the sentence to be able to predict correctly. These types of dependencies are called long-term dependencies; because there is a relatively significant distance between information and

relevant features, and this information is used to make a prediction. Unfortunately, in practice, the greater the distance, the more difficult RNN neural networks become to learn these dependencies; because they deal with the problem of vanishing gradient or exploding gradient [6]. Efforts have been made to deal with the process of modeling long-texts. Recently, LSTM has become more popular in the field of sentiment analysis. The Short-Term Memory Network (LSTM) was proposed by Hockeriter and Schmidber in 1997 [7], which is a kind of recursive neural network that reduces the problem of vanishing/exploding gradient.

Unlike a traditional recursive neural network in which the content is re-written at each time step, in an LSTM recursive neural network the network can decide on the preservation of current memory through the introduced gates. Intuitively, if the LSTM unit detects an important feature in the input sequence in the initial steps, it can easily transmit this information over a long distance, thus it can receive and retain such potential long-term dependencies [8].

In recent years, LSTM has been used more in the field of sentiment analysis and has undergone many changes. One of its common type is Bi-LSTM, which consists of two LSTMs that use additional previous information, thereby increasing memory capability. The basic idea of Bi-LSTM is to have new and old LSTM training sequences, both connected to the same output layer. This structure provides complete content information from the past and future for each point of the input layer [7]. Another type is the GRU proposed by Chou et al., in which the forget gate and the input gate are merged and convert to a single update gate. The unit state and hidden state are also integrated, so the GRU model is simpler than the standard LSTM model [8]. Although good results have been obtained by the use of LSTM models, many cases in sources and texts show that with the increase in textual data features, training data is very accurate in network training, but in evaluating the model with testing data, the accuracy of the model is reduced and it faced with the problem of over-fitting in sentiment analysis and long-texts processing [9].

In addition, there is the problem of unbalanced datasets in the sentiment analysis of the long texts. Unbalanced data is generally referred to as a set of data in which the number of samples representing a class is less than other samples in different classes. This becomes problematic in time classification, in that a class, which is generally an absolute or minority class, is not represented in the datasets. In other words, the number of incorrect observations exceeds the correct observations in a class. This problem has been frequently observed in sentiment analysis of long texts, and the low quality of data, due to class imbalances or incorrect labeled items, may negatively affect classification performance [10]. Therefore, in this paper, a new method based on feature selection algorithms to reduce the dimensions of input structures (feature reduction) of LSTM models is presented, which aims to reduce computational time and increase the accuracy of predictions in testing data. In this regard, we examine the effect of combining correlation-based

feature selection methods such as chi-square test and principal component analysis (PCA) with LSTM, Bi-LSTM, and GRU-LSTM models in sentiment analysis.

1.1. Problem statement

Into the last couple of decades, the demand for collecting and analyzing social media data has increased. One great source for this data is Twitter. Words of sentiment texts can be utilized for the decision-making process in Sentiment Analysis, as they contain user's reviews about different events or topics. However, tweets are unstructured data and tend to be very large in volume when collected. Consisting of millions of records. While processing the collected raw data for data analytics purposes, they have to be converted into another form of representation, called features. With a large amount of data, the number of features tends to be very high due to containing irrelevant and redundant features, reaching hundreds of thousands in dimensionality. When applying machine learning algorithms on the high dimensional feature space, the performance of the algorithms will be adversely affected and degraded. The high dimensionality of feature space also leads to over fitting the learning algorithm, increasing learning time, and increasing memory requirements for data analytics.

Examining previous research, we conclude that these models are usually very accurate in training data. In the test data, the accuracy of the model has decreased. These models face over fitting in sentiment analysis and text processing due to the high number of features. Also, there is an unbalanced data set problem in sentiment analysis [10].

To alleviate the problem of high dimensional feature space, feature selection techniques are always recommended. Feature selection selects the most informative features from the high dimensional feature space using a calculation to remove irrelevant and redundant features. Consequently, solving the problems caused by high dimensionality.

1.2. Challenge of Sentiment Analysis

There are several problems associated with them, as expressed in the text. The most prevalent challenges are as follows:

1.2.1. Identification of Spam Analysis

Many people can try to write fake or unfair reviews on a product review site called a spam review, promote their products by giving unfair positive feelings, or degrade their competitors' products by providing false-negative emotions. Spam identification is critical for business communities. This will affect user experience because if a false opinion cheats the user, he/she will never want to use the system again [11-13].

1.2.2. Sarcastic Sentences

Sarcasm may be used to damage (or offend) or may be used for comedic effects. Reviews can include humorous and ironic expressions. In other words, the polarity of a sentence can be false positive. For example, "What a nice car, it stopped working on the second day," in such a situation, positive words can have a negative sense or meaning. Sarcastic or sarcastic phrases can be difficult to recognize, leading to an inaccurate interpretation of feelings [11-14].

1.2.3. The uncertainty of the Word Sense

Depending on the context, a word may have a different meaning. For example, the term "small size" has a positive polarity for mobile phones but a negative polarity for room sizes. Analyzing the correct meaning of the word is thus a costly task [11, 12].

1.2.4. Negation

Negations reverse the polarity of the sentence, and it should be dealt with accordingly. Otherwise, the product of sentiment analysis is affected [11-12].

1.2.5. The Slang Language and Abbreviations

Abbreviations and slang are short kinds of words to use in any language of communication. Usually, they are encountered when dealing with text processing and sentiment analysis. Consideration and handling of these is an additional overhead in sentiment analysis [13]. There are other questions, too. Some of them have to be considered during preparations, such as slang, small tweet, URLs, including tweets, emoji, and hashtags [13] while analyzing sentiment analysis text info.

1.3. Application of Sentiment Analysis

As one of the trendy fields of research, Sentiment analysis has many effective applications, some of them are stated below:

1.3.1. Industries or Business Organization

Sentiment analysis or opinion mining has an essential role in improving the business. Companies could take feedback from their customers about their products or level of services in a short time and hence improve their business. Amazon is one example of many companies using sentiment analysis for detecting customer satisfaction with their products [12].

1.3.2. Government Intelligence:

Sentiment analysis allows extracting people's opinion on a particular government policy to understand public reaction on implementing a specific plan. Additionally, politicians could use sentiment analysis to expose voters' views about different issues to work in a better way [15].

1.3.3. Education

The study of sentiment also plays a significant role in the field of education. Since the students or their parents are providing feedback at the end of the session/semester, the review of this feedback will improve the teaching-learning process. The faculty will come to know the desires and shortcomings of the students. This study will enable the faculty to improve its teaching methods. It can be beneficial for management if they summarize the faculty's overview input into an automated review tool and if the evaluation process can be carried out efficiently [11].

1.3.4. E-Commerce

Purchasing or selling online goods/services is called e-commerce. The study of sentiment is useful in this sense. If anyone wants to purchase something online, they can look at the feedback provided by other people who purchased the same product before purchasing one. If all the reviews are analyzed and summarized, the buyer can save a lot of time and effort. Sentiment analysis can help to summarize the various reviews. Similarly, if a consumer wants to sell anything, he should take a market survey before he quotes his purchase price. Opinions derived from reviews help us make a variety of choices, such as "what books to purchase," "what hotel to go to," "what movie to watch," "what doctor to go to," "what phone to buy," [12, 16, 17].

1.4. Sentiment Analysis Approaches

Sentiment analysis is a technique that utilizes computational models to categories the views that is stated in the text sentences to find out whether the views about a specific topic, service or product, the movie is neutral, positive or negative. Researchers divide Sentiment analysis techniques into three different approaches: machine learning, lexicon-based, and hybrid approaches.

Machine learning approaches apply famous machine learning algorithms. The lexicon-based approach relies on a sentiment lexicon, a collection of known and pre-compiled sentiment terms. It is divided into a dictionary-based approach and a corpus-based approach that uses statistics or semantic methods to find sentiment polarity. The hybrid approach combines both approaches and is very common with sentiment lexicons by playing a vital role in most methods [18]. Figure 1.2., illustrates different sentiment analysis techniques:

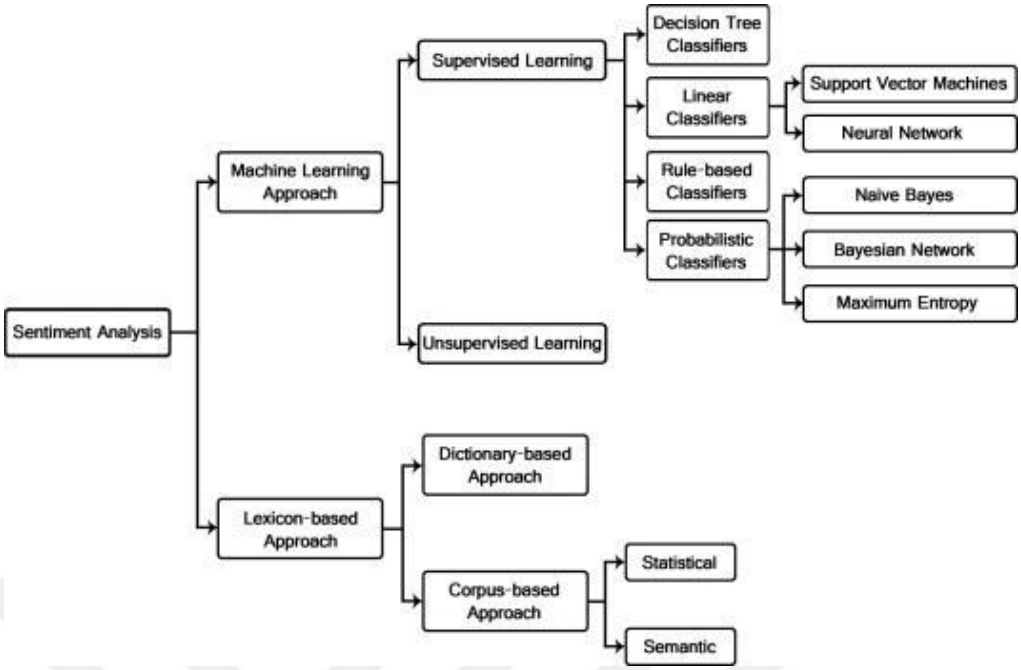


Figure 1.2. Different sentiment analysis techniques.

1.5. Machine Learning Approach

It is able to automatically manage a large amount of online data is more common than any other approach. The machine learning approach utilizes a classification method to classify the input text. There are two types of approach to machine learning:

1.6. Supervised Learning Based Approach

Supervised machine learning techniques are used for sentiment recognition, machine learning techniques. Classification usage is common. The data is partitioned into two classes. One is a numbered training set, and the other is a test set.

The model is trained on a training set and then validated using the test set of data. Critical algorithms in this group include SVM, Naïve Bayes, Logistic Regression, and Decision trees, and Neural Network. Since these techniques are highly dependent on the training data, their accuracy is vulnerable to being negatively impacted by an instance of incorrect training data or less training examples.

1.6.1. Unsupervised Learning Based Approach

Although supervised learning methods require a labelled training set of data to train the model, unsupervised learning techniques do not. When it is difficult to classify the input data, these methods are useful. The most well-known technique for unsupervised learning-based pattern

recognition is cluster analysis. Several clustering algorithms are used, like hierarchical clustering, k-means clustering [19, 20].

1.7. Lexicon-Based Approach

Most lexicon-based strategies are used to find the words in a text with a particular opinion about something. The results are obtained by adding the scores for subjective terms to account for specific text content, and the maximum score determines the overall polarity. The corpus-based approach is focused on corpora and dictionaries.

1.7.1. Dictionary Based Approach

In the Dictionary-based approach, opinion words are manually handled and a seed list is prepared. Then we search for dictionaries and thesaurus to find synonyms and antonyms of text. The newly found synonyms are added to the seed list. This process continues until no new words are found. The Dictionary-based approach's drawback is it difficult to find context or domain-oriented opinion words [21].

1.7.2. Corpus-Based Approach

A corpus-based approach aims to provide dictionaries linked to a particular subject field. This dictionary is created from a list of opinion seed terms, and the list is expanded using the corpus data. Since there is little context-oriented text on this domain, the Corpus has no problems with text information. With semantic approaches and statistical approaches, this can be achieved. LSA is one of many mathematical techniques, and WordNet is one of the semantic approaches. The corpus-based approach eliminates the context-specific and domain specific problem posed by the Dictionary-based approach.

1.8. Hybrid Approach

The hybrid approach combines machine learning and lexicon processing and increases overall performance. One methodology supported by most of the researches is to classify the polarity of the data using sentiment lexicon and then train the machine learning classifiers with the help of results from the sentiment lexicon [12, 22].

1.9. Related Work

Twitter, is one of the foremost prevalent online social networks and a platform that can be used for microblogging, sharing views and opinions using various forms of expression like

acronyms, abbreviations, emoticons. Grammar and spelling are not restricting users to communicate on the platform as such these features assist in categorizing sentiments in tweet bodies.

Sharma S. et al. analyzed Twitter data through real-time data extraction. Preprocessing and feature extraction was applied to the text data, Chi-square test and principal component analysis (PCA), and different machine learning classifications (SVM, Naïve Bayes, Random Forest) Logistic Regression) were conducted against performance indicators [23].

Rana, S., Singh, A. have proposed classifying movie reviews applying the Naïve Bayes and Linear SVM classifiers. They achieved that utilising the classifiers after omitting synthetic words gives a more accurate result. Their result reveals that SVM achieves better accuracy than the Naïve Bayes classifier. Both algorithms distinctly performed better for genre drama, reaching 87% with SVM and 80% with the Naïve Bayes algorithm [24].

Paul S. et al. attempted to classify text and sentiments in Twitter data by examining mean k-clustering and TF-IDF to weigh features depending on the simple Bayesian classifier and multiple Chi-squared feature selection algorithms. To show strength and weak points, this approach was compared to other variants of the Bayesian. After executing the experiments, the study concluded that the proposed method specifies more fantastic performance than too many others [25].

F Rustam et al. have a voting classifier (VC) to help sentiment analysis for such organizations on the US Airline dataset, Positive, negative and neutral tweets were classified based on their sentiments. A range of machine learning classifications has also been evaluated used as performance metrics for accuracy, accuracy, recall and F1 score. When they trained a model using, they achieved an accuracy of %78.9 and 79.1% with TF and TF-IDF feature extraction with the US Airline dataset, respectively [26].

