Republic of Iraq

Ministry of Higher Education and Scientific Research

University of Anbar

College of Computer Science and Information Technology
Department of Computer Science

Sentiment Analysis Model Based on Multi-Layer Perceptron for Social Networks

A Thesis
Submitted to the Department of Computer Science - College of Computer Science and Information Technology, University of Anbar as a Partial Fulfillment of the Requirements for Master Degree of Science in Computer Science

By

Anmar Hameed Afair

Supervised By

Assist. Prof. Dr. Khalid Shaker Jasim

1440A.H. *2019 A.D.*

بسم الله الرحمن الرحيم

(يُؤْتِي الحِكْمَةَ مَن يَشَاءُ وَمَن يُؤْتَ الحِكْمَةَ فَقَدْ أُوتِيَ خَيْرًا كَثِيرًا
وَمَا يَذَّكَّرُ إِلاَّ أُوْلُوا الأَلْبَابِ)

صدق الله العظيم

(البقرة: ٢٦٩)

اسم الطالب: أنمار حميد عفر عماش

الكلية: كلية الحاسوب وتكنولوجيا المعلومات - قسم علوم الحاسبات

عنوان الرسالة:

Sentiment Analysis Model Based on Multi-Layer Perceptron for Social Networks

طبقا لقانون حماية حق المؤلف رقم ٣ لسنة ١٩٧١ المعدل العراقي فأن للمؤلف حق منع اي حذف او تغيير للرسالة او الاطروحة بعد اقرارها و هي الحقوق الخاصة بالمؤلف وحده والتي لا يجوز الاعتداء عليها. فلا يحق لأحد ان يقرر نشر مصنف احجم مؤلفه عن نشره او اعادة نشر مؤلف لم يقر مؤلفه بذلك، فأذا قام بذلك اعتبر عمله غير مشروع لانه استعمل سلطة لا يملكها قانونا.

Supervisor's Certification

*I certify that I read this thesis entitled "**Sentiment Analysis Model Based on Multi-Layer Perceptron for Social Networks**" that was carried out under my supervision at the Department of Computer Science of the University of Anbar, by the student "**Anmar hameed afair**" and that in my opinion it meets the standard of a thesis for the degree of Master of Science in Computer Science.*

Signature:
Name: Assist. Prof. Dr. Khalid Shaker Jasim
Date: /   /

Linguist Certificate

I certify that, I read this thesis entitled " ***Sentiment Analysis Model Based on Multi-Layer Perceptron for Social Networks****" and I found it linguistically adequate.*

Signature:
Name: Dr. Sadir A. Fadhil (Linguist Authority)
Date: / /

Certification of the Examination Committee

*We the examination committee certify that we have read this thesis entitled " **Sentiment Analysis Model Based on Multi-Layer Perceptron for Social Networks"** " and have examined the student " **Anmar Hameed Afair** ", in its contents and what is related to it, and that in our option it is adequate to fulfill the requirements for the degree of **Master of Computer Science**.*

Signature:
Name: **Prof. Dr. Belal Al-Khateeb** **(Chairman)**
Date: / / 2020

Signature:
Name: **Assist. Prof. Dr. Boshra F. zopon Al_bayaty** **(Member)**
Date: / / 2020

Signature:
Name: **Assist. Prof. Dr. Esam Taha Yassen** **(Member)**
Date: / / 2020

Signature:
Name: **Assist. Prof. Dr. Khalid Shaker Jasim** **(Supervisor)**
Date: / / 2020

Approved by the Dean of the College of Computer Science and Information Technology, University of Anbar.

Signature:
Name: **Assist. Prof. Dr. Salah Awad Salman**
Title: **Dean of the College**
Date: / / 2020

Acknowledgements

First of all, I would like to thank and praise Allah, Lord of the world who helped me to complete this thesis.

I would like to express my appreciation and gratitude to my Supervisor "Assist. Prof. Dr. Khalid Shaker Jasim" for his guidance, helpful suggestions and continuous support me during the period of my study.

I wish to thank the staff of Computer Science Collage at University of Anbar for their help and support.

I would like to extend my thanks and deep appreciation to my family who has sacrificed so much in order to achieve my goal.

I would like to thank my friends for their support me.

Researcher

Dedication

I would like to dedicate this work:

To my candles that light my life

My father and My Mother

To my second half, my wife for her patience and continued support

To my heart and my hope in life … My daughter "Hawraa"

To My Brothers and My sisters

To All My friends

Abstract

Social media and other online platforms contain a large amount of data in the form of text, audio, video and image. Sentiment analysis (SA) has become a field of computational studies. In general, SA deals with the mining of information related to sentiments. Therefore, SA is necessary in texts in the form of messages or posts to determine whether a sentiment is negative or positive. SA is a focused on the extraction of emotions and opinions of people about products, events, movies, videos and music from structured, semi-structured or unstructured textual data. SA is an interesting topic in the field of research and technology. It combines natural language processing techniques with data mining approaches for building systems. The main problems that exist in the current techniques are: inability to perform well in different domains, inadequate accuracy and performance in sentiment analysis based on insufficient labelled data, incapability to deal with complex sentences that require more than sentiment words and simple analysing. Therefore, an approach that can classify sentiments into two classes, namely, positive sentiment and negative sentiment is proposed. A multilayer perceptron (MLP) classifier has been used in this document classification system. The aim of the present research is to provide an effective approach to improving the accuracy and speed of SA systems. This objective is achieved via four main steps. The **first step** is pre-processing aimed at noise removal or data filtering. It also involves prep-processing linguistic data using natural language processing (NLP) techniques. During this process, the input dataset is filtered and processed to provide highly accurate data, reduce the dataset size and shorten the processing time. The **second step** is applying feature extraction using the term frequency–inverse document frequency (TF-IDF) technique for dimensionality reduction by which an initial set of raw data is reduced to easily manageable groups for processing. The **third step** is applying feature selection using the chi-square method to reduce the features of documents and thereby shorten the processing time and improve system accuracy. In the **final step,** the special structures of the multilayer perceptron (MLP) classifier are designed to determine whether a sentiment is positive or negative. The proposed approach is applied to and tested on two datasets, namely, a Twitter dataset and a movie review dataset; the accuracies achieved reach 0.85% and 0.99% respectively

Table of Contents

Chapter One: General Introduction

Chapter Two: Theoretical Background

Chapter Three: The proposed System

Chapter Four: Results and Discussion

Chapter Five: Conclusions and Future Work

List of Tables

List of Figures

List of Algorithms

List of Abbreviations

Abbreviation	Meaning
AI	Artificial Intelligence
ANN	Artificial Neural Networks
API	Application Program Interface
BP	Backpropagation
CNN	Convolutional Neural Networks
FN	False Negatives
FP	False Positives
IDF	Inverse Document Frequency
IMDB	Internet Movie Database
IOT	Internet of Things
IR	Information Retrieval
LSA	Latent Semantic Analysis
LSTM	Long Short-Term Memory
ME	Maximum Entropy
ML	Machine Learning
MLP	Multilayer Perceptron
NB	Naive Bayes
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
POS	Parts-Of-Speech
SA	Sentiment Analysis
SGD	Stochastic Gradient Descent
SNS	Social Network Service
SVM	Support Vector Machine
TF	Term Frequency
TF-IDF	Term Frequency – Inverse Document Frequency
TN	True Negatives
TP	True Positives

Chapter One

Introduction

1.1 Introduction

Social media sites are popular destinations in the online world. Millions of users visit social networking sites, such as Facebook, YouTube and Twitter. Online social media represent a fundamental shift in how information is produced, transferred and consumed. With the explosive growth of social media (i.e., reviews, forum discussions, blogs and social networks) on the Web, individuals and organisations increasingly use public opinion for their decision making [1,2].