Cheng, Y. et al. provide a multi-channel model that combines the CNN and the Bi-LSTM network with an attention mechanism (MC-AttCNN-AttBiGRU). The model will pay the model's capacity. The IMDB and Yelp 2015 data basket's experimental results indicate that the model proposed can extract more rich text features than other simple models and achieve 92.90% [27].

Wazery, Y. et al. implemented two main sentiment analysis methods support vector machine, naive Bayes, decision tree, and K-nearest neighbour. The second method is the deep neural network, an RNN with LSTM. Two approaches were also used by three Twitter datasets: IMDB, Amazon, and Airline. We also highlight a comparison between different algorithms, and the experimental results show the recurrent neural network using LSTM with the highest accuracy of 88%, 87%, and 93% [28].

Rane, A. et al. worked on a dataset from US Airlines and performed a multi-class analysis of sentiment. Seven classification strategy analyses were carried out: Gaussian Naïve Bayes, Random Forest, SVM, K-Nearest Neighbors, Decision Tree, Regression Logistics, and AdaBoost.

Classifiers were trained on 80% of the data and tested on 20% of the data. The test result is the positive, negative and neutral feeling of the tweet. Based on the results achieved, the accuracy of each classification approach was calculated. The classification techniques used include Adaboost, which combine several other classifications into a strong classification with a precision of %84.5 [29].

A. Kumar et al. used a particle swarm-based feature selection was performed to increase the efficiency of sentiment analysis on Twitter data. Tweets received from two standard Twitter datasets, SemEval 2016 & SemEval 2017, have been reviewed. The proposed feature selection using PSO is better than the basic group learning algorithm trained by conventional Tf-Idf. By implementing particle swarm optimization, an average of 8.5% improvement is achieved inaccuracy with a 33% reduction in the feature set [30].

Zainuddin, N. et al. according to the accuracy of sentiment analysis, a principal component analysis (PCA) based feature selection method is presented that can determine the most appropriate feature set for sentiment analysis. Feature selection helps to reduce redundant features and eliminate irrelevant features that affect classification accuracy. In this study, a feature selection method is presented to classify Twitter sentiment based on the principal component analysis (PCA). PCA is integrated with the Sentiwordnet dictionary-based method to carry out classification with the Support Vector Learning (SVM) framework. Experiments performed on the HCTS dataset and STS benchmark has 94.53% and 97.93%, respectively. Compared with other statistical feature selection methods, the proposed method of this study shows promising results in improving sentiment analysis performance [31].

S. Paudel et al. has implemented a new approach for selecting the appropriate number of features using chi-square, and features are selected using the default score threshold. It has been found that there is a relationship in learning-based techniques among the logarithm of the number of features selected, the output of the sentiment classification, and this relationship is independent of the learning-based process. New results in this study show that researchers can often select the right number of features in learning-based methods to achieve the best performance in sentiment classification. This will help researchers select the right features to improve the efficiency of learning-based algorithms [32].

Iqbal et al. introduced a hybrid approach to address scalability issues that arise as feature set sizes increase in sentiment analysis, using a Genetic Algorithm (GA)-based technique that reduced feature set size by up to 42% without compromising accuracy. They compared their proposed (GA)-based feature reduction technique to two other well-known techniques, Principal Component Analysis (PCA) and Latent Symantec Analysis (LSA). They confirmed that the GA-based methodology outperformed PCA and LSA by 15.4 per cent and 40.2 per cent, respectively. They have used three sentiment analysis methods: SentiWordNet, Machine Learning, and Machine

Learning with GA optimized feature selection. In all cases, the SWN method has a lower accuracy than the other two listed methods, with the best accuracy of 56%. Their built model, which integrates GA, reduces feature size by 36% - 43% and increases performance by 5% compared to the ML approach due to reduced feature size. They put their proposed model to the test with six different classifiers, namely (J48, NB, PART, SMO, IB-k, and JRIP). Using GA-based feature selection on the Twitter and feedback dataset, the NB classifier demonstrated the highest accuracy (around 80%). However, in the geopolitical dataset, IB-k outperformed other classifiers with an accuracy of 95% [33].

Zhai et al. used feature selection is performed using the Chi-2 feature selection method with single and double words along with the Support Vector Machine (SVM) and Naïve Bayesian as classifiers. Accuracy is increased as the number of features increases. 85.69% for 100 features, which eventually rises to 94.17% for 1,300 features, followed by 96.74 % accuracy for 2000 features. In the meantime, IG remains below Chi-2 for its exclusive features. Also, we can see that the selection of features combined with the selection of features can affect classification efficiency. The extraction of context-related multi-features with little redundancy decreases internal redundancy, resulting in improved classification performance [34].

Iqbal et al. used different features (NB, SVM, and Maximum Entropy) classifications to classify movie reviews in the IMDB dataset and tweets in the Stanford Twitter Sentiment140 data set for public opinion. The experiment includes four distinct sets of features, each of which is a combination of various single features, as follows: combined Single word features with Stopword filtered word features (set 1), Unigram with Bigram features (set 2), Bigram with Stopword filtered word features (set 3) and Most Informative Unigram with Most Informative Bigram features (set 4) (set4). Chi 2 was used as a supervised feature selection technique to improve efficiency by choosing the most informative features. Furthermore, Chi-2 aids in the reduction of the training data scale. Their results show that combining both unigram and bigram features and feeding them to the Maximum Entropy algorithm yields the best results in terms of (F1- the score, precision, and recall) when compared to two other algorithms, as well as by 2-5 per cent when compared to using a single feature and baseline model, which is the Senti- Wordnet process. Furthermore, the maximum achieved accuracy using Maximum Entropy in conjunction with uni-gram and bi-gram was 90% [35].

Ahmad et al. evaluated the overall efficacy of SVM on three pre-labelled datasets. Three ratios of training and test data have been used to classify each data set: 70:30, 50:50 and 30:70. The findings were compared based on the measures of precision, recall and f-score. In the first data collection, the ratio of 70:30 was better with 70.4% precision and 70.3% reminder, while 50:50 was better with 67.9% in the f-score. 70:30 surpassed the other two rates for the second dataset. With the 50% rate of the 3rd data collection, the best result for the SVM obtained the best results with

accuracy, retrieval, and f-score 80.4% of all three metrics. The ratio 70:30 with the first dataset performed better in the precision of 70.4% and recall of 70.3%, while 50:50 performed better in the f-score of 67.9%. For the second dataset, 70:30 outperformed the other two rates. In the third dataset, with the 50:50 rate, SVM performed better results in precision, recall, and f-score 80.4% for all three metrics, it was also the best ratio that the SVM could achieve the best result [36].

Mzhda et al. have been investigated sentiment analysis as one of the user problems of big data analytic. Furthermore, 70% of the 1,500,000 Tweets data set implemented by the University of Michigan were utilized for the training set, and the rest has been used for the test set. One Bag of words has been made from the training set and used by Multinomial Naïve Bayes algorithm and Forward Neural Network. Words containing 6000 positive and negative words have been used in this work. All of them are applied to the test set. Naïve Bayes has the highest F-Score among all of them, with F-Score of 76%. Neural Network has the second-highest accuracy, with F-Score of 74%, while the Existing Bag of words has the last accuracy with an F-Score of 68%. Apache Hive has been used to implement the algorithms inside Hortonworks Hadoop, which is a single cluster and portable Hadoop environment. For preprocessing purpose, python NLTK libraries have been used to enhance the algorithms' accuracy by two percentages. Twitter using OAuth API, and 50,000 tweets from five categories (iPhone 6, Obama, Kurd, Marriage and ISIS) [37].

1.10. Thesis Objectives

1. Create an automatic text analysis system to analyze people's opinions from the text about brands, products, services, and items to continually improve products' quality, provide better customer service, and receive feedback from target customers to precisely determine how to enhance the quality of their items or services.
2. Investigating the impact of using two different dimensionality reduction approaches (feature selection methods), Principle Component Analysis (PCA) and Chi-Square, to improve sentiment classification.
3. Applying three well-known supervised deep learning models. Namely, Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (Bi-LSTM) Moreover, Gated Recurrent Unit (GRU) on the original feature set and after feature selection.

2. SENTIMENT ANALYSIS ALGORITHMS AND TOOLS

As introduced in chapter one, Sentiment Analysis utilizes new technologies. In this chapter, all tools, techniques, technologies, architectures, concepts, and algorithms that have been used for implementing this work will be briefly discussed.

Python, which is used as a primary programming language, feature extraction techniques are described. Overview of feature selection methods is presented, followed by describing feature selection methods, namely, Principal Component Analysis (PCA) and Chi-Square.

Finally, three deep learning models, Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (Bi-LSTM) Moreover, Gated Recurrent Unit (GRU), will be described accurately in the upcoming sections.

2.1. Python

Python is a dynamically semantic, interpreted, object-oriented, high-level programming language. It combines high-level built-in data structures with dynamic typing and dynamic linking to make it suitable for Rapid Application Development and as a scripting or glue language for connecting existing components [38]. Python is simple, straightforward to learn, and syntax emphasizes readability, hence decreases the cost of program maintenance.

Python modules and packages are more programmatic than object-oriented, allowing the program to be structured into smaller pieces and reused modules and packages. Source and binary versions of Python's standard library are accessible on all major platforms without charge. Like several other dynamic languages, general-purpose languages, Python is commonly used by the data science community. When it is commonly referred to as the simplest to comprehend and understand, it is frequently referred to as the Go language. It has become the primary programming language for open data science because it can make faster algorithm improvements and include FORTRAN or C algorithms. As AI, machine learning, and predictive analytics continue to advance, Python professionals' demand rises. It is commonly used for web creation, scientific computing, and data mining [39].

2.2. NLTK

The technology utilized to help computers understand humans' natural language is Natural Language Processing (NLP). It is a subfield of artificial intelligence that deals with computers and humans through natural language. To extract meaning from human languages, most Natural Language Processing techniques depend on machine learning. Natural Language Toolkit (NLTK) is a forum for developing text analysis programs. In 2001, Steven Bird and Edward Loper published

the platform as part of a computational linguistics course at the University of Pennsylvania [40]. This toolkit is one of the most popular NLP-based libraries, containing packages for making machines understand human language and react appropriately. Tokenization, Stemming, Lemmatization, Punctuation, Character count, Part of speech tagging (POS), and word count are some of these packages that are used in text sentiment analysis [41]. In this study, all applied pre-processing, preparation, and algorithms have been implemented using Python as the primary programming language, Python version of a TensorFlow library and Keras.

2.3. Word Embedding

Word embedding is a main deep learning technique used in the numeric representations of words that are useful to solve the problem in natural language processing. Neural networks in NLP do not receive raw words as input since the networks can only understand the numbers. Hence, words should transform into feature vectors or word embedding's [42]. The word vectors can be learned by feeding a large group of raw text into a network and training it for a sufficient time. After training word embedding, it used to extract similarities between words or other relationships.

This method has gotten a great deal of consideration in the text, including sentiment analysis, due to its capabilities to take the syntactic and semantic similarities between words. For example, vectors for the words food and rice will have higher similarities than the rice and car vectors. Recently, different strategies developed to create meaningful models that can learn word embedding is from huge texts. The most popular methods are word2vec [43, 44] and global vectors (Glove) [45]. At present, both methods are among the most reliable and useful word embedding methods that can turn words into meaningful vectors.

2.4. Dimensionality reduction methods (Feature selection methods)

There are lots of features, many of which are either unused or have little data load. If you do not delete these features, it does not pose a problem in terms of information, but it does increase the computational load for sentiment analysis by the network. Also, it allows us to store a lot of useless data along with valuable data. Many solutions and algorithms have been proposed for the feature selection problem.

On the other hand, the large volume of features reduces the accuracy of classification results in the test data. In other words, the problem of overfitting occurs. Overfitting happens when the learning model has a high error in the test data and does not have generalizability on test data. This problem occurs when there is a high correlation between the independent variables, and the model may not have a high validity despite the high value of the correlation coefficient between the independent and dependent variables. In other words, although the model looks good, it does not

have meaningful independent variables. To select effective features and solve this problem, the method of features dimension reduction has been used. Selecting effective features for classification is an NP-Hard problem and can be solved using evolutionary algorithms.

Feature reduction has a significant impact on the text classification results [46]. Therefore, it is incredibly crucial to determine the suitable selection algorithm to reduce dimensions: Mutual Information (1v1I), Information Gain (IG) [47, 48], Term Frequency-Inverse Document Frequency (TF-IDF) [64], Gini Coefficient (GI, Principal Component Analysis (PCA) and Chi-Square Statistics (CHI) are some of the standard feature reduction algorithms. To increase the text classifier's scalability, Chi-square and PCA are two-dimensionality reduction approaches used in combination with deep learning models.

2.4.1. Feature Selection Methods based on chi-square

Chi-square correlation-based feature selection is one of the filter-based feature selection methods. This feature selection method measures the significant difference between the observed frequency and the two features' expected frequency. According to the two-variable data, the chi-square statistic is predicted based on the difference between what is seen in the data and what is expected in the absence of a relationship between variables. To fully appreciate the concept, two feature categories are studied [49] namely: independent (or predictor) and dependent (or response). The features that are heavily dependent on the reaction are being selected. However, when both are independent properties expected, and observed values are similar, they have a lesser Chi-Square value. Producing a significant value of the Chi-square is means having an error in the independence hypothesis. To make it more concise, any feature with high value can be used to train the model.

$$x_c^2 = \sum \frac{(O_i - E_i)^2}{E_i} \tag{2.1}$$

In this formula, we show the observed frequency with O, the expected frequency with E, and period i. The high value of chi-square indicates that the hypothesis of independence of two features is not correct. As the correlation between the independent and dependent variables increases, the chi-square index shows a weaker fitting. The chi-square value's sensitivity is to the sample size (this index is usually significant in the high number of samples). If there is a correlation between the independent category feature (predictor) and the dependent category feature (response), the probability of correct prediction of the class occurrence increases. As a result, the feature with the high value of chi-square is selected as the associated feature.

2.4.2. Feature Selection Based on Principal Component Analysis

The principal component analysis (PCA) method is a multivariate statistical method that is used to identify the causes of changes and unknown data according to the data structure. In this method, data classification with the help of a linear conversion returns the data in terms of maximum values of their variance. This conversion takes the data to a new coordinate system where the largest variance of the data is on the first coordinate axis, the second variance on the second coordinate axis, and same goes for other data. With this point of view, this method is also used to reduce dimensions. Thus, the components that have the larger variance remain to determine the changes process. On the other hand, this method is also used to analyze and classify large volumes of data because it establishes a special relationship as a model between large numbers of variables that are not seemingly related.

PCA seek new set of bases which correspond to the highest variance in the data and Transform n-dimensional normalized data to a new n-dimensional basis, the new dimension with the most variance is the first principal component.

Let us assume that the original matrix includes 'b' observations and 'a' dimensions, and it is required to decrease the dimensionality into a 't' dimensional subspace, then its transformation can be given by the following equation.

$$Y = (E^{z} X) \tag{2.2}$$

In the above equation, E is the projection matrix that contains eigenvectors corresponding to t highest eigenvalues, and where Xab is the mean centered data matrix [49].

2.5. Classification Algorithms

In general, classification can be explained as categorization in which concepts and objects are recognized, differentiated and understood. In data mining and machine learning, classification can be referred to as the problem of which a new observation belongs to a class. Classification is supervised learning in which there is a predefined class for classifying new instances. The process of training a machine learning model involves providing an algorithm with training data to learn from. Furthermore, the testing data is used to see how good the algorithm can predict new instances based on its training. The result of the testing of an algorithm determines its performance or accuracy. In this section, deep learning techniques that were used in this work will be explained:

2.6. Deep Learning techniques

Deep learning which a specific subfield of machine learning, in deep learning, the software trains itself by exposing multilayer neural networks to vast amounts of data. Advances in deep

learning have provided excellent natural language processing results, including sentiment analysis across multiple datasets [50]. The most significant value of deep learning is that we do not need to extract features manually. Instead of that, they take word embedding as input containing context information, and the middle layers of the neural network learn the features during the training phase by themselves. Words are expressed in the high dimensional vector and feature extraction performed by the neural network [51].

Deep learning starts very rapidly because it provides superior performance on various issues and makes problem-solving much more comfortable because it is fully automatic.

The deep neural network is about assigning inputs to targets through a deep chain of simple data transformations (layers), and such layers are learning by observing many samples of input and targets. Transformation, performed through a layer that parameterized by own weights, also termed parameters. Learning layers means discovering a series of values for the weights of all layers in the network so the network will precisely set the input samples for the targets associated with them. The deep neural network can include many million parameters, and getting the right value for each parameter looks like a dispiriting duty. Changing the value of the individual parameter will influence the behavior of all other parameters [51].

2.7. Sentiment Classification models

In sentiment analysis, classification essentially means categorizing data into different classes based on some calculations to determine the sentiment of the text as Positive or Negative. In this study, three supervised deep-learning algorithms were applied, namely, Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (Bi-LSTM) Moreover, Gated Recurrent Unit (GRU).

2.7.1. Recurrent Neural Network (RNN)

A Recurrent Neural Network is an artificial deep neural network. This method is used in several NLP studies. They are designed to identify the characteristics of a sequence of data Recurrent neural networks were created in 1980, but such networks have been widely used in recent years. The main reasons are the progress generally made in the design of neural networks and the significant improvement in computing power, particularly the efficiency of parallel processing units of graphics cards. These kinds of neural networks are useful for processing hidden or sequential data, in which each neuron or processing unit can maintain an internal state (memory) to retain the data of previous input as shown in the Figure 2.1. This feature is significant in various applications related to hidden data [52].

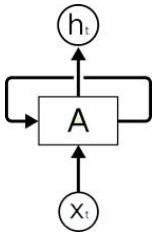


Figure 2.1. Recurrent Neural Network

This feature helps the network maintain the internal state to understand and discover the connection between different words in longer sequences. It should be noted that when we read a sentence, we infer its meaning according to the context in which each word is placed [53].

The main idea of this type of architecture is to exploit this data series structure. The name of this neural network is derived from the fact that these types of networks operate recursively. It means this operation is performed for each element of a sequence (word, sentence), and its output depends on the current input and previous operations. This is done by repeating one output of the network at the time *t-1* with the network's input at the time *t* (that is, the output of the previous step is combined with the new input of the current step) [54]. These cycles allow data to enter from a one-time step to the next time step. In other words, these types of networks have a loop inside them through which they can pass data to neurons while reading input Figure 2.2.

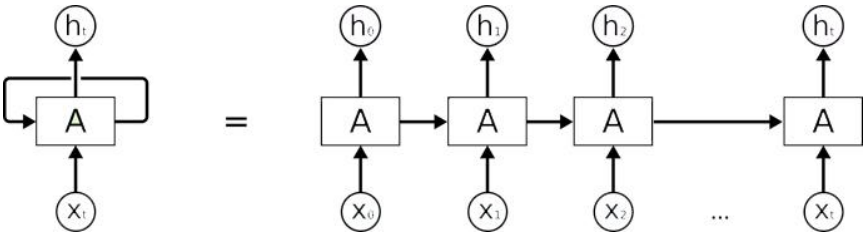


Figure 2.2. Opened loop of recurrent neural network

The recurrent neural network needs data from the beginning of the sentence to predict correctly. These dependencies are called long-term dependencies; there is a relatively significant distance between the relevant data and the point at which this data is used to make a prediction. Unfortunately, in practice, the greater the distance, the more difficult it to learn these dependencies because they either have the problem of gradient vanishing or gradient explosion [55].

When gradients are transferred from the end of the network to the beginning of the network in the back-propagation process, these problems occur during deep network training. The gradients coming from the end layers have done several multiplications to reach the beginning layers, and thus their values start to shrink (less than 1). This process goes on, and these values become so

small that (in very deep networks) the training process is disrupted and practically stopped because the gradients have minimal values that do not affect the weight changes. This problem is called gradient vanishing. In the same way, there is a possibility of gradient explosion, and the values of the gradient gradually become so large that eventually, the model will make a mistake, and an overflow will occur in the calculations. We can better understand the nature of the loop in this kind of network by opening it Figure 2.3.

A traditional recurrent neural network should theoretically produce sequences of any complexity (if considerable enough), but practically, we find that this network is incapable of storing data related to past inputs for a long time. In addition to weakening the network's ability to model long-term structures, this "forgetting" causes these networks to be exposed to instability during sequence generation. The problem (which is common to all conditional generation models) is that if the network predictions depend only on a few recent inputs and these inputs are generated by the network itself, and there is very little chance for the network to correct past mistakes.

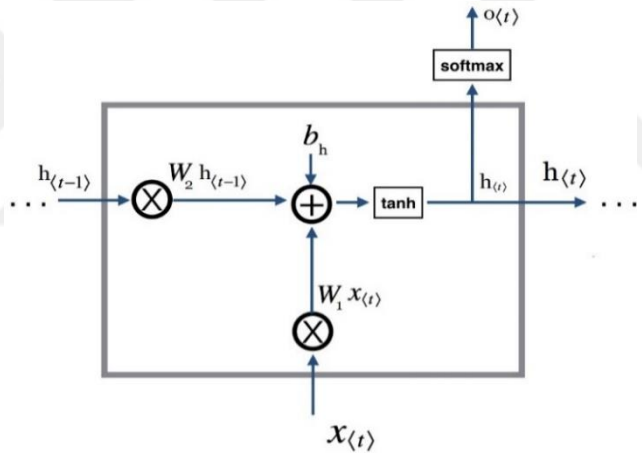


Figure 2.3. Details of a recurrent neural network cell

As shown in the Figure 2.4., in each time step, we receive two inputs, and two outputs are generated. There is no mechanism for transferring a feature from the initial steps to the final steps in this structure. There is no mechanism to deal with the issue of gradient vanishing [55-56].

$$h_t = tanh(W_1 x_t + W_2 h_{t-1} + b_h) \tag{2.3}$$

$$O_t = softmax(W_3 h_t + b_o) \tag{2.4}$$

Practically, we see that at each time step, the contents of each cell are replaced by new values from the previous time step and the new input, and our hidden memory vector is only effective for the very last few limited time steps (i.e. the effectiveness is local and limited to a few recent time

steps). We see this in the picture below Figure.2.4. We have also said before that theoretically, the network should be able to do this. It should be able to work with any sequence length, but in practice, this does not happen. Why? The reason is that there is nothing to impose on this issue. There is nothing in the network's mechanisms that forces the network to retain a feature from previous time steps and ignore new inputs (although, in theory, this cannot be done flawlessly unless changes occur in the activation function).

Having a longer-term memory has a stabilizing effect because even if the network fails to understand its recent history, it can still complete its prediction by looking back. Long short-term memory or LSTM is a recurrent neural network architecture designed to store and access data better than the traditional version.

LSTM is a specific type of RNN that has more advanced functions and manages information flow. The standard RNN has an issue of gradient vanishing or exploding. To conquer these issues, an LSTM intended by Hochreiter and Schmidhuber [56]. Figure 2.4., shows an image of a standard recurrent neural network cell.

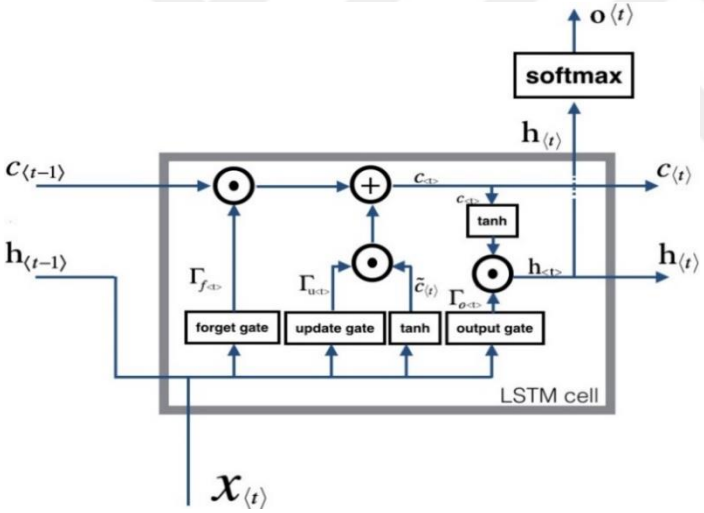


Figure 2.4. Three gates through which the network controls the data flow

2.7.2. Sentiment Analysis Using LSTM Model

LSTM is a specific type of RNN that has more advanced functions and manages information flow. The standard RNN has an issue of gradient vanishing or exploding. To conquer these issues, an LSTM intended by Hochreiter and Schmidhuber [57]. Unlike a traditional recurrent neural network in which content is rewritten at every step, in an LSTM recurrent neural network, the network can decide on preserving current memory through the introduced gates. Intuitively, if the LSTM unit detects an essential feature in the initial steps' input sequence, it can easily transmit this data over long distances. So, it receives and maintains such possible long-term dependencies. The

LSTM network was first introduced in 1997. Minor changes have been made to LSTM since then. In the LSTM neural network, we encounter new concepts that did not exist in the traditional recurrent neural network [58]. In this network, there are three gates through which the network controls the data flow Figure 2.5.

- Forget gate

The forget gate,Γ_f is responsible for controlling the data flow from the previous time step. This gate determines whether the memory data from the previous time step can be used or not, and if something has to be entered from the previous time step, to what extent should it be.

$$\Gamma_f = \sigma(W_f.[h_{t-1}, X_t] + b_f) \tag{2.5}$$

Here W_f is the weight matrix that controls the behaviour of the forget gate. For simplicity, we combine the X_t and h_{t-1} vectors. If we do the above operation, because we are using the sigmoid activation function, the result will be a vector called Γ_f, which will have values between zero and one. This vector will then be multiplied by C_{t-1} in the next phrase. Therefore, if the forget gate values are zero (or tend to zero), it practically means that the content of C_{t-1} is not considered (the network throws away the data provided by C_{t-1} and pays no attention to it). Similarly, if the vector Γ_f values are equal to one, this data is stored by the network. Intermediate values also cause the network to use the same amount of content from the previous time step (i.e. one part is discarded, and the other part is used) [58].

- Update Gate or Input Gate

The update gate shown as Γ_u controls the flow of new data. This gate determines whether new data should be used at the current time step or not, if the answer is yes, to what extent should it be.

$$\Gamma_u = \sigma(W_u.[h_{t-1}, X_t] + b_u) \tag{2.6}$$

In order to update weights, we need a new vector that we can add it to the previous memory state.

$$\hat{c}_t = tanh(W_c.[h_{t-1}, X_t] + b_c) \tag{2.7}$$

And finally, we update the memory:

$$c_t = \Gamma_f \circ c_{t-1} + \Gamma_u \circ \hat{c}_t \tag{2.8}$$

In the above phrase, first part specifies how much data is used from the previous part (memory from the previous time step) and the second part contains new data that is used.

- Output gate

The output gate, shown as Γ_o, specifies how much of the previous time step data is transferred to the next time step data along with the current time step data.

$$c_t = \Gamma_f \circ c_{t-1} + \Gamma_u \circ \hat{c}_t \tag{2.9}$$

$$h_t = \Gamma_o . tanh(c_t) \tag{2.10}$$

In addition to these three gates, there is a memory cell, or C for short. The network has an input of hidden memory or h and input or X and produces two outputs (one output is Ct and the other output is h_t which is itself divided into two parts, one part is transferred to the next time step, and another part is used to generate output at the current time step if needed.) We said that the LSTM network is the most popular solution to the gradient vanishing problem. Now that you have seen and understood the above mechanisms, you should realize the reason for this critical issue. The designed mechanisms play two primary roles: First, each unit can easily remember a particular feature in the input flow if necessary for the next long time steps. Any feature recognized as necessary by the LSTM forget Gate remains and is transmitted without being overwritten.

More importantly, these new mechanisms in practice create shortcuts that ignore several time steps, thus allowing the production error to be easily transferred to the post-propagation phase without fading too quickly, and this reduces the problems associated with vanishing gradients. Although the LSTM has more durable memory units than RNNs, it still ignores information that is far from the current point. This issue becomes more apparent when analyzing sentiment at the document level. To store more extended information by LSTM, various models have been proposed to increase the ability of LSTMs for storing long-term information [57].