Public and private opinion about a wide variety of subjects is expressed and spread continually via numerous social media platforms. Millions of people use social network sites to express their emotions, share their opinion and disclose details about their daily lives. However, people write anything, such as social activities or any comment on products, through online communities and provide an interactive forum where consumers inform and influence others. Moreover, social media provide an opportunity for businesses to connect with their customers by enabling them to advertise or speak directly to customers about their products and services [3,4].

Sentiment analysis (SA) or opinion mining is the computational study of people's opinions, appraisals, attitudes and emotions towards entities, individuals, issues, topics, events and their attributes using information retrieval and computational linguistics. It is used to identify positive, negative or neutral opinions, emotions and evaluations. The task is technically challenging and practically useful. For example, businesses always want to find public or consumer opinions about their products and services. Potential customers also want to know the opinions of existing users before they use a service or purchase a product [4,5].

The inception and rapid growth of the field coincide with those of social media on the Web. Since early 2000, SA has grown to be one of the most active research areas in natural language processing (NLP), which uses advanced techniques for machine learning, information retrieval, Web mining and information extraction [6].

SA has spread from computer science to management sciences and social sciences, such as marketing, finance, political science, communications, health science and even history, due to its importance to business and society in general. This proliferation is due to the fact that opinions are central to almost all human activities and are key influencers of people. It can be believed the reality and preferences are, to a considerable degree, conditioned upon how others see and evaluate the world. For this reason, whenever individuals or organisations make decisions, they often seek out the opinions of others [5].

At present, users who intend to purchase products are no longer limited to the opinions of their friends and family because many user reviews and discussions about consumer products are available in Web-based public social networks. However, finding and monitoring opinion sites on the Web and distilling the available information remain a difficult task because of the proliferation of diverse sites. Each site typically contains a large volume of texts, and users cannot easily identify relevant sites to extract and summarise the opinions in them. Thus, automated SA systems are required. Moreover, many start-ups focus on providing SA services. Many large corporations have also built their own in-house capabilities. These practical applications and industrial interests provide strong motivations for SA research [7,8].

Existing research has produced numerous techniques for various tasks of SA, including supervised and unsupervised methods. In the supervised setting, early research used all types of supervised machine learning methods, such as neural networks, support vector machine (SVM), maximum entropy and naïve Bayes (NB). Unsupervised methods include various approaches that exploit sentiment lexicons, grammatical analysis and syntactic patterns [9,10].

The main problems that exist in the current researches and techniques on sentiment analysis are: inability to perform well in different domains, inadequate accuracy and performance in sentiment analysis based on insufficient labelled data, collecting and preprocessing data, incapability to deal with complex sentences and abbreviation that require more than sentiment words and simple analysing [5,6,9].

1.2 Twitter and Movie Reviews Sentiment Analysis

Twitter is a well-liked and currently the fastest growing microblogging service. Twitter permits its users to post standing updates or ‘tweets’ to a network of followers and exploits various communication services (e.g., cell phones, emails, Internet interfaces or different third-party services). Twitter SA is trickier than broad SA because of the slang words, misspellings and repeated characters. Although some users consider the restriction of 140 characters a severe limitation, others argue that this feature distinguishes Twitter from other microblogging services and that the short data are straightforward to consume and quick to unfold. Although tweets are restricted in size, individuals worldwide post on Twitter many times each day, with their content varying greatly depending on their interests and behaviours; such content contains extensive information, ranging from lifestyle stories to recent local and foreign news, events and products; and reflects users’ political and religious beliefs and opinions[11,12,13].

Twitter has become one massive corpus of current information and opinion. However, the accepted fact is that most tweets are pointless babbles, conversational in nature or spam. The sentiment can be found in the comments or tweets to provide useful indicators for many different purposes [11,14].

One domain of reviews is that of movie reviews, which affect writers, users, film critics and production companies. The reviews obtained from movie review sites can be used as a reference for movie fans to identify recommended movies and as a medium for movie producers to gauge the public’s response to the movies released [15,16] .

The movie reviews posted on websites are informal and lack structured grammar. The opinions expressed in movie reviews genuinely reflect the emotions being conveyed. The great use of sentiment words to provide movie reviews inspired us to derive an approach to classify the polarities of movies using such sentiment words. SA is the process of computationally identifying and categorising opinions expressed in a text, especially to determine whether the writer’s attitude towards a particular topic, such as a product, is positive, negative or neutral [17].

With available technologies, we can determine whether a movie is good or bad from the point of view of users and identify the reasons why existing opinions are positive or negative. Most SA studies are presently focused on social media sources, such as Internet Movie Database (IMDb), Twitter and Facebook, which require approaches to be tailored to serve the increasing demand for opinions in the form of text [19,18].

1.3 Motivation

Social media platforms, such as blogs and asocial networking sites, content communities and virtual worlds, have become one of the most powerful sources for news, markets and industries. They are a wide platform full of users’ thoughts, emotions, reviews and feedback, all of which can be used in many aspects.

Twitter has been selected as the study object because it is a better approximation of public sentiment as opposed to conventional internet articles and web blogs. The reason is that it contains limited characters to present one’s idea in a concise and in an effective manner. Moreover, the response on Twitter is prompt and general because the number of users who tweet substantially exceeds the number of those who write web blogs on a daily basis.

In addition, we select movie reviews from the Internet Movie Database (IMDb) because opinions expressed in movie reviews provide a genuine reflection of positive or negative emotions towards movies. Movie reviews, unlike Twitter, have an unlimited text length, and they are annotated by human users using their sentiments towards the movies being described. Both sets are used in this thesis to highlight the difficulties of guessing sentiments in short and often ungrammatical English texts as in Twitter as opposed to long and grammatically correct English texts used in movie reviews.