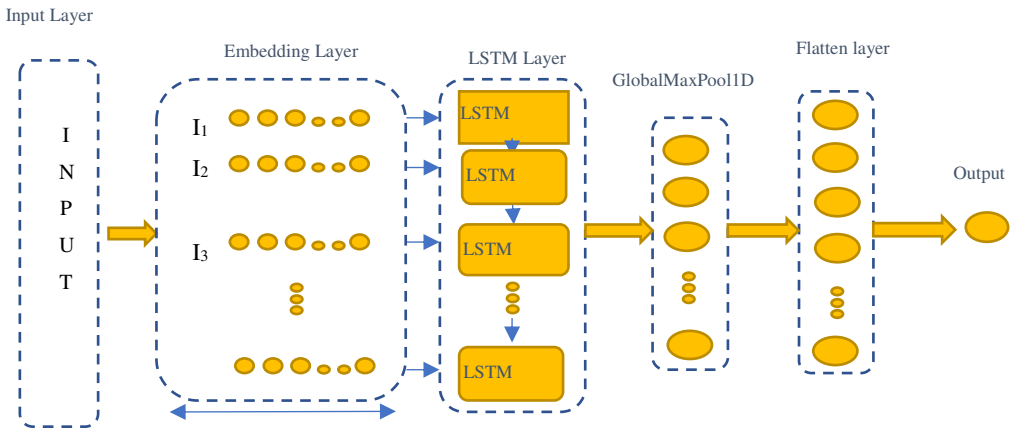


Figure 2.5. Sentiment Analysis using LSTM Model

The structure of sentiment analysis using the LSTM model is shown in Figure 2.5. The input layer contains the data sample as a sequence of unit indices of the same length; in the LSTM layer,

the information is propagated completely forward, indicating that the state of time t depends only on the information before time t. The outputs of the LSTM layer are transferred to the maximum global pooling layers. The Flatten layer integrates the data obtained from the pooling layer into a single layer, and the binary entropy is used as a loss function for binary sentiment classification in the output layer. We have presented it based on Equation (2.11).

$$Binary\ cross\ entropy = -\frac{1}{m}\sum_{i}^{m}(y_i * log(p(y_i)) + (1 - y_i) * log(1 - p(y_i))) \quad (2.11)$$

In equation (2.11), m represents the total number of predicted samples, y_i represents the real labels, and $p(y_i)$ represents the probability of the real labels.

2.7.3. Sentiment Analysis using Bi-LSTM model

The Bi-LSTM model consists of two LSTM layers and is able to read input reviews in both forward and backward directions. Forward LSTM processes information from left to right, and its hidden state can be represented as $\vec{h}_t = LSTM(x_t, \vec{h}_{t-1})$; while backward LSTM processes information from right to left and its hidden state can be shown as $\overleftarrow{h}_t = LSTM(x_t, \overleftarrow{h}_{t-1})$. Finally, the Bi-LSTM output can be summarized as h_t = [h ⃗_t, h ⃖_t] by merging backward and forward state Figure 2.6.

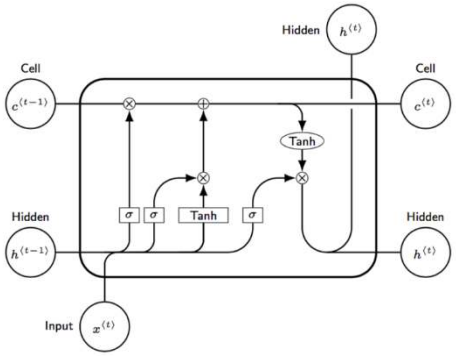


Figure 2.6. The Structure of Bi-LSTM Network

The structure of sentiment analysis using the Bi-LSTM model is shown in Figure 2.7. The input layer contains the data sample as a sequence of unit indices of the same length, and the outputs of the Bi-LSTM layer are transmitted to the global maximum and global average of the pooling layers simultaneously. The previous and next layers retrieve each feature's maximum and average values in the Bi-LSTM layer, respectively [58]. We used only one global pooling layer as an option for the dense layer(s). The inter-connected layer takes the top global and average global layers and

merges them into a single layer before passing to the final layer, considering binary entropy as a loss function in the output layer [59, 60].

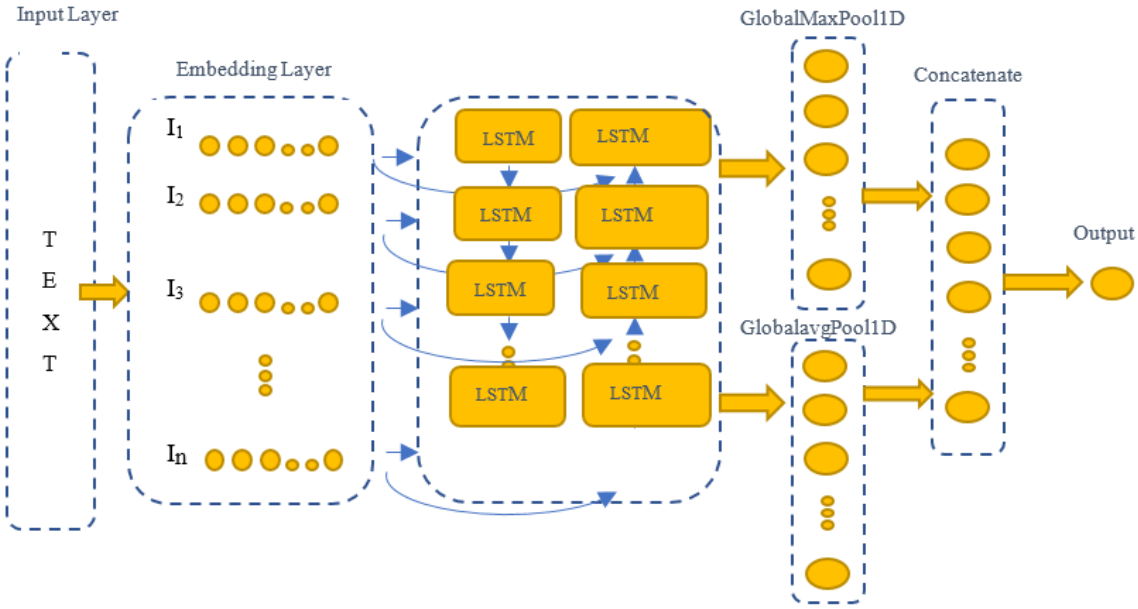


Figure 2.7. Sentiment Analysis using Bi-LSTM Model

2.7.4. Sentiment Analysis using GRU model

The Gated Recurrent Unit network architecture was proposed by Chou et al. in 2014. This architecture is designed to address the shortcomings of traditional recurrent neural networks, such as gradient vanishing problem and reducing overload in LSTM architecture. GRU is generally considered as a modified version of LSTM because both of these architectures have the same design and, in some cases, give excellent results in the same way. GRU uses concepts called update gateway and resets gateway. These gates are two vectors used to decide whether data is transmitted to the output. The particular point about these gates is that they can be trained to retain data about long time steps without changing over time (during different time steps). In Figure 2.8., the structure of the LSTM network is compared with GRU [61].

One of the main problems in the LSTM network is that the ability to retain multiple features from several steps back in time is associated with problems. In the GRU recurrent neural network, the concept of update gates has been used to solve this problem. An update gate is a switch that specifies whether to use the previous state or input (or a combination of both) in a time step. The network can easily affect a state from several previous time steps in the next few time steps in long sequences using this new feature. In other words, the network will keep elements from the distant past in its memory and exploit it [62].

$$c_t = \Gamma_u \circ \hat{c}_t + (1 - \Gamma_u) \circ c_{t-1} \tag{2.12}$$

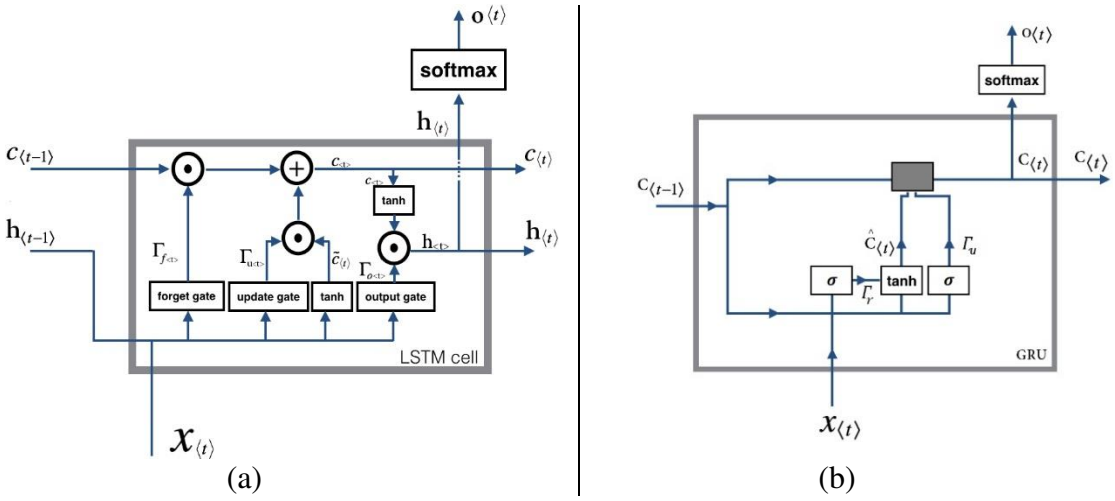


Figure 2.8. The structure of LSTM network. (b) The structure of GRU network

The general idea here is that you can specify to use the state related to the previous time step or the new state using 1 or 0 gamma, according to the new input and the previous state in a time step. More precisely, the network can bring the result of the *Γu* to zero by tending *w2* to negative numbers, and thus the beginning of the term became ineffective, so, the previous state is used and transferred to the next time step. Here, the update gate acts as a switch that allows the network to specify how much data from previous time steps is needed for the current time step [16].

In the GRU network, the reset gate displayed as *Γr*, which essentially acts as a switch that allows the network to determine how much past data is not needed in the current step and how much of the previous step data could be used in the current step. More precisely, if the switch became zero, the gate forces the network to act as if it were reading the first part of the input sequence, enabling the network to forget the previously calculated state and provide an intermediate state by moving away from zero [62].

$$\Gamma_r = \sigma(W_3.[C_{t-1}, X_t] + b_r) \tag{2.13}$$

To update, we need a new vector that we can compile with the previous memory state.

$$\hat{c}_t = tanh(W_r.[\Gamma_r.C_{t-1}, X_t] + b_c) \tag{2.14}$$

And finally, we update the memory:

$$C_t = \Gamma_u.C_t + (1 - \Gamma_u).C_{t-1} \tag{2.15}$$

The GRU neural network can easily store or filter data from previous time steps using these two new capabilities (update gate and reset gate) and exploit the long sequences - in which the traditional recurrent neural network has faced many problems - and eliminate the shortcomings of the traditional recurrent neural network. These types of networks are very powerful and have been

widely used due to their lower overload than LSTM and their greater power than traditional RNNs [62]. The structure of sentiment analysis using the LSTM model is shown in Figure 2.9.

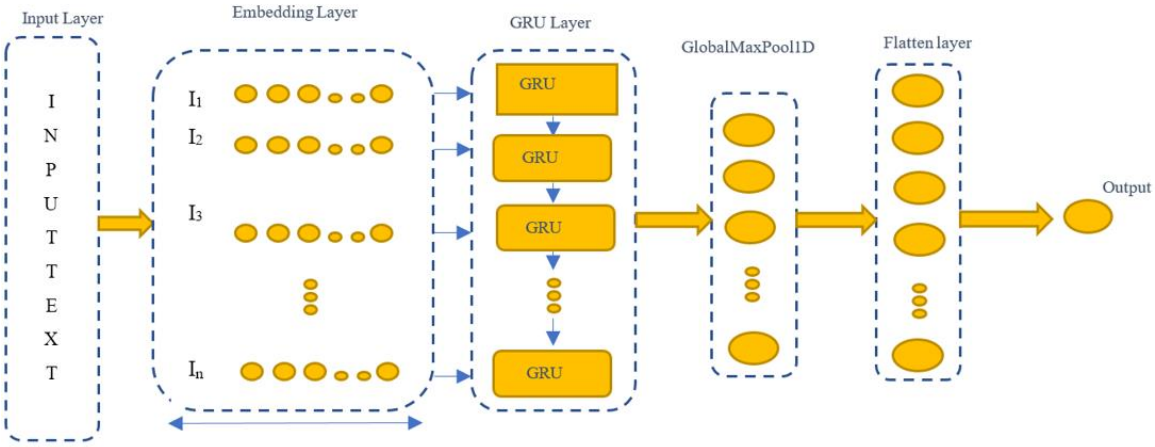


Figure 2.9. Sentiment Analysis using The GRU Model

3. IMPLEMENTATION AND ANALYSIS OF PROPOSED SYSTEM

In this chapter, the implementation and analysis of the proposed system for classifying sentiment analysis data using data mining and machine learning techniques are presented in detail.

3.1. Introduction

The applied algorithms that will be discussed in this chapter have been implemented in Python Jupiter Notebook that is an open-source web application that allows developers to build and share documents that contain source codes, equations, visualizations and narrative text. Uses include data cleaning and transformation, numerical simulation, statistical modelling, data visualization, machine learning, and much more.

In this research, the following benchmark datasets have been used (Yelp, US Airline, and, IMDB) to investigate sentiments about each one. Then each dataset went through the preprocessing step, which incorporates a variety of data cleaning and text normalization processes. Then feature extraction techniques used to generate features from cleaned datasets.

Finally, generated features were classified using three different supervised deep learning models, Namely, Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (Bi-LSTM). Moreover, Gated Recurrent Unit (GRU) in three different manners with each model as followings: classification on the original feature set, classification on the reduced feature set by using Chi-2 and PCA feature selection method.

3.2. Proposed System Model

Selecting the input that is right, intelligent, and fit to designing a learning model for prediction is of particular importance. The structure of the LSTM is not a predetermined program, and its weights in the learning process are determined based on the input data. Therefore, if the input data be more prosperous, the network is trained better, and also its performance will be better in predicting new data. This approach seeks to find the relations between the input variables to achieve the desired goal without considering and solving the model's equations. After identifying the goal-related data, preprocessing methods are used to know more detailed information about this set. There are unknown and unusual data in the selected set of inputs. On the other hand, the raw data used to learn the model includes data related to each other due to overlapping inputs. This correlation throughout the learning process can confuse the network in achieving the desired goal and reduce its generalization ability. Data preprocessing avoids using trial-and-error methods and recognizes the most important variables affecting modelling using intelligent techniques [59-60].

In Figure 3.1., which shows the main points of our proposed model architecture. In this model, there is an input layer, an ensemble of global maximum and average pooling layer, an average layer, an LSTM layer, a global maximum layer, and a sigmoid layer at the output. Further details of every layer are provided in the following subsections.