1.4 Related works

In this era, information sharing through social media has increased, and most users actively share their personal ideas and information publicly. This information for an analyst or researcher is a gold mine that could serve as a source of valuable information for strategic decision making. Thus, SA has attracted growing interest. However, several factors of SA have yet to be substantiated by research. This deficiency should push the development of the research area in various ways, including the augmentation of machine learning methods in NLP and information retrieval, enhancement of the World Wide Web to provide training datasets for machine learning algorithms and the realisation of commercial and intelligent applications that the area provides [9,8,5].

A large number of research works focus on sentiment classification with different goals and using different supervised and unsupervised methods. They are generally based on machine learning models.

- **Shoushan Li et al.,** in 2010, mainly focused on the automatic SA method for classifying tweets. They suggested that sentiment classification is used to classify a text according to its sentimental polarities, such as favourable or unfavourable. Sentiment classification achieves superior performance. For sentiment classification, all classifiers including the polarity shifting detector, three base classifiers and the meta-classifier in stacking are trained by SVM using the SVM-light tool 3 with Logistic Regression method for probability measuring. It comprises four levels, namely, word level, phrase level, sentence level and document level. The two types of document level in sentiment classification are term counting (lexicon-based) and machine learning approaches (corpus-based) [20].

In the same year, **Pak and Paroubek** proposed a model to classify tweets as positive and negative. They created a twitter corpus by collecting tweets using Twitter API and automatically annotating those tweets using emoticons. Using this corpus, they developed a sentiment classifier based on the multinomial NB method that uses features such as N-gram and parts-of-speech (POS) tags.

However, the training set they used was not highly efficient because it contained only tweets with emoticons [21].

- **Agarwal et al.,** in 2011, developed a three-way model for classifying sentiments into positive, negative and neutral classes. They experimented with models such as a unigram model, feature-based model and tree kernel-based model. The tree kernel-based model represents tweets as a tree. The feature-based model uses 100 features, and the unigram model uses over 10,000 features. They concluded that the features which combine the prior polarity of words with their POS tags are the most important in classification tasks. The tree kernel-based model outperforms the other two models [22].

- **Moraes et al.**, in 2012, proposed to automate the task of classifying single-topic textual reviews by using document-level sentiment classification in expressing positive or negative sentiments. In general, supervised methods consist of two stages, namely, extraction/selection of informative features and classification of reviews by using learning models, such as SVM and NB. SVM has been extensively and successfully used as a sentiment learning approach, whereas ANNs have rarely been considered in comparative studies in the SA literature. This work presented an empirical comparison between SVM and ANN in terms of their use in document-level SA. The experiments indicated that the results of ANN are superior or at least comparable results to those of SVM [23].

- **Singh et al.,** in 2013, evaluated and compared the performances of three approaches, namely, two machine learning-based classifiers (NB and SVM), the unsupervised semantic orientation approach (SO-PMI-IR algorithm) and the SentiWordNet approaches for sentiment classification on movie reviews. The results showed that the accuracy of classification by NB is marginally better than of SVM and is close to that of the SO-PMI-IR algorithm. The SO-PMI-IR algorithm obtains impressive accuracy levels and seems to be the best option due to its unsupervised nature. The authors concluded that NB’s performance can be compared with the popularly believed superior performance of SVM, at least for sentiment classification [24].

● **Pablo and Marcos**, in 2014, presented variations of NB classifiers for detecting the polarity of English tweets. Two different variants of NB classifiers were built: baseline (trained to classify tweets as positive, negative and neutral) and binary (uses a polarity lexicon and classifies tweets as positive and negative; neutral tweets are neglected). The features considered by these classifiers were lemmas (nouns, verbs, adjectives and adverbs), polarity lexicons and multiple words from different sources and valence shifters. The experiments showed that the best performance is achieved by using a binary classifier between two sharp polarity categories, namely, positive and negative. To detect tweets with and without polarity, the system uses a basic rule that searches for polarity words within analysed tweets/texts. When the classifier is provided with a polarity lexicon and multiword, it achieves a 63% F-score [25].

In the same year, **Suchdev et al.** analysed people's sentiments in their tweets and found that these sentiments pushes companies to bring up profits from their products by increasing the quality and the recommendations given by customers' tweets. They discussed a sentiment summarisation system which takes documents for analysis as input and generates a detailed document summarising the opinions in the input documents [26].

● **Łukasz et al.**, in 2015, proposed a new method called 'frequentiment' which robotically evaluates sentiments (opinions) of users from an Amazon review data set. They then extended the study by developing a dictionary of words on the basis of the calculation of the probabilistic frequency of words present in the text and the evaluation of the influence of polarity scored by separating the features present in the text. They analysed the outcomes produced by unigram, bigram and trigram lexicon using lexicon-based supervised and unsupervised machine learning approaches. They also compared 37 machine learning methods (lexicons, ensembles with lexicon predictions as input and supervised learners) to evaluate the results of the analysis of the Amazon dataset [27].

In the same year, **Duncan and Zhang** used a dataset gathered from Twitter API for SA. The authors used a feedforward neural network to classify a Twitter dataset into positive or negative tweets. The accuracy obtained using the feedforward neural network was 74.15% [28].

In the same year, **Tripathy et al.** attempted to classify SA for movie reviews using machine learning techniques. Two different algorithms, namely, NB and SVM, were implemented. These algorithms were used to classify sentimental reviews as either positive or negative. They observed that the SVM classifier outperformed the NB classifier in predicting the sentiment of a review [29].

● **Asghar et al.**, in 2016, studied the development of a lexical resource for health-related sentiment classification using supervised and unsupervised learning algorithms. The health-related reviews were classified as either positive or negative on the basis of the developed lexicon. The proposed approach is based on a dataset of health reviews and corpus-based sentiment detection and scoring. In each iteration, the vocabulary of the lexicon is updated automatically from an initial seed cache, irrelevant words are filtered, words are declared as medical or nonmedical entries, and the sentiment class and score are finally assigned to each word. The results obtained demonstrate the efficacy of the proposed technique. The lexicon uses the principles of probability theory and the quantum of domain-specific words [30].

In the same year, **Indriani et al.** obtained data by using the Twitter API. The authors compared the performances of NB smoothing methods in improving the SA of tweets [31].

● **Zhou et al.**, in 2017, used a set of several traditional NLP features, domain-specific features and word embedding features from tweets and their metadata and adopted supervised machine learning algorithms to perform sentiment polarity classification (positive and negative). Specifically, tweet metadata and user metadata were used in the domain to improve the results. The system performance ranks above average [32].

In the same year, **Ramadhani and Goo** used the deep feedforward neural network and MLP to classify sentiments in a Twitter dataset. The accuracy obtained by using the deep feedforward neural network was 75.03% and that obtained by using MLP was 67.45% [6].

In the same year, **Baid et al.** analysed movie reviews using various techniques, such as NB, K-nearest neighbour and random forest. Data were collected from 1000 positive and 1000 negative user-created movie reviews archived on the IMDb. The best results were obtained by the NB classifier, followed by those obtained by the random forest classifier and the K-nearest neighbour classifier [17].

● **Han et al.**, in 2018, proposed a novel model based on a three-layer neural network model to capture context information for the sentiment classification task. The model builds a hybrid neural network model using convolutional neural networks (CNNs) and long short-term memory (LSTM) for word context extraction and document representation, respectively. The authors used the local semantic information of a word context to improve sentence representation. They also provided a comparative analysis of the results for three different datasets. The experiment results showed that the model achieves significant and consistent improvements relative to other state-of-the-art methods [33].

In the same year, **Elbagir et al.** used machine learning algorithms and scikit-learn in the SA of Twitter data. The Twitter datasets made publicly available by NLTK Corpora were used, and an efficient feature was created by using a feature extraction technique. The authors trained and tested various machine learning classifiers, such as MultinomialNB, BernoulliNB, LogisticRegression, SGD classifier, SVC, LinearSVC, and NuSVC. Experimental results demonstrate that BernoulliNB, LogisticRegression, and SGD classifier achieved an accuracy of as high as 75% [34].

In the same year, **Karakuş et al.** performed sentiment classification on Turkish movie reviews collected from www.beyazperde.com to train and test deep learning models. The entire dataset consists of 44,617 samples, including positive and negative reviews. In the study, several deep learning models were used for a binary sentiment classification problem. CNN, LSTM, CNN–LSTM and MLP were employed, and CNN–LSTM achieved the best results and performance [35].

● **Murthy et al.**, in 2019, proposed and introduced a distributed real-time Twitter SA and visualisation framework by implementing a novel algorithm for Twitter SA called Emotion–Polarity–SentiWordNet. The framework was applied to build an interactive web application called 'TwitSenti', which can benefit companies and other organisations in identifying people's sentiments (positive and negative) towards aspects such as brands and current events, which, in turn, facilitate decision making and planning of marketing strategies. The algorithm was validated against three existing classifiers, and the results proved that the Emotion–Polarity–SentiWordNet framework provides the highest accuracy of 85%. In addition, the framework showed the best scalability results when evaluated through a web app as four node clusters and proved to be fast and easily scalable with massive data [36].

In the same year, **D. M. Reddy et al.** classified tweets into positive and negative sentiments. However, instead of using traditional methods or preprocessing text data, the authors used the distributed representations of words and sentences to classify the tweets. They used LSTM networks, CNNs and ANN [37].

In the same year, **Ali et al.** implemented an SA classifier for an IMDb dataset of 50K movie reviews using three deep learning networks (MLP, CNN and LSTM) in addition to a hybrid network CNN_LSTM. Word2vector technique was used for word embedding. CNN_LSTM outperformed all other implemented deep learning techniques. A review dataset of English movies was used, and the result showed that the proposed deep learning techniques (MLP, CNN, LSTM and CNN_LSTM) outperformed SVM, NB and RNTN [38].

1.5 Problem statement

People today depend on microblogging sites, such as Twitter, Facebook and Tumblr, to communicate with their relatives and the rest of the world. SA involves determining the opinions and emotions of users related to certain events. The main problems that exist in the current researches and techniques on sentiment analysis are: inability to perform well in different domains, inadequate accuracy and performance in sentiment analysis based on insufficient labelled data, collecting and pre-processing data, incapability to deal with complex sentences and abbreviation that require more than sentiment words and simple analysing [12,9,7].

The research on SA is long standing. Today, SA is a major issue in the field of research and technology. With the daily increase in the number of users of social networking websites, large amounts of data are produced in the form of text, audio, videos and images. For example, Twitter generates large volumes of opinion texts in the form of tweets available for SA. The resulting voluminous information hinders the prompt extraction, reading, analysis, summary and organisation of sentences into an understandable format. It also enhances the need to determine whether sentiments are negative or positive [9,7,2]

The work of thesis attempts to classify human sentiments into two categories, namely, positive and negative, to deeply understand human thinking and provide insights which can be used in various ways.

1.6 Objectives

This study mainly aims to design and implement an efficient and highly accurate classification system. The following objectives are outlined to accomplish this aim:

1. To explore the ability of the multilayer perceptron (MLP) classifier on the automatic classification of text into positive or negative sentiment.

2. To investigate the best number of hidden layers and nodes in artificial neural networks (ANNs) on handling sentiment analysis problems.

3. To assess the effect of chi-square in increasing the ability of feature selection process.

1.7 Thesis Overview

This thesis comprises five chapters. A brief overview of the chapters is as follows:

Chapter 1: A general introduction about SA is presented. The research motivation, problem statements, objectives of the study and several related works and arrangement are also discussed.

Chapter 2: The theoretical background of the thesis is presented.

Chapter 3: The proposed system, the design and implementation steps of the proposed system and its algorithms are introduced, along with an explanation of each technique.

Chapter 4: In the 'Results and Discussion' section, the test results of implementing the proposed system in Chapter 3 are presented, and the performance evaluation of the system is discussed.

Chapter 5: The 'Conclusions and Future Work' section provides the conclusions of the thesis and presents recommendations for future work.

Chapter Two

Theoretical Background

2.1 Introduction

With the rapid growth of online social media and the advancement of web technology, internet users are given access to large volumes of data in the web which can be retrieved using many techniques. The internet has become a platform for online learning, exchanging ideas and sharing opinions. Social networking sites, such as Twitter, Facebook, Google and YouTube, have rapidly gained popularity because they allow people to share and express their views about topics, discuss with different communities or post messages across the world. Mining is used to extract valuable information from large amounts of data [39,7].

SA focuses on the analysis and understanding of emotions from text patterns. It identifies the opinion or attitude that a person has towards a topic or an object, and it seeks to identify the viewpoint underlying a text span. SA is useful in social media monitoring to automatically characterise the overall feeling or the mood of consumers, as reflected in social media, towards a specific brand or company and determine whether they are viewed positively or negatively on the web. For automating the task of classifying a single-topic textual review, document-level sentiment classification is used in expressing positive or negative sentiments [17,40].

This chapter provides a theoretical overview of social networks and presents SA and the main approaches for sentiment extraction. The preprocessing of data, feature extraction and feature selection are also explained. In addition, ANNs and the evaluation of performance using a confusion matrix are detailed.

2.2 Applications of sentiment analysis

SA has many applications in various fields. The following points highlight some of these applications [7,19]:

1- Applications in Marketing: At present, people tend to refer to reviews of products available online prior to making a purchase. For many businesses, online opinion decides the success or failure of their products. Thus, SA plays an important role in businesses. Businesses also wish to extract sentiments from online reviews to improve their products and, in turn, their reputation and boost customer satisfaction.

2 -Applications in Politics: SA offers many valuable uses for political organisations by enriching their understanding of social media sentiments. Social media feedback has been used to inform political leaders of potential threats, problems or issues with their organisations. SA has previously played a role in predicting elections and acquiring citizens' responses on important issues, such as increasing prices and changing the constitution.