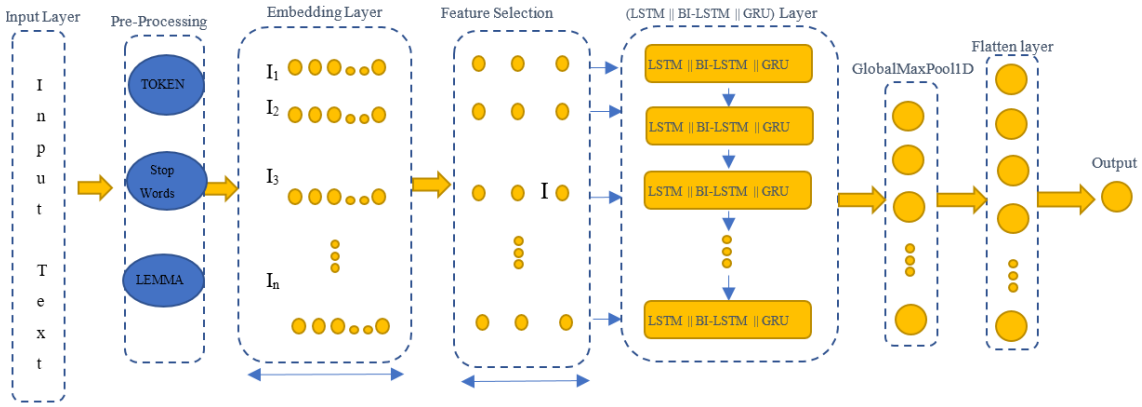


Figure 3.1. Proposed model

3.2.1. Embedding Layer

Feature extraction is converting text data into a set of features or numerical representation of words or phrases. The main concept of feature extraction in sentiment analysis is that all the words used in a sentence or text can be expressed by a set of decimal numbers (in the form of a vector). Each set of numbers is considered a reliable "word vector" that is not necessarily useful to us; a set of word vectors that is useful for our intended applications is the word vector that understands the meaning of words, the relation between them, and the content of different words as its natural way. All word vectors are stacked in a word embedding matrix $L_w \in R^{d \times |V|}$, when |V| is the size of words, and d is the dimension of word embedding. These word vectors can be pre-taught from the text's body with embedding learning algorithms such as Word2vec and Glove. The word2vec algorithm uses a distributed representation for each word. Consider a multi-dimensional vector as an example. Each word is represented by a distribution of numerical values (weights) on different vector elements. Thus, instead of a one-to-one mapping between an element in a vector and a word in the dictionary. The representation of a word is spread across all the elements in a vector, and each vector's element plays a role in determining the meaning of a large number of words [62].

3.2.2. Feature Selection layer

Generating data on the scale of millions is just a matter of seconds on Twitter, through shared tweets and retweets. When data analytics and sentiment analysis occur, it has to deal with a tremendous amount of features generated from Twitter text. Using TF-IDF and BoW as feature

extraction techniques generates many features when used to generate different n-gram ranges. This introduces the problem of high-dimensionality of data, which leads to data sparsity and high dimensional feature space. This problem affects the performance of classification algorithms. Also, high dimensionality means a large number of features, which makes learning algorithms to overfit and increases time and space requirements. So, high dimensionality should be handled before applying sentiment analysis classification because not all of the generated features are relevant to sentiment analysis tasks, as well as they, contain redundant data. High dimensionality the problem of feature space can be solved in an automatic manner by using feature selection methods that recognize patterns within data, which retains the most informative features for classification task and eliminates the irrelevant and redundant features. So, it's always vital to use the feature selection techniques while dealing with data of high dimensional nature. The proposed model makes use of two feature selection methods, namely, Chi-2 is first applied to the original feature space to remove irrelevant data. Then, PCA is second applied to the original feature space to remove irrelevant data [63].

3.2.3. The layers of the LSTM models

Three different LSTM models are included in this layer to better understanding their performance. In the LSTM model, the information is fully forwarded, indicating that the status of time t depends only on the information before time t. However, to describe the input review's general meaning, the following information is just as effective as the previous information. Therefore, for better representing the textual information, the Bi-LSTM model was used. Therefore, to better display the textual information, the Bi-LSTM model was also tested in the layer. However, the ability to retain several features from several previous steps was problematic in the Bi-LSTM model. Therefore, in this layer, the GRU network model was also examined. In this model, the forget gate and the input gate are combined and converted into a single update gate [63].

3.2.4. Global maximum Pooling Layer

The outputs of the LSTM layer are transmitted simultaneously to the global maximum Pooling layer. The previous and next layers retrieve the maximum values of each feature in the LSTM layer, respectively. We used only one global pooling layer as an option for the dense layer(s) [64].

3.2.5. Flatten layer

A flattening layer collapses the spatial dimensions of the input into the channel dimension. Flatten layer is used to flatten the input. If flatten is applied to a layer with input shape as batch size, 2,2, then the layer's output shape will be batch size, 4 [64].

3.2.6. Output

In the output, we use binary cross-entropy as a loss function to classify binary sentiment.

3.3. Experimental Setup

This analysis was performed on a computer running on core-i7, using the 16GB RAM version on the experiment and an Nvidia GeForce GTX 1080 Ti GPU. We used Python 3.7 as a supported programming language in TensorFlow, and it was set up for us with the TensorFlow version 1.14 and the latest version of the Keras library. In the following subsections, we summarize the datasets, supported by the preprocessing of data, and finally, we explain the hyper parameter settings to optimize our proposed system.

3.4. Analysis of Datasets

Three benchmark sentiment analysis text datasets, US Airline, Yelp, IMDB, were used. Each dataset, including the negative and positive labels, consists of the text sentiment column and the label column. We also divided the dataset into an 80% training dataset used to train the model and a 20% testing dataset to evaluate the model. In the details of each dataset.

3.4.1. Yelp

Yelp dataset is collected from real Yelp businesses and aimed for academic purposes. It has 366K user profiles in a total of 2.9M edges, and 1.6M reviews and 500K tips for 61K businesses and total check-ins over time for each of the 61K businesses. The data of four countries (U.S., UK, Germany, and Canada) are collected to make the dataset diverse. All the data are separated into 5 JSON files: business, review, user, check-in, and tip. The content of them is accessible to understanding. The business file explicitly presents each business's properties, such as the business's unique internal ID, name, geo-location, classes, and so on. The check-in file is a supplementary feature of each business. It offers the collected check-in time for each business in each hour of the week. The user file includes basic information about each user, like yelping time, name, fans, and social correlations with other users. Tip and review files are the opinion of a specific user to a business.

The difference between tip and review is that the tip always presents short, while the review is long enough for users to express their feedback. In this work, I used 10000 record that have been split into 6863 positive and 3137 negative sentiment. You can find YELP Dataset in reference [65]. The dataset was obtained from Yelp Dataset Challenge in 2015. There are five levels of ratings from 1 to 5.

3.4.2. US Airline

The US Airline dataset was extracted from multiple sources for the top 10 US airline carriers, such as Twitter tweets and online reviews of Skytrax. It analyzed a total of 14640 tweets from 7700 users. Twitter data was scrapped as of February 2015 and contributors were asked to identify positive, negative and neutral tweets first, followed by a categorization of negative reasons (such as "late flight" or "rude service") [66].

3.4.3. IMDB

The IMDB dataset contains 50K movie reviews for natural language processing or text analysis. It is a dataset for binary sentiment classification which contains considerably more data than previous benchmark datasets. We prepare a category of 25,000 highly polar movie reviews for training and 25,000 for testing. Therefore, we use classification or deep learning algorithms to predict the number of positive and negative reviews [67].

3.5. Preprocessing of Text Data

Pre-processing is a significant step in converting the text in a human language into a machine-readable form for more processing. It affects the efficiency of other steps. The pre-processing step aims to make the data more machines readable to reduce ambiguity in feature extraction. In this work, some steps were used to normalize the text, converting the upper case to lower case, removing duplicate text, and stopping words, numbers, multiple spaces, special characters, a single character, and punctuation marks URLs, Html, mention, and hashtags. Also performing lemmatization for words, it is the process of substituting words with a stem or base words to decrease inflectional structure to a typical root structure and expand slangs and abbreviations [68].

3.5.1. Basic Pre-processing Steps

Due to unstructured nature of data that include data that contain no information for sentiment analysis, and they need to be removed through some basic steps of preprocessing and normalization procedures which are performed as follows [68]:

1. Remove duplicate sentiment texts
2. Lowercase all texts
3. Replace emojis with their meaning
4. Remove URLs
5. Remove user mentions
6. Remove Hashtag sign
7. Remove email address
8. Parsing HTML tags
9. Slang correction: is the process of converting abbreviated slang phrases into expanded original phrases i.e converting "omg" to "oh my god", "4u" to "for you".
10. Expand contraction: is the process of converting abbreviated English helping verbs into expanded form i.e "can't" to "can not", "we'll" to "we will".
11. Remove punctuation marks such as ", "", ?, ., etc.
12. Remove numbers
13. Remove extra white spaces
14. Reduce repeated characters to only two i.e "veeeeeery" to "veery", and "gooooooood" to "good".

After all preprocessing steps stated above, we removed duplicate word for the second time, to make sure there is no duplicate words.

3.5.2. Tokenization

The process of breaking down sentences into single words is called Tokenization. The split words are called Tokens. The purpose of Tokenization is aiding further analysis, such as removing stop words, and lemmatization.

3.5.3. Remove Stop Words

Stop words are most commonly used words in languages, and contain no sentiment such as: "we", "They", "Are". Stop words should be filtered out before sentiment classification take place.

3.5.4. Lemmatization

Lemmatization is the process of obtaining the lemma, or base form of the words called lemmatization. Example: "hoping" to "hope". Figure 3.2 shows a sample of pre-processed text sentiments and Figure 3.2., shows the overall pre-processing stages.

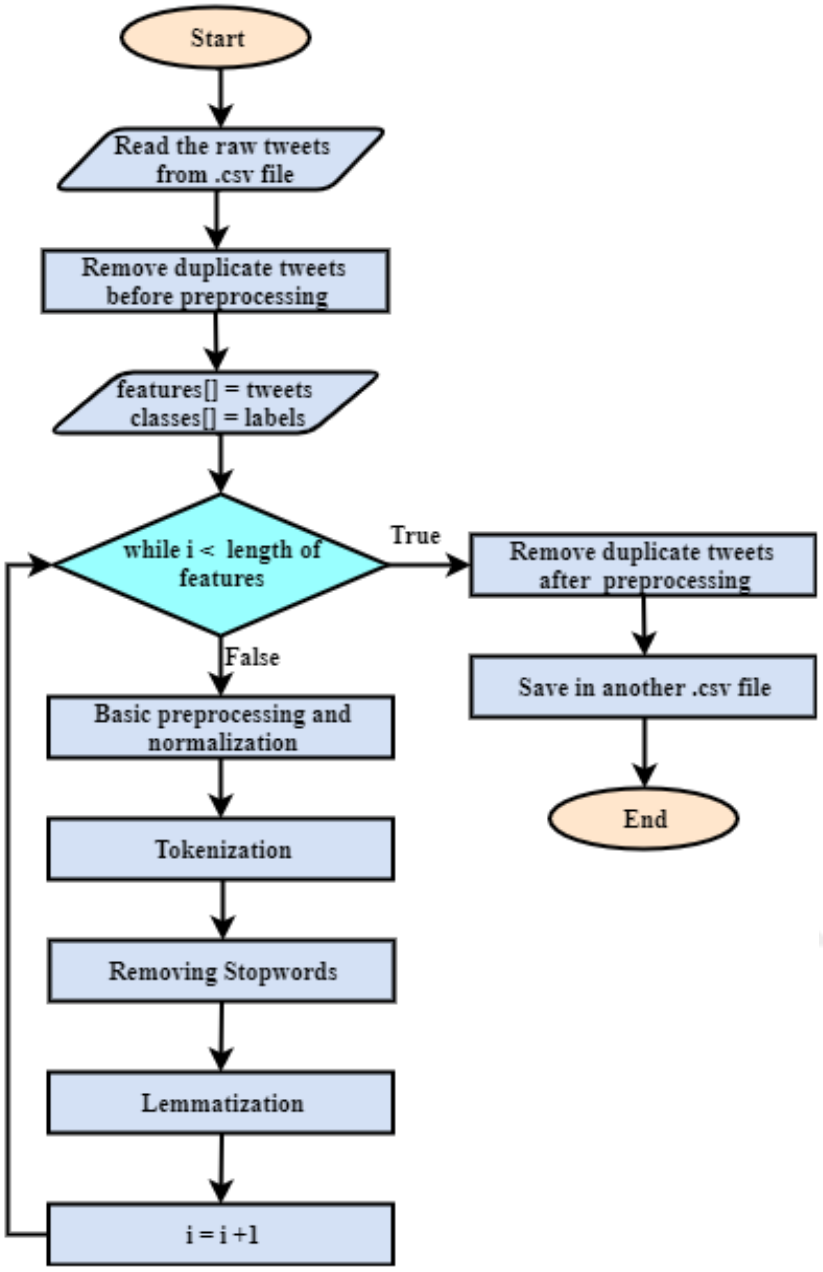


Figure 3.2. Flowchart of Pre-processing Steps

3.5.5. Hyper Parameters Setting

The result of each technique achieved in the following configurations: each of the models configured with a dropout layer to restrict the neural network from memorizing the training set, which is useful to prevent the over fitting. The models compiled with the Adam optimizer with the batch size of 128 for 10 epochs, the network for LSTM and Bi-LSTM models has 32 neurons, and 256 for GRU model, performed using Tensor flow and Keras. Also, we have created a network configuration, as explained in Table 3.1. [68].

Table 3.1. Parameters setting in LSTM, Bi-LSTM AND, GRU Models

Parameter	LSTM	Bi-LSTM	GRU
Training approach	CV	CV	CV
Optimizer	Adam	Adam	Adam
Loss function	Binary cross entropy	Binary cross entropy	Binary cross entropy
Learning rate	0.001	0.001	0.001
Batch size	300	300	300
Max. length	6000	7000	7000
Epochs	10	10	10
Hidden layer size	32	32	256
Drop out	0.5	0.25	0.25
Activation function for hidden layer	tanh	tanh	tanh
Activation function for output layer	Sigmoid	Sigmoid	Sigmoid

3.5.6. Loss function

The loss function is a measure of the suitability of the model in terms of its ability to predict new values. One common method for finding the minimum loss function is to use the derivative and the Gradient Descent algorithm. The choice of the appropriate loss function depends on several factors such as the presence of outliers or data, the type of machine learning algorithm, the cost time of the algorithm and the simplicity of calculating the derivative, and so on.