3-Applications in Healthcare: Attention towards SA has flourished over the last two decades because of the immense popularity of social media. The phenomenal rise in blogging has been observed in health communities, such as medical forums, which are swamped by millions of users (many of whom are patients) seeking health-related information, sharing medical problems or experiences and opting for informational support or opinions from other users (patients, health professionals or doctors). As these webpages contain users' health-related experiences and medical for use by practitioners and patients, SA tools need to be developed for use in the medical field.

4- Applications in Finance: SA can also be used in the financial world. Investors can easily follow their favourite companies and monitor their sentiment data in real time. With SA, business investors can acquire business news easily and aggregate this information to make sound financial decisions.

5- Applications in Smart Homes: Smart homes are supposed to be the technology of the future. In the future, an entire home could be networked, and users can control any part of such homes using a tablet device. Recently, extensive research has explored the Internet of Things (IOT). SA could likewise be applied to IoT. Like for example, based on the current sentiments or emotions of users, a smart home could alter its ambiance to create a soothing and peaceful environment. SA can also be used in trend prediction.

2.3 Social Networks

From the early 90's until today, the World Wide Web has exerted a significant impact on almost all areas of our everyday life, from the way information is created and transferred to the way in which people communicate with on another. As technology evolves and becomes increasingly affordable, the growing use of the World Wide Web has created numerous possibilities and needs. This environment has led to the transition from the passive consumption of information, or the so-called Web 1.0, to the current interactive Internet, in which users can easily publish their own content and communicate with other users [41,42].

One of the most important features of Web 2.0 is the spread of social media and social networking sites. Certain subgroups of these online social services are designed exclusively for the adoption or preservation of existing social relations among people, whereas others are aimed at facilitating the sharing of files created by users–members (user-generated content sharing sites). The popularity of these services has offered a variety of possibilities and opportunities for users not only at interpersonal and creative levels but also at a professional level. Therefore, the study of these social networks and the interactions of users has attracted interest from the research community [41,43].

2.3.1 Social Network Types

Online social networks are mainly designed for interaction. However, in reality, different types of interactions and communities exist. For example, one may have a community of his/her friends with whom he/she interacts differently from his/her colleagues. Hence, online social networks are designed under different categories, as presented in the next subsection [44,41].

2.3.1.1 Social–Personal Orientation

The first category is the ‘friendship-oriented’ social networks, which are mostly used to communicate with friends, relatives and family members. In these platforms, users tend to share information about their private life which includes demographic (age, ethnicity, gender, relationship and marital status), psychographic (hobbies, interests, feelings and life events), location (places of residence, places visited or places to be visited). Online social network platforms in this category include Facebook , Twitter and other. [46,45,41].

2.3.1.2 Professional–Career Orientation

Social networks under this category are designed to provide easy access to career opportunities. These social networks are generally platforms where professionals can find their future employees and partners. In addition, this platform can be used by job or employee seekers. Professionals can also communicate with people they know from their respective industries (such as classmates, colleagues, bosses and partners) who could be opening a new connection to a new career opportunity. They are considered social networks because users maintain their social graphs which comprise connections with their professional environment. Such platforms support similar functionalities, such as social and friendship-oriented social networks. The most popular social network in this category is LinkedIn [46,44,43].

2.3.1.3 Communities of Interest

Social networks in this category aim to bring together people who share the same interests. These networks were designed on the basis of the hypothesis that people who have the same interests are also interested in interacting together. People who belong in the same community of interest share their ideas about their passion without knowing much about the lives of the other people in the community. Examples of platforms of this category are MySpace, Stack Overflow, Quora and Dogster [46,44,41].

2.3.2 Twitter

Microblogging is one of the most popular forms of online social communication in today’s world. With Twitter, users normally post short textual messages of up to a few hundred characters to mainly update their circle of friends or followers about their daily life, the events around them and their reactions to news and events that they might have witnessed [6].

Twitter is undoubtedly the most popular microblogging site at the present time. It is essentially an online social networking and microblogging service that allows its users to post short textual messages called tweets. Through this platform, users can share short messages and links to other websites, images or videos. The word ‘Twitter’ refers to a series of extremely short calls or sounds called tweets. Twitter is similar to Facebook, Tumbler and Google as it can be accessed from desktops, laptops and mobile phones. All users registered with Twitter can write and post short messages limited to 140 characters or 200 bytes. In general, a message on the microblog is written by a single user and read by one or thousands of users, who are also called followers [47,45].

For over a decade, Twitter has been among the top three social network platforms in terms of user numbers and user and advertiser satisfaction rates. People use Twitter with various purposes, such as political discussions, lifestyle stories, recent local and foreign news and events and education [48,45].

In this study, we use Twitter data for the following reasons [45]:

- The scope of microblogging tends to grow continuously day by day. It is easy to use, and people can share and provide opinions on certain topics. Thus, it is an essential source.
- Twitter contains limited characters to present one’s idea concisely and effectively.
- Twitter generates a vast number of messages that increase exponentially. The extracted data can be especially large.
- Twitter users vary from person to person as they can be politicians, film stars, celebrities, athletes and world leaders. Hence, tweets contain messages by users of different castes, religions, and genders.

- Twitter users exist worldwide, and thus Twitter data are expressed in different languages.

2.3.2.1 Characteristics of Twitter Data

Twitter was developed to provide a short and precise description of one's thoughts and sentiments; therefore, tweets are formulated to comprise a maximum of 140 characters, although they can contain videos, pictures and other tweets [45].

Twitter data consist of '#Hashtags', which is the most important and meaningful symbol in the Twitter platform. This number sign, pound sign or hashtag is used to identify the topics, events, companies or keywords in every tweet on Twitter. For example, '#DonaldTrump' on Twitter showcases all current or live information, such as news, photos and videos, about the US President Donald Trump. Thus, # is a primary symbol to identify persons, companies, sports or public events around the world to which people react via Twitter [49].

Another important attribute or symbol on Twitter is '@', followed by a word or name representing users' Twitter ID. For example, '@narendramodi' in a Twitter comment refers to the username (narendramodi) used by the prime minister of India. Moreover, users can see their followers, tweets, retweets and likes and reply to other accounts using their username, i.e., '@username'. A single tweet reflects the number of people following it, the date and time it was posted, its retweet status and the text blog where the user wrote the comment. Herein, the text attribute of the Twitter dataset containing user opinion is considered for SA using the machine learning algorithm [49,45,34].

2.3.3 Internet Movie Database (IMDb)

IMDb is one of the most popular online databases for movies and personalities. It is a platform where millions of users read and write movie reviews. This database provides a large and diverse dataset for SA. IMDb contains approximately 83 million registered users, 5.6 million movie titles and 9.5 million personalities. Many users actively post movie reviews, thus providing a rich and diverse dataset of people’s sentiments and opinions. The task of classifying the overall sentiment polarity of movie reviews is highly desirable. From a researcher’s perspective, the sentiment in a movie review is generally associated with a rating (e.g., the number of stars), which is helpful for the classification process. From a user’s perspective, the rating serves as a recommendation tool for movie selection. From a producer’s perspective, the rating can be actively used for marketing and advertising purposes [38,18,17,15].

2.4 Data Mining

Data mining was introduced in the 1990s. Data mining, a branch of computer science, is the process of analysing data from different perspectives and summarising them into useful information. The extracted information is used in prediction and decision making for the benefit of organisations. The information gathered can be used in applications regardless of the field in which it is found. Data mining is used in many organisations such as insurance, banking and retail; in science research areas, such as astronomy and medicine; and in government security, such as the detection of criminals and terrorists [50,51].

In the 1960s, data were stored on computers, disks and tapes. In the 1980s, data were stored in relational databases with the help of structured query language. With the rapid technological growth from data to databases, the methods to deliver useful information from large quantities of data are in demand. Many areas have contributed to the growth of data mining. The following four areas are considered important: artificial intelligence, machine learning, statistics and data analysis [50,6,3].

As illustrated in Figure 2.1, text mining is a component of web content mining. Text mining is the process of analysing unstructured text, extracting patterns and transforming them into knowledge. Such knowledge can be used in various applications, including the simple grouping of similar documents in a collection of documents to build market intelligence. Text classification is a task in text mining used in applications which require strategic decision making. SA is the text classification technique that determines the subjective value of a text document, that is, it's positive or negative orientation. This sentiment information can help sense the public's opinion on products, political events, issues, policies and businesses [50,8,3].

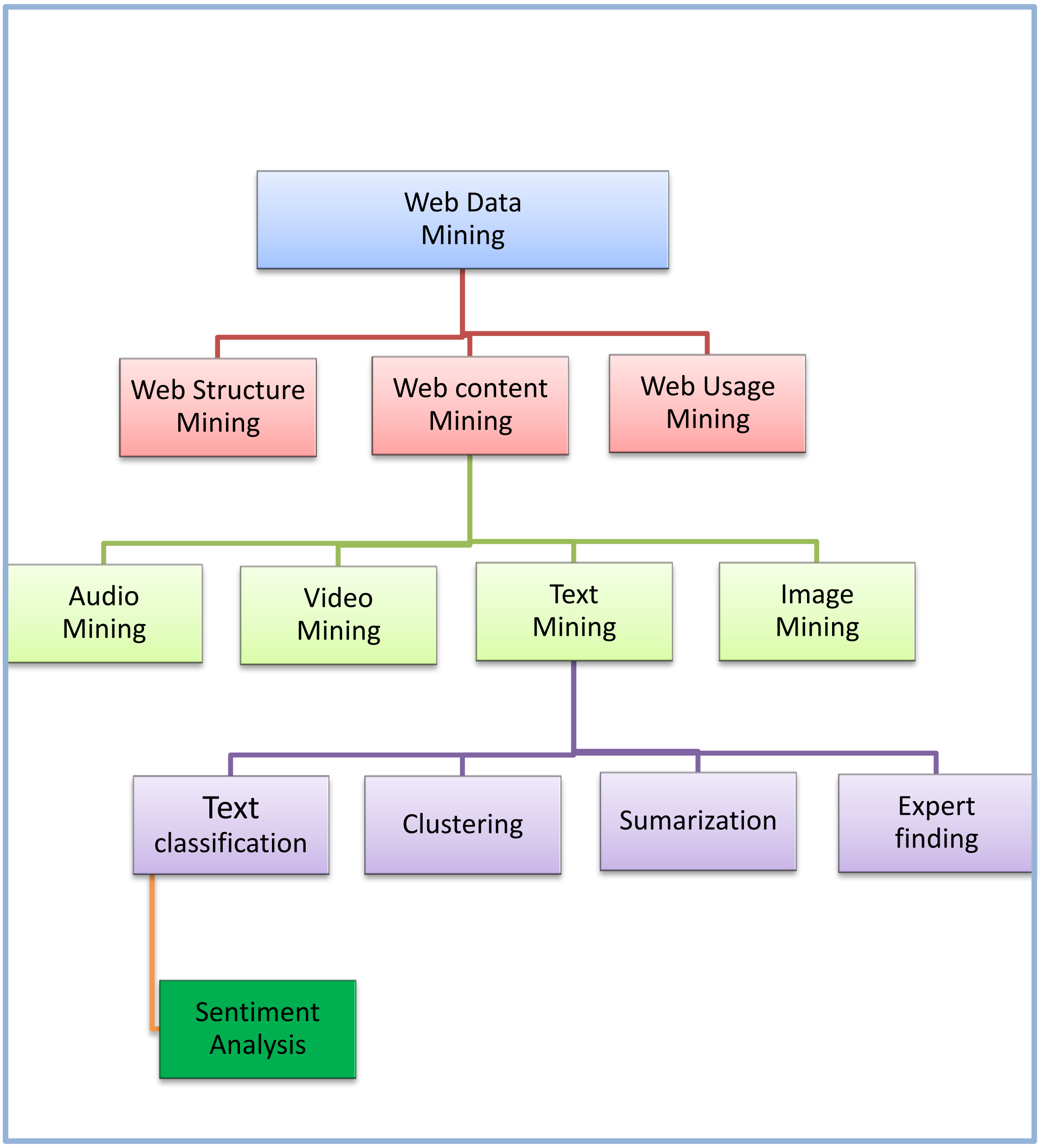

Figure 2.1: Classification of Web data mining [8]

2.5 Sentiment Analysis

SA, also known as opinion mining, is an ongoing research in the text mining field. SA belongs to the category of textual mining, which involves mining text in its simplest form to remove noisy data and building systems that can identify and extract opinions within text. SA is a combination of NLP, computational linguistics, informational retrieval, machine learning and artificial intelligence [52,3,5].

Companies or event organisers respectively seeking to have successful and well-established businesses or events should know the feedback and sentiments of their target customers or other users who might have reacted to such them via social media. Amidst technological advancement, expressing emotions, feelings and views regarding any situation has became incredibly easy through social networking sites. The reactions of customers and attendees in social media are open ended and may contain feedback in the form of written text [11,7,3].

Public opinion about businesses is readily available in the form of social media blogs. These blogs contain valuable information that can allow analysts to extract decision-making information through social media platforms. Nevertheless, assessing this achievement requires a standard process, for which SA comes into play. SA and opinion mining are two processes that aid in classifying and investigating the behaviour and approach of customers towards brands, products, events, companies and their services. SA is the automatic process of extracting emotions from users' written text by processing unstructured information and preparing a model to extract knowledge from it. SA is an interesting topic, and its development has many practical applications. Figure 2.2 shows the architecture of SA [21,12,5].