Loss function, also called the objective function, will be used to computes a distance score by comparing the prediction of the network and the real object to estimate how far the predicted output is from the real object. After computing the distance score between the predictions of the network and the real target by using a loss function, this score is utilized as a feedback sign to slightly improve the value of weights, in a way that will reduce the loss score, this improvement is a function of the optimizer that performs what is called the Backpropagation algorithm [69-71].

3.5.7. Optimizer

In the Backpropagation algorithm, at first, weights of the network are appointed with random values. Hence the network only performs a sequence of arbitrary shifts. Generally, the results are ideally far from what they should be, so the loss score is too high. However, in each case the network handles, the weights are slightly modified in the right trend, and the loss score decreases; that is, the training loop that iterated enough times, giving weight values that reduce the loss

function. The lowest loss network is the network where the outputs are closest to the targets [69-71].

3.5.8. Training approach

Cross-validation is an effective method for assessing a model's success in relation to unobserved results. In this study, 80 per cent of the data was used for training and 20% for testing.

3.5.9. Learning rate

In LSTM models, a backpropagation algorithm updates the models' weight to what is known as the phase size or learning rate. Its low value can result in tedious calculations and a longer training time, while its high value can cause system instability.

The step size or learning rate is the sum by which the weights are updated during training. The learning rate is a configurable hyper parameter used in model training with a small positive value, usually in the range of 0.0 to 1.0. If you set the learning rate too low, training will move very slowly because you are making very small changes to the weights in your network. However, setting the learning rate too high can result in undesirable divergent behaviour in the loss function [69-71]. In LSTM models, a backpropagation algorithm updates the models' weight to what is known as the phase size or learning rate. Its low value can result in tedious calculations and a longer training time, while its high value can cause system instability. The learning rate in LSTM models was 0.0001.

3.5.10. Batch size

Batch size indicates the number of samples that a neural network can process before updating its internal parameters. A smaller amount of batch size may slow down the training process, while a large amount requires high memory. We used 300 batch size to achieve better results in this study.

3.5.11. Epochs

Epochs is the number of times that the learning algorithm will work through the entire training dataset. One epoch means that each sample in the training dataset has had an opportunity to update the internal model parameters. An epoch is comprised of one or more batches. The number of epochs is traditionally large, allowing the learning algorithm to run until the error from the model has been sufficiently minimized.

3.5.12. Maximum Length

Increasing the length of input data could theoretically increase the model's predictive performance, but this might not occur due to data sparsity. Alternatively, if the maximum duration is too short, we throw away too much valuable knowledge. The overall length in LSTM models was 6000.

3.5.13. Regularization

Regularization strategy is proposed in order to ensure model success in large measure when dealing with real world problems by selecting features and discriminating between more beneficial and less helpful features. Its proper application strengthens the model's resistance to overfitting. Overfitting can be detected during training when the error on training data decreases to a very small value while the error on new or test data increases to a large value [71].

3.5.14. Dropout

It is very difficult to train models with many different structures, because it is a frustrating task to find the hyperparameters corresponding to each structure, and training models with different structures requires a lot of computing resources. Moreover, a huge network generally requires a lot of training data, but there may not be enough training data to obtain different subsets for training different networks. Even if many different large networks can be trained, in scenarios where fast response is important, it is not practical to use all large networks at the same time during actual application testing.

Dropout is a technology to solve these problems. It can prevent overfitting and provide an approximate method combining different neural network structures with different numbers of exponents. Dropout refers to dropping some units in a neural network. Discarding a unit refers to temporarily removing the unit from the network, along with all its input and output connections. The choice of which unit should be discarded is random. In the simplest case, each unit has a fixed independent probability p. p can be obtained from the validation set or simply set to 0.25 [69-71].

3.5.15. Performance Evaluation Metrics

In building any deep learning model, one of the primary tasks is to evaluate its performance. The performance of each technique used in this work is measured by computing different metrics. The ultimate purpose behind working with other metrics is to understand how well a deep learning model will perform on unseen data. In this work, the following metrics are used [71, 72]:

- Accuracy is the proportion of the accurately analyzed samples to the total number of samples.

$$Accuracy = (T_P + T_N)/(T_P + T_N + F_P + F_N) \quad (3.1)$$

- Precision is the number of accurate positive analyzed samples to the classifier's predicted positive results.

$$Precision = T_P/(T_P + F_P) \quad (3.2)$$

- The recall is the number of accurate positive analyzed samples to the number of all relevant samples.

$$Recall = T_P/(T_P + F_N) \quad (3.3)$$

- F1 score computes precision and recall of the test to calculate the score.

$$F_1 score = 2 * (Precision * Recall)/(Precision + Recall) \quad (3.4)$$

In the above equations, TP is the true positive and predicted correctly, FP is the false positive and predicted incorrectly, TN is the true negative and predicted correctly, FN is the false negative and predicted incorrectly.

4. RESULTS AND DISCUSSION

This chapter evaluation of the Results in the YELP, US Airline, and IMDB Datasets:

4.1. Introduction

This chapter summarizes the results attained by the proposed model in detail inaccuracy based on experiments in different feature selection methods and without feature selection method. The experiments are carried out on different datasets, namely, (YELP, US Airline, and IMDB). After preprocessing, all datasets are subjected to feature generation. Three supervised deep learning models have been used for classification, namely, Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (Bi-LSTM) Moreover, and Gated Recurrent Unit (GRU). The accuracy comparison of the model will be illustrated in charts and explained in upcoming sections in three different cases, including classification on original feature sets, classification after applying filter-based feature selection method (Chi-square), and classification after applying the principal component analysis (PCA) method.

We found a large difference between the accuracy of these models in training and testing data. Therefore, the prediction accuracy of LSTM, Bi-LSTM and GRU models before and after combination with feature selection methods were compared with each other. The results are presented on the following YELP, US Airline and IMDB datasets.

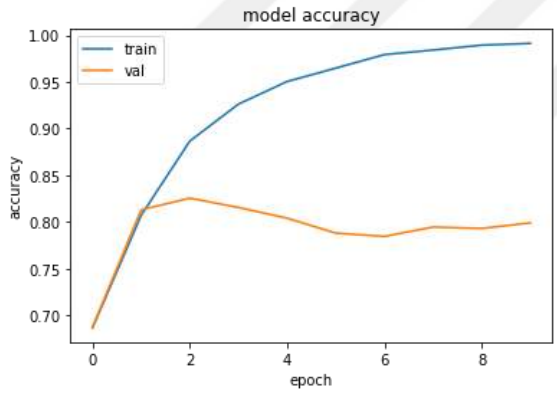
4.2. Evaluation of the Results in the YELP Dataset

The evaluation of the results of LSTM, Bi-LSTM and GRU models before and after combination with chi-square and PCA feature selection methods in the YELP dataset are presented in Table 4.1.

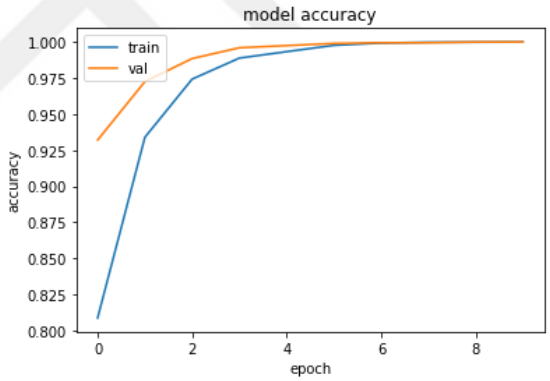
Based on the results obtained in Table 4.1., the prediction accuracy of the LSTM model on the YELP dataset before combining with feature selection methods was 78.7%, and the accuracy of this model has been increased to 99.4% after combining with the feature selection method based on chi-square correlation. Also, the prediction accuracy of combining Bi-LSTM and GRU models with chi-square feature selection method was 100% and 99.95%, respectively, and the prediction accuracy of combining LSTM, Bi-LSTM and GRU models with principal component analysis (PCA) based feature selection was 99.6%, 99.8% and 99.6%, respectively. Examining the results obtained from the hybrid models, we conclude that combining the Bi-LSTM model and the chi-square feature selection method has the best result or 100%, in the YELP dataset.

Table 4.1. Performance of the proposed model for YELP dataset.

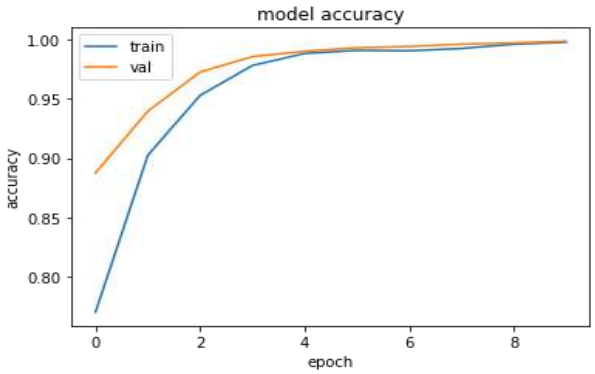
Techniques	Class	Precision	Recall	F1 score	Accuracy
LSTM	positive	65.3%	68.2%	66.7%	78.7%
	negative	52.3%	83.5%	84.3%	
Bi-LSTM	positive	67%	74.6%	76%	85%
	negative	87.8%	83.2%	85.4%	
GRU-LSTM	positive	71.8 %	58.7%	64.6%	79.9%
	negative	82.6%	89.5%	85.9%	
Chi- square with LSTM	positive	98.4%	99.8%	99.1%	99.4%
	negative	99.9%	99.2%	99.6%	
Chi- square with Bi-LSTM	positive	100%	100%	100%	100%
	negative	100%	100%	100%	
Chi- square with GRU-LSTM	positive	100%	99.8%	99.9%	99.9%
	negative	99.9%	100%	99.9%	
PCA with LSTM	positive	99.6%	99%	99.3%	99.6%
	negative	99.5%	99.8%	99.7%	
PCA with Bi-LSTM	positive	99.8%	99.6%	99.7%	99.8%
	negative	99.8%	99.9%	99.8%	
PCA with GRU	positive	98.7%	100%	99.3%	99.6%
	negative	100%	99.4%	99.7%	



Bi-LSTM before feature selection



Bi-LSTM with chi-square



Bi-LSTM with PCA

Figure 4.1. The best accuracy of the models before and after of the feature selection (PCA and chi-square) on YELP Dataset

Cheng et al. [27] used the CNN multi-channel network as a feature extractor and Bi-GRU as a learning model in the YELP dataset, but their proposed method's accuracy also reached 92.9% at the best situation. In contrast, the proposed approach combining the GRU model with the chi-square feature selection method and feature selection based on principal component analysis (PCA) in the YELP dataset was 99.9% and 99.6%, respectively.

Figure 4.1., shows that we have overfitting due to a large number of features in the YELP dataset. This problem has been solved with the feature selection methods, and the accuracy of the model has been increased.

4.3. Evaluation of the Results in the US Airline Dataset

The evaluation of the results of LSTM, Bi-LSTM and GRU models before and after combination with chi-square feature selection and principal component analysis (PCA) in the US Airline dataset are presented in Table 4.2.

Based on the results obtained in Table 4.2, the prediction accuracy of the LSTM model on the US Airline dataset before combining with feature selection methods was 73.36%, and the accuracy of this model has increased to 93.95% after combining with feature selection based on chi-square correlation method. Also, the prediction accuracy of combining Bi-LSTM and GRU models with chi-square feature selection method is 96.14% and 97.91%, respectively, and the prediction accuracy of combining LSTM, Bi-LSTM and GRU models with feature selection based on principal component analysis (PCA) was 99.61%, 96.24% and 94.80%, respectively. Examining the results obtained from the combined models, we conclude that the combination of the GRU model with the chi-square feature selection method has the best possible result of 97.91% in the US Airline dataset.

Table 4.2. Performance of the proposed model for US Airline dataset.

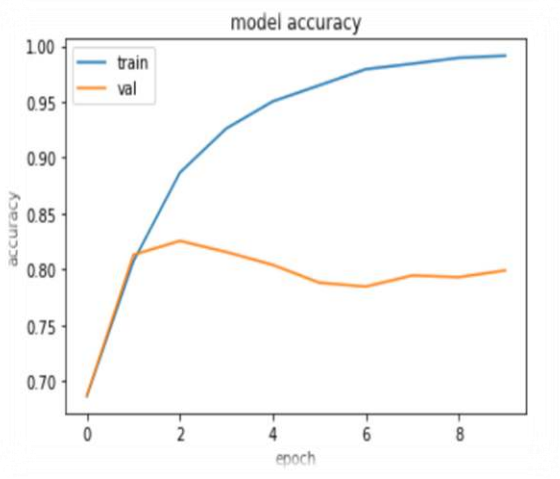
Techniques	Class	Precision	Recall	F1 score	Accuracy
LSTM	positive	83.91%	83.95%	83.93%	73.36%
	negative	55.99%	56.98%	56.48%	
	neutral	70.21%	58.93%	64.08%	
Bi-LSTM	positive	85.52%	83.63%	84.56%	74.28%
	negative	57.9%	57.62%	57.76%	
	neutral	69.02%	63.17%	65.97%	
GRU-LSTM	positive	84.01%	84.6%	84.3%	73.15%
	negative	56.12%	52.97%	54.5%	
	neutral	68.54%	59.82%	63.89%	

Table 4.2. Performance of the proposed model for US Airline dataset (continue).