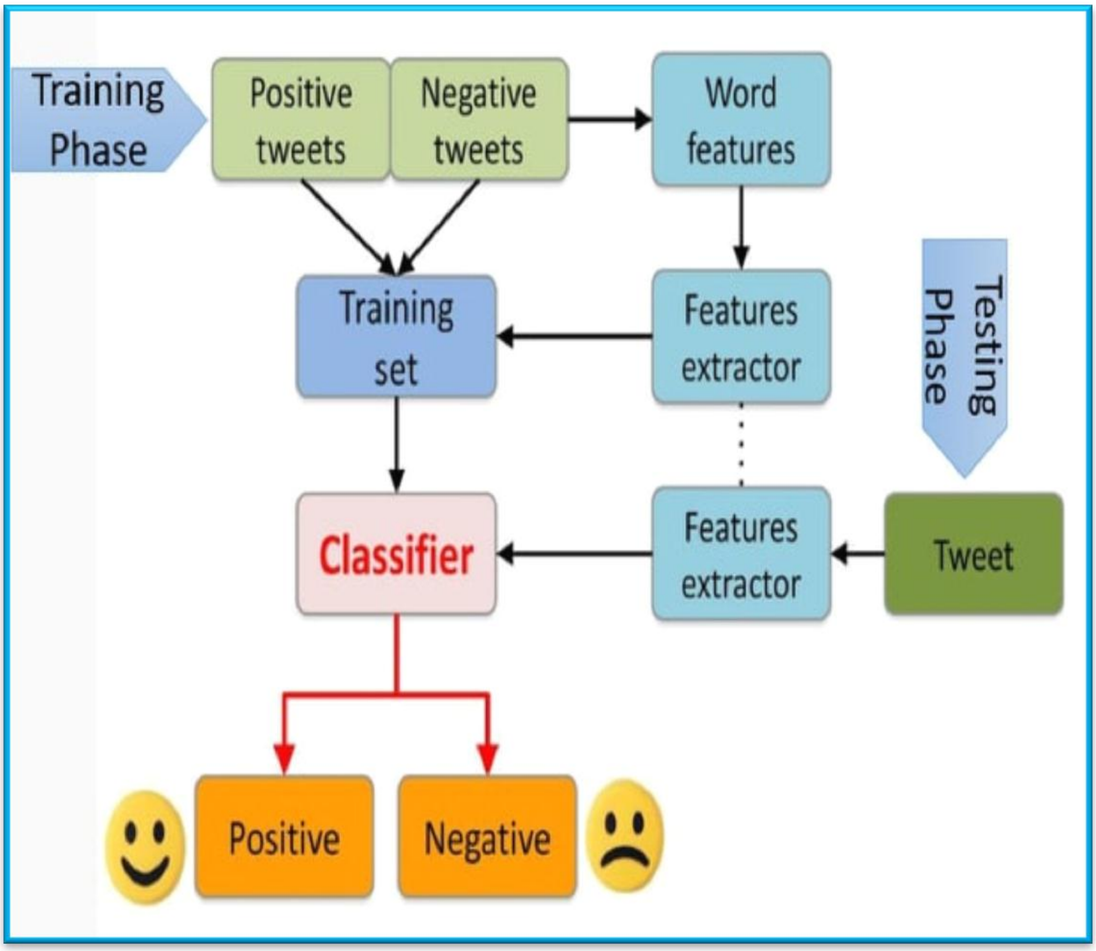

Figure 2.2: Sentiment analysis architecture [21]

The sentiment classification techniques are briefly explained in Figure 2.3. One of these techniques, that is, neural network technology, is used in our proposed system.

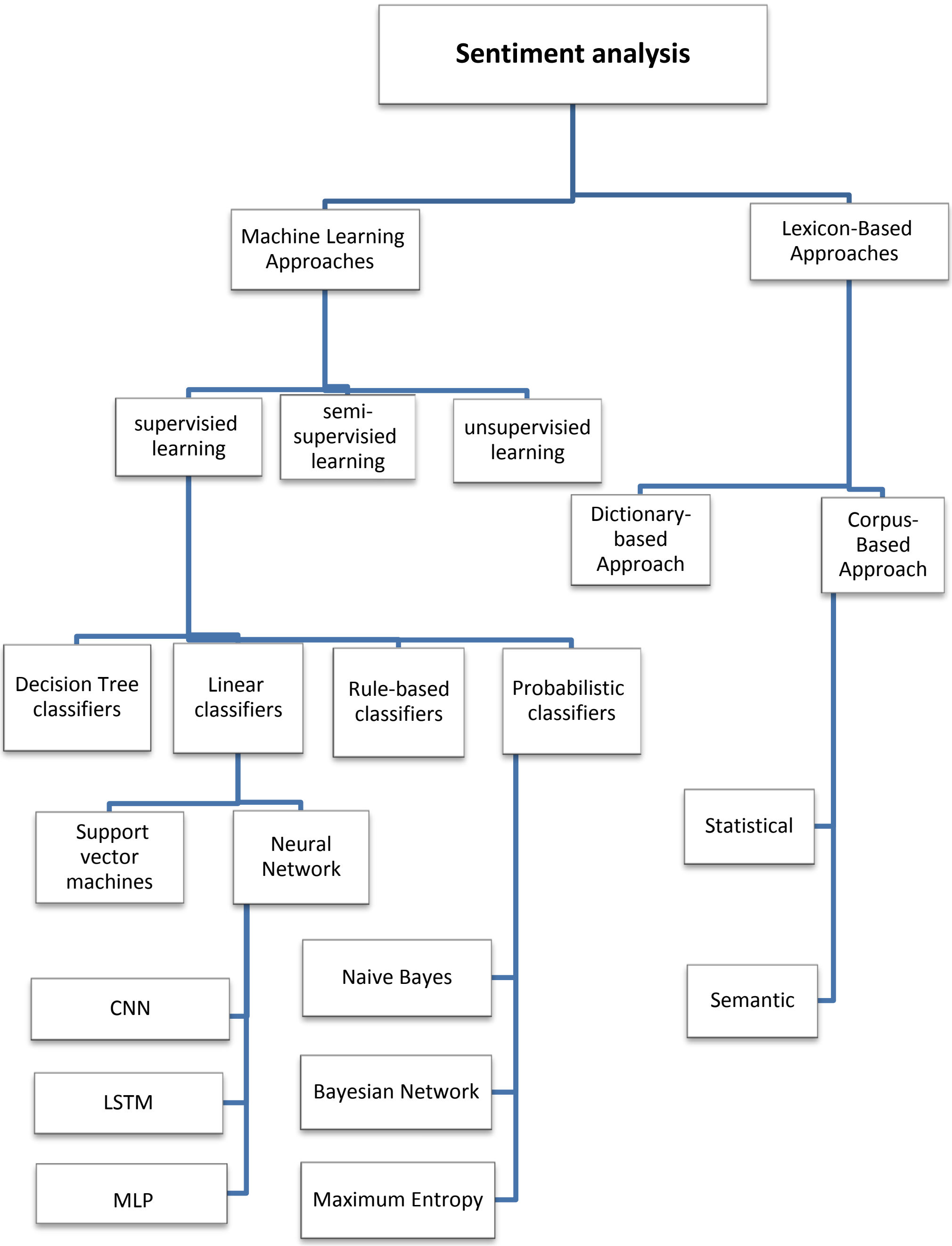

Figure 2.3: Sentiment classification methods [48]

2.5.1 Approaches for SA

SA refers to the general method of extracting polarity from semantic orientation, which refers to the strength of words and polarity text or phrases. The two main approaches for extracting sentiment automatically are the lexicon-based approach and machine learning-based approach [14].