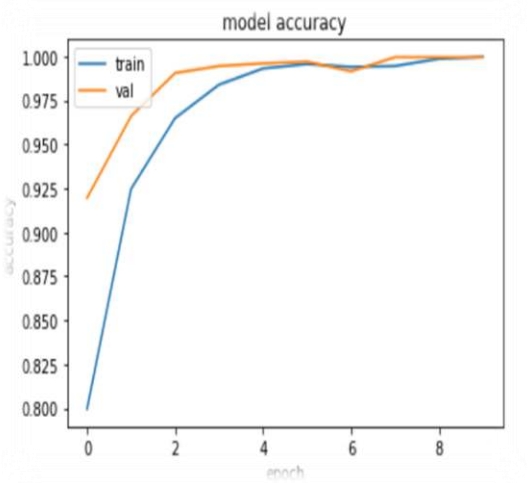
Techniques	Class	Precision	Recall	F1 score	Accuracy
Chi- square with LSTM	positive	97.62%	97.36%	97.49%	93.95%
	negative	90.62%	88.44%	89.52%	
	neutral	92.59%	89.29%	90.91%	
Chi- square with Bi-LSTM	positive	98.59%	97.95%%	98.27%	96.14%
	negative	93.85%	93.1%	93.47%	
	neutral	95.05%	94.2%	94.62%	
Chi- square with GRU-LSTM	positive	99.67%	98.76%	99.22%	97.91%
	negative	96.96%	97.27%	97.12%	
	neutral	96.46%	97.32%	96.89%	
PCA with LSTM	positive	99.13%	98.38%	98.76%	96.61%
	negative	94.86%	94.86%	94.86%	
	neutral	94.88%	95.09%	94.98%	
PCA with Bi-LSTM	positive	99.45%	97.85%	98.64%	96.24%
	negative	90.8%	96.63%	93.62%	
	neutral	97.16%	91.52%	94.25%	
PCA with GRU-LSTM	positive	97.55%	98.82%	98.18%	94.8%
	negative	95.28%	87.48%	91.21%	
	neutral	94.47%	91.52%	92.97%	

Rustam et al. [26] have a voting classifier based on logistic regression and stochastic gradient descent classifier in the US Airline dataset. Soft voting is used to combine LR and SGDC probability and have been evaluated used as performance metrics for accuracy, accuracy, recall, and F1 score. The proposed voting classifier performs better with both feature extraction methods and achieves an accuracy of 0.78.9% and 0.791 with TF and TF-IDF, respectively, at the best situation.

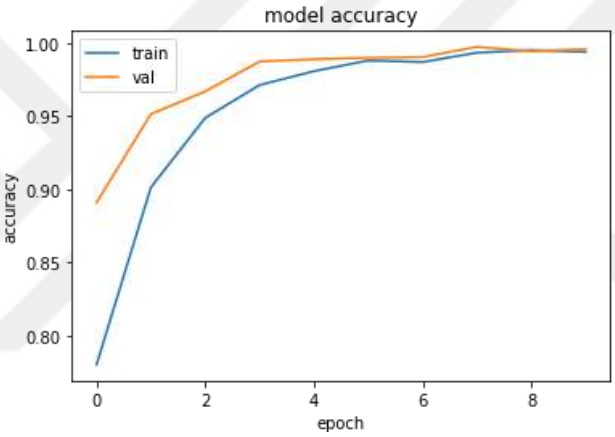
Figure 4.2., shows that we have overfitting due to the large number of features in the US Airline dataset. This problem has solved with feature selection methods, and the accuracy of the model has been increased.



Bi-LSTM with out feature selection



GRU with chi-square



LSTM with PCA

Figure 4.2. The best accuracy of the models before and after of the feature selection (PCA and chi-square) on US Airline Dataset

4.4. Evaluation of the Results in the IMDB Airline Dataset

The evaluation results of LSTM, Bi-LSTM and GRU models before and after combination with chi-square feature selection and principal component analysis (PCA) in IMDB dataset are presented in Table 4.3.

Table 4.3. Performance of the proposed model for IMDB Airline dataset.

Techniques	Class	Precision	Recall	F1 score	Accuracy
LSTM	positive	78.57%	78.5%	78.54%	78.72%
	negative	78.87%	78.93%	78.9%	
Bi-LSTM	positive	78.92%	79.19%	79.05%	79.19%
	negative	79.46%	79.19%	79.32%	

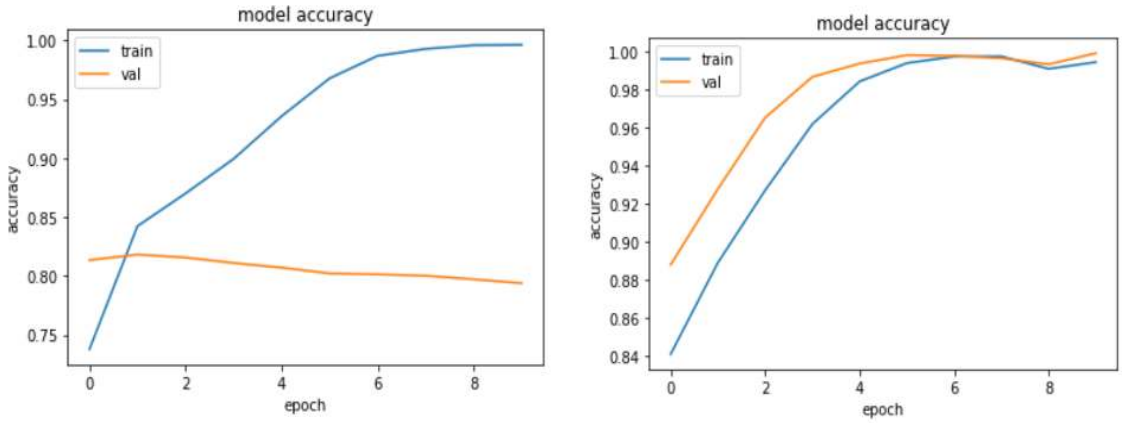
Table 4.3. Performance of the proposed model for IMDB Airline dataset (continue)

Techniques	Class	Precision	Recall	F1 score	Accuracy
GRU-LSTM	positive	77.38%	82.58%	79.89%	79.39%
	negative	81.65%	76.25%	78.86%	
Chi- square with LSTM	positive	99.6%	99.4%	99.5%	99.5%
	negative	99.41%	99.6%	99.5%	
Chi- square with Bi-LSTM	positive	99.9%	99.94%	99.92%	99.92%
	negative	99.94%	99.9%	99.92%	
Chi- square with GRU-LSTM	positive	99.98%	99.78%	99.88%	99.88%
	negative	99.78%	99.98%	99.88%	
PCA with LSTM	positive	98.86%	99.66%	99.26%	99.26%
	negative	99.66%	98.87%	99.26%	
PCA with Bi-LSTM	positive	99.94%	99.58%	99.76%	99.76%
	negative	99.58%	99.94%	99.76%	
PCA with GRU	positive	99.97%	99.89%	99.97%	99.89%
	negative	99.86%	99.81%	99.82%	

Based on the obtained results in Table 4.3., the prediction accuracy of the LSTM model on the IMDB dataset before combining with feature selection methods was 78.72%, but the accuracy of this model after combining with feature selection method based on chi-square correlation has been increased to 99.50%. Also, the prediction accuracy of combining Bi-LSTM and GRU models with chi-square feature selection method is 99.92% and 99.88%, respectively, and the prediction accuracy of combining LSTM, Bi-LSTM and GRU models with feature selection based on the principal component analysis (PCA) was 99.26%, 99.76% and 99.89%, respectively.

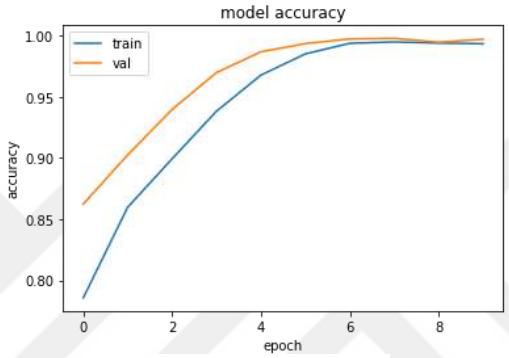
Examining the results obtained from the hybrid models, we conclude that combining the Bi-LSTM model and the chi-square feature selection method with 99.92% had the best result in the IMDB dataset. The evaluation of the results shows that the feature selection had a positive effect on increasing the LSTM models' accuracy. Comparing the results obtained in the IMDB data set with recent studies also shows that the proposed model is highly accurate.

Chen et al. [27] used the CNN multi-channel network as a feature extractor and Bi-GRU as a learning model in the IMDB dataset, but their proposed method's accuracy also reached 91.70% at best. In contrast, the proposed approach combining the GRU model with the chi-square feature selection method and feature selection based on principal component analysis (PCA) in the IMDB dataset was 99.88% and 99.89%, respectively. Figure 4.3., shows that we have overfitting due to the large number of features in the IMDB dataset. This problem is solved using feature selection methods, and the accuracy of the model has been increased.



GRU with out feature selection

Bi-LSTM with chi-square



GRU with PCA

Figure 4.3. The best accuracy of the models before and after of the feature selection (PCA and chi-square) on IMDB dataset

4.5. General evaluation of results

According to the results obtained in the IMDB, YELP and US Airline datasets, we faced the problem of overfitting due to the high number of features. Evaluating the models before the chi-square and PCA feature selection methods solves this problem by selecting highly dependent features on the response and increasing the accuracy of the LSTM, Bi-LSTM and GRU network, show Figure 4.4.

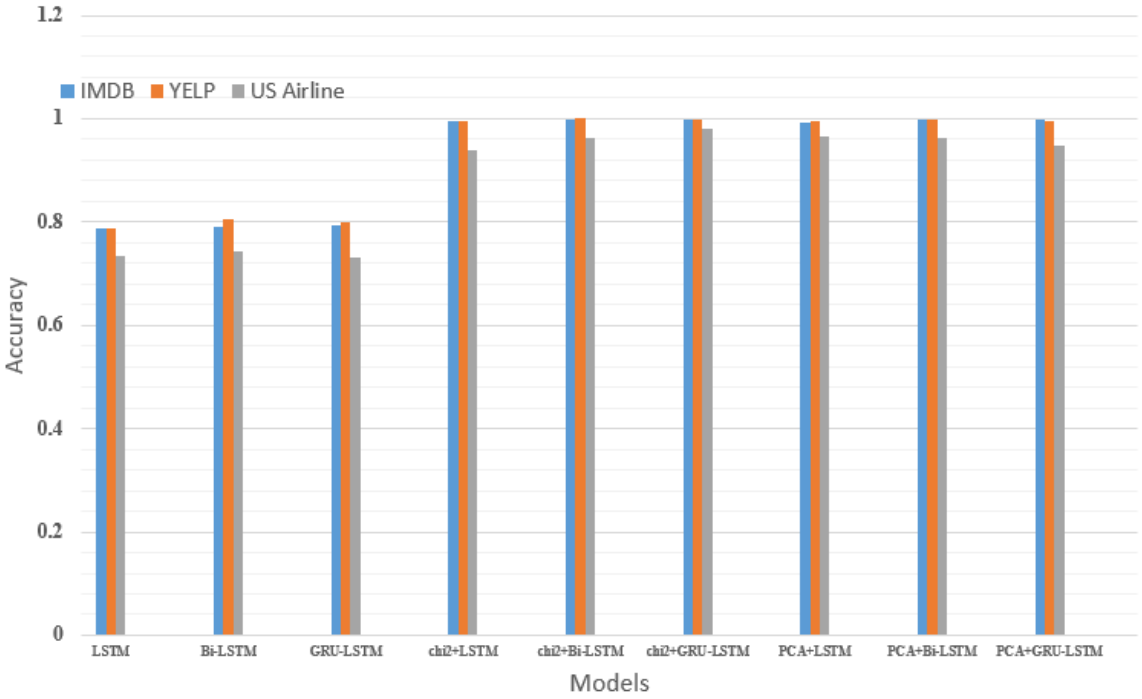


Figure 4.4. The accuracy of models on IMDB, YELP and US Airline datasets

5. CONCLUSION

In this thesis, a classification model was proposed in order to classify sentiment analysis texts into positive and negative based on their polarity. The proposed model uses different feature selection methods, filter-based method Chi-2 and a principal component analysis method (PCA) and three deep learning models. Three different benchmarks dataset has been used:

Eventually, we compared all three classification performance results. The implementation of these three strategies has been described accurately in previous chapters. The results obtained from different classification strategies are compared to show the proposed model's impacts in enhancing the overall accuracy. Conclusions for all three benchmark dataset are summarized as the following:

1. In the Yelp dataset, the accuracy gradually increases as the number of features increase. Starting from 100 features, when using 500 features by chi-square, and select 289 features by PCA, with 512 features, the maximum attained an accuracy of Bi-LSTM is 100% using chi-square, and also the maximum attained an accuracy of GRU-LSTM is 99.8% using PCA. While the highest accuracy without feature selection methods is 78.75%.
2. In the US Airline dataset, the accuracy gradually increases as the number of features increase. Starting from 10 features, when using 20 features by chi-square, and select 12 features by PCA, with 23 features, the maximum achieved accuracy of GRU-LSTM is 97.9% using chi-square, and also the maximum attained an accuracy of LSTM is 96.6% using PCA. While the highest accuracy without feature selection methods is 73.36%.
3. In the IMDB dataset, the accuracy gradually increases as the number of features increase. Starting from 100 features, when using 326 features by chi-square, and select 300 features by PCA with 1125 features, the maximum achieved an accuracy of Bi-LSTM is 99.9% using chi-square, and also the maximum attained accuracy of GRU-LSTM is 99.8% using PCA. While the highest accuracy without feature selection methods is 78.72%.
4. Results show that:
    - The proposed model obtained higher accuracy than three state-of-the-art recurrent approaches;
    - Feature selection and dimensionality reduction methods are more effective in increasing accuracy in sentiment analysis;
    - Implementing the network, an imitation strategy shows that our proposed model is strong for text classification.

RECOMMENDATIONS

The chi-square and principal component analysis (PCA) feature selection methods requires further studies and its comparison with other feature selection methods. It is also recommended to study the combination of feature selection with other LSTM models such as CIFG-LSTM in order to compare it with the methods presented in this study.

REFERENCES

[1] Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. Ain Shams engineering journal, 5(4), 1093-1113.

[2] Akyol, S., & Alatas, B. (2020). Sentiment classification within online social media using whale optimization algorithm and social impact theory based optimization. Physica A: Statistical Mechanics and its Applications, 540, 123094. [1](1)

[3] Yadav, A., & Vishwakarma, D. K. (2020). Sentiment analysis using deep learning architectures: a review. Artificial Intelligence Review, 53(6), 4335-4385.

[4] Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), e1253.

[5] Basiri, M. E., Nemati, S., Abdar, M., Cambria, E., & Acharya, U. R. (2021). ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis. Future Generation Computer Systems, 115, 279-294.

[6] Wang, Y., Huang, M., Zhu, X., & Zhao, L. (2016, November). Attention-based LSTM for aspect-level sentiment classification. In Proceedings of the 2016 conference on empirical methods in natural language processing (pp. 606-615).

[7] Zhou, J., Lu, Y., Dai, H. N., Wang, H., & Xiao, H. (2019). Sentiment analysis of Chinese microblog based on stacked bidirectional LSTM. IEEE Access, 7, 38856-38866.

[8] Lin, Y., Li, J., Yang, L., Xu, K., & Lin, H. (2020). Sentiment analysis with comparison enhanced deep neural network. IEEE Access, 8, 78378-78384.

[9] Salur, M. U., & Aydin, I. (2020). A novel hybrid deep learning model for sentiment classification. IEEE Access, 8, 58080-58093.

[10] Umer, M., Imtiaz, Z., Ullah, S., Mehmood, A., Choi, G. S., & On, B. W. (2020). Fake news stance detection using deep learning architecture (cnn-lstm). IEEE Access, 8, 156695-156706.

[11] Saifee, V., & Jay, T. (2013). Applications and challenges for sentiment analysis: a survey. International Journal of Engineering Research & Technology (IJERT), 2.

[12] Aggarwal, D. (2018). Sentiment Analysis: An insight into Techniques, Application and Challenges. International Journal of Computer Sciences and Engineering, 6(5), 697-703.

[13] Wagh, R., & Punde, P. (2018, March). Survey on sentiment analysis using twitter dataset. In 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA) (pp. 208-211). IEEE.

[14] Patil, H. P., & Atique, M. (2015, December). Sentiment analysis for social media: a survey. In 2015 2nd International Conference on Information Science and Security (ICISS) (pp. 1-4). IEEE.

[15] Behdenna, S., Barigou, F., & Belalem, G. (2018). Document level sentiment analysis: A survey. EAI Endorsed Transactions on Context-aware Systems and Applications, 4(13).

[16] Abbasi, A., Chen, H., & Salem, A. (2008). Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. ACM Transactions on Information Systems (TOIS), 26(3), 1-34.

[17] Kaur, H., & Mangat, V. (2017, February). A survey of sentiment analysis techniques. In 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) (pp. 921-925). IEEE.

[18] Rane, A., & Kumar, A. (2018, July). Sentiment classification system of Twitter data for US airline service analysis. In 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC) (Vol. 1, pp. 769-773). IEEE.

[19] Zhang, X., & Zheng, X. (2016, July). Comparison of text sentiment analysis based on machine learning. In 2016 15th International Symposium on Parallel and Distributed Computing (ISPDC) (pp. 230-233). IEEE.

[20] Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. J. (2011, June). Sentiment analysis of twitter data. In Proceedings of the workshop on language in social media (LSM 2011) (pp. 30-38).

[21] Kumar, K. S., Desai, J., & Majumdar, J. (2016, December). Opinion mining and sentiment analysis on online customer review. In 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC) (pp. 1-4). IEEE.

[22] Jain, S. K., & Singh, P. (2018, December). Systematic survey on sentiment analysis. In 2018 first international conference on secure cyber computing and communication (ICSCCC) (pp. 561-565). IEEE.

[23] Sharma, S., & Jain, A. (2020). An Empirical Evaluation of Correlation Based Feature Selection for Tweet Sentiment Classification. In Advances in Cybernetics, Cognition, and Machine Learning for Communication Technologies (pp. 199-208). Springer, Singapore.

[24] Rana, S., & Singh, A. (2016, October). Comparative analysis of sentiment orientation using SVM and Naive Bayes techniques. In 2016 2nd International Conference on Next Generation Computing Technologies (NGCT) (pp. 106-111). IEEE.

[25] Paudel, S., Prasad, P. W. C., Alsadoon, A., Islam, M. R., & Elchouemi, A. (2018, July). Feature selection approach for Twitter sentiment analysis and text classification based on Chi-Square and Naïve Bayes. In International Conference on Applications and Techniques in Cyber Security and Intelligence (pp. 281-298). Springer,

[26] Rustam, F., Ashraf, I., Mehmood, A., Ullah, S., & Choi, G. S. (2019). Tweets classification on the base of sentiments for US airline companies. Entropy, 21(11), 1078.

[27] Cheng, Y., Yao, L., Xiang, G., Zhang, G., Tang, T., & Zhong, L. (2020). Text sentiment orientation analysis based on multi-channel CNN and bidirectional GRU with attention mechanism. IEEE Access, 8, 134964-134975.

[28] Wazery, Y. M., Mohammed, H. S., & Houssein, E. H. (2018, December). Twitter sentiment analysis using deep neural network. In 2018 14th International Computer Engineering Conference (ICENCO) (pp. 177-182). IEEE.

[29] Rane, A., & Kumar, A. (2018, July). Sentiment classification system of Twitter data for US airline service analysis. In 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC) (Vol. 1, pp. 769-773). IEEE.

[30] A. Kumar, A. Jaiswal, Particle Swarm Optimized Ensemble Learning for Enhanced Predictive Sentiment Accuracy of Tweets. In: Singh P., Panigrahi B., Suryadevara N., Sharma S., Singh A. (eds) Proceedings of ICETIT 2019. Lecture Notes in Electrical Engineering, vol 605. (2020) , Springer, Cham. https://doi.org/11007/978-3-030-30577-2_56.

[31] Zainuddin, N., Selamat, A., & Ibrahim, R. (2016, August). Twitter feature selection and classification using support vector machine for aspect-based sentiment analysis. In International conference on industrial, engineering and other applications of applied intelligent systems (pp. 269-279). Springer, Cham.

[32] Paudel, S., Prasad, P. W. C., Alsadoon, A., Islam, M. R., & Elchouemi, A. (2018, July). Feature selection approach for Twitter sentiment analysis and text classification based on Chi-Square and

Naïve Bayes. In International Conference on Applications and Techniques in Cyber Security and Intelligence (pp. 281-298). Springer, Cham.

[33] F. Iqbal, J. Maqbool, B. C. M. Fung, R. Batool, A. M. Khaytak, S. Aleem, and P. C. K. Hung, "A Hybrid Framework for Sentiment Analysis Using Genetic Algorithm Based Feature Reduction,". IEEE Access, vol. 7, no. c, pp. 14637–14652, 2019.

[34] Y. Zhai, W. Song, X. Liu, L. Liu, and X. Zhao, "A Chi-square Statistics Based Feature Selection," 2018 IEEE 9th Int. Conf. Softw. Eng. Serv. Sci., pp. 160–163, 2018.

[35] N. Iqbal, A. M. Chowdhury, and T. Ahsan, "Enhancing the Performance of Sentiment Analysis by Using Different Feature Combinations," . Int. Conf. Comput. Commun. Chem. Mater. Electron. Eng. IC4ME2 2018, pp. 1–4, 2018.

[36] M. Ahmad and S. Aftab, "Analyzing the Performance of SVM for Polarity Detection with Different Datasets,". Int. J. Mod. Educ. Comput. Sci., vol. 9, no. 10, pp. 29–36, 2017.

[37] M. H. Hama, S. A. Mahmood, "Sentimental analaysis of big data using naive bayes and neural network," . no. May, pp. 25–29, 2015.

[38] What is Python? Executive Summary. Available on: https://www.python.org/doc/essays/blurb/?fbclid=IwAR09Saz904XrAHjy7eGF3paceUvzh0ZclXlOG61ufbe6ELYVxbeguVkdm0.

[39] E. Loper and S.Bird, "NLTK: the Natural Language Toolkit,". CoRR.cs.CL/0205028. 10.3115/1118108.1118117, 2002.

[40] Introducing the Natural Language Toolkit (NLTK). Available on: https://code.tutsplus.com/tutorials/introducingthe-natural-language-toolkit-nltk–cms-28620?fbclid=IwAR1grtEs_Ihbgf7_dmdutzd_2pISOjEx11f9z040 Sq9E9Xhn3kECk6GUgWQ/.

[41] Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013) Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[42] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S. and Dean, J. (2013) Distributed representations ofwords and phrases and their compositionality. In Advances in Neural Information Processing Systems, pp. 3111–3119. arXiv:1310.4546v1.

[43] Pennington, J., Socher, R. and Manning, C.D. (2014) Glove: global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural LanguageProcessing (EMNLP), pp. 1532–1543. Association for Computational Linguistics, Doha.

[44] Xu, S., Liang, H., & Baldwin, T. (2016, June). Unimelb at semeval-2016 tasks 4a and 4b: An ensemble of neural networks and a word2vec based model for sentiment classification. In Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016) (pp. 183-189).

[45] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in ICML, 1997.

[46] S. M. H. Dadgar, M. S. Araghi, and M. M. Farahani, "A novel text mining approach based on tf-idf and support vector machine for news classification," in 2016 IEEE International Conference on Engineering and Technology (ICETECH). IEEE, 2016, pp. 112 116.

[47] Özyurt, F. (2020). A fused CNN model for WBC detection with MRMR feature selection and extreme learning machine. Soft Computing, 24(11), 8163-8172.

[48] Umer, M., Imtiaz, Z., Ullah, S., Mehmood, A., Choi, G. S., & On, B. W. (2020). Fake news stance detection using deep learning architecture (cnn-lstm). IEEE Access, 8, 156695-156706.

[49] Jamal, N., Xianqiao, C., & Aldabbas, H. (2019). Deep learning-based sentimental analysis for large-scale imbalanced twitter data. Future Internet, 11(9), 190.

[50] Abdalla, G., & Özyurt, F. (2020). Sentiment Analysis of Fast Food Companies With Deep Learning Models. The Computer Journal.

[51] Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6(02), 107-116.

[52] Basiri, M. E., Nemati, S., Abdar, M., Cambria, E., & Acharya, U. R. (2021). ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis. Future Generation Computer Systems, 115, 279-294.

[53] Patel, P., Patel, D., & Naik, C. (2021). Sentiment Analysis on Movie Review Using Deep Learning RNN Method. In Intelligent Data Engineering and Analytics (pp. 155-163). Springer, Singapore.

[54] Dang, N. C., Moreno-García, M. N., & De la Prieta, F. (2020). Sentiment analysis based on deep learning: A comparative study. Electronics, 9(3), 483.

[55] A. Kursat Uysal ; Y. L. Murphey , Sentiment Classification: Feature Selection Based Approaches Versus Deep Learning, IEEE International Conference on Computer and Information Technology (CIT),2017

[56] Jelodar, H., Wang, Y., Orji, R., & Huang, S. (2020). Deep sentiment classification and topic discovery on novel coronavirus or covid-19 online discussions: Nlp using lstm recurrent neural network approach. IEEE Journal of Biomedical and Health Informatics, 24(10), 2733-2742.

[57] Long, F., Zhou, K., & Ou, W. (2019). Sentiment analysis of text based on bidirectional LSTM with multi-head attention. IEEE Access, 7, 141960-141969.

[58] S. Kumar, K. De, P. Pratim Roy, Movie Recommendation System Using Sentiment Analysis From Microblogging Data, IEEE Transactions on Computational Social Systems (2020).

[59] Hameed, Z., & Garcia-Zapirain, B. (2020). Sentiment classification using a single-layered BiLSTM model. Ieee Access, 8, 73992-74001.

[60] Ni, R., & Cao, H. (2020, July). Sentiment Analysis based on GloVe and LSTM-GRU. In 2020 39th Chinese Control Conference (CCC) (pp. 7492-7497). IEEE.

[61] Seo, S., Kim, C., Kim, H., Mo, K., & Kang, P. (2020). Comparative study of deep learning-based sentiment classification. IEEE Access, 8, 6861-6875.

[62] Krishnaveni, N., & Radha, V. (2019). Feature selection algorithms for data mining classification: a survey. Indian J Sci Technol, 12(6).

[63] Rao, G., Huang, W., Feng, Z., & Cong, Q. (2018). LSTM with sentence representations for document-level sentiment classification. Neurocomputing, 308, 49-57.

[64] Sharma, A. K., Chaurasia, S., & Srivastava, D. K. (2020). Sentimental Short Sentences Classification by Using CNN Deep Learning Model with Fine Tuned Word2Vec. Procedia Computer Science, 167, 1139-1147.

[65] Mohit. Yelp-dataset.” Kaggle, 22 March. 2018, https://www.kaggle.com/mohit473/yelp-data-set.

[66] Eight, Figure. “Twitter US Airline Sentiment.” Kaggle, 16 Oct. 2019, www.kaggle.com/crowdflower/twitter-airline-sentiment.

[67] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Ng, and C. Potts, ``Learning word vectors for sentiment analysis," Assoc. Comput. Linguistics (ACL), Hum. Lang. Technol., OR, USA, 2011, pp. 142_150. https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

[68] Mahmood, S. A., & Qasim, Q. Q. (2020). Big Data Sentimental Analysis Using Document to Vector and Optimized Support Vector Machine. UHD Journal of Science and Technology, 4(1), 18-28.

[69] Severyn, A., & Moschitti, A. (2015, August). Twitter sentiment analysis with deep convolutional neural networks. In Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval (pp. 959-962).

[70] Wen, S., Wei, H., Yang, Y., Guo, Z., Zeng, Z., Huang, T., & Chen, Y. (2019). Memristive LSTM network for sentiment analysis. IEEE Transactions on Systems, Man, and Cybernetics: Systems.

[71] Ali, N. M., Abd El Hamid, M. M., & Youssif, A. (2019). Sentiment analysis for movies reviews dataset using deep learning models. International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol, 9.

[72] Allen, J. J., Chambers, A. S., & Towers, D. N. (2007). The many metrics of cardiac chronotropy: A pragmatic primer and a brief comparison of metrics. Biological psychology, 74(2), 243-262.

CURRICULUM VITAE

Mohammed Hussein ABDALLA

[redacted]

[redacted]

[redacted]

RESEARCH EXPERIENCES

- ✓ Computer Programming languages available (C-C++, MATLAP, Python.)
- ✓ Computer programs you can use (SPSS, Microsoft Office, Photoshop.)

WORK EXPERIENCE

[redacted]

ACADEMIC ACTIVITIES

Paper:

Abdalla, M. H., & Karabatak, M. (2020, June). To Review and Compare Evolutionary Algorithms in Optimization of Distributed Database Query. In 2020 8th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-5). IEEE.

Abdalla, M. H., Özyurt, F. (2021, February). Improving Sentiment Analysis Based on Gated Recurrent Unit Model by Using Feature Selection. In 1st International Conference on Computing and Machine Intelligence (ICMI 2021) Istanbul, Turkey (ISDFS) (pp. 245-249).