2.5.1.1 Lexicon-Based Approaches

Lexicon-based methods use a sentiment dictionary with opinion words and match these words with available data to determine polarity. They assign sentiment scores to opinion words describing how positive and negative these words are. Lexicon-based approaches mainly rely on a sentiment lexicon, that is, a collection of known and precompiled sentiment terms, phrases and even idioms developed for traditional genres of communication; an example is the opinion finder lexicon. This approach has two sub classifications [12,14,19]:

1- Dictionary-based approach:

It is based on the usage of terms (seeds) that are generally collected and annotated manually. This set grows by searching the synonyms and antonyms in a dictionary. An example of this dictionary is WordNet, which is used to develop the thesaurus called SentiWordNet.

2- Corpus-based approach:

The objective of the corpus-based approach is to provide dictionaries related to a specific domain. These dictionaries are generated from a set of seed opinion terms that grows through the search of related words by means of statistical or semantic techniques.

- Methods based on statistics, including latent semantic analysis
- Methods based on semantics, including those that use synonyms and antonyms or relationships from a thesaurus such as WordNet

2.5.1.2 Machine Learning Approaches

Machine learning-based approaches use classification techniques to classify text into classes. This approach has mainly two types [53,19,14]:

1- **Unsupervised learning:**

It does not involve categories and provide accurate targets at all; therefore, it relies on clustering. Unsupervised methods are used when finding labelled training documents is difficult.

2- **Supervised learning:**

It is based on labelled datasets; thus, labels are provided to the model during the process. These labelled datasets are trained to obtain meaningful outputs when encountered during decision making. The success of these learning methods mainly depends on the selection and extraction of the specific set of features used to detect sentiments. The machine learning approaches are applicable to SA mainly belong to supervised classification. In machine learning techniques, two sets of data are required:

1. Training set

2. Test set

Several machine learning techniques have been formulated to classify tweets into classes. Machine learning techniques such as neural networks, NB, maximum entropy (ME) and SVM, have achieved great success in SA. Machine learning starts with collecting training data. Then, we train a classifier on the training data. Once a supervised classification technique is selected, the next important decision to make is to select features. Features can tell us how documents are represented [53].

2.6 Preprocessing of Documents

One of the most important steps in information retrieval and SA is document preprocessing, which is aimed at preparing documents or texts to be used. The preprocessing step can involve many substeps before a document can be deemed ready to use. Figure 2.4 describes the preprocessing stage [55,56,2].

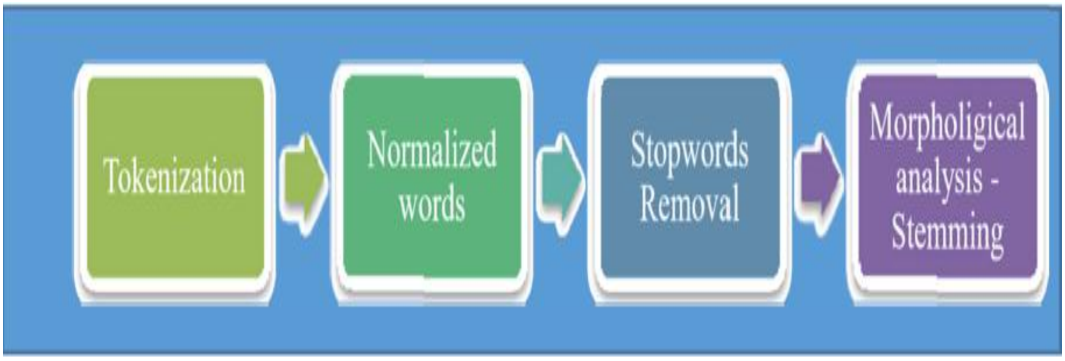

Figure 2.4: Preprocessing stage [54]

1. Tokenization

Tokenisation is important in NLP. In tokenisation, a text document is separated into tokens using a space to split each word from another. Tokens could be numbers, symbols and words. A word's boundaries are specified by a tokenizer which uses punctuation marks and white spaces as delimiters between major segments or words [55,49,45].

2. Normalisation

Normalisation is a preprocessing stage that uses NLP. Data normalization is essentially a type of process wherein data within a database is reorganized in such a way so that users can properly utilize that database for further queries and analysis. Normalising datasets and removing extra characters are considerably important. The general aim is to clean up text by removing punctuation marks and numbers. In English documents, normalisation involves converting capital letters to lowercase letters. Therefore, normalisation in English is relatively easy, especially given the many packages and tools that support English and are available in most programming languages [56,55,49,45].

3. Stop Word Removal

Documents have large dimensionality because of redundant words called stop words. Various words appear repeatedly in documents. However, they basically have no effect because they are used to join the words in sentences. Generally, stop words have no contribution to the content or context of textual documents. They present an obstacle in understanding the content of documents because of their high occurrence rate [57,56,55,49]. Table (2.1) lists some examples of English stop words.

Table 2.1: Examples of English stop words

Word	Word	Word
I	in	else
A	Is	Who
About	It	Will
An	Of	With
Are	On	The
As	Or	What
At	That	Was
Be	The	We
By	This	Us
Com	To	You

4. Stemming Process

Stemming is the process of reducing words to their word stem, base or root form (for example, books—book and looked—look). The main two algorithms are the porter stemming algorithm (removes common morphological and inflexional endings from words) and the Lancaster stemming algorithm (a relatively aggressive stemming algorithm). The aims of the stemming approach are as follows: deleting affixes from words (suffixes and prefixes), keeping time and memory space, precise matching of stems and decreasing the number of features in the feature space [57,55,49,45].

2.7 Feature Extraction

Feature extraction is the process of reducing dimensionality in which an initial set of raw data is reduced to small, manageable groups for processing. In the context of feature extraction, all original attributes are converted to a new attribute space without eliminating them. This process is conducted by substituting the original attributes using a reduced representation group of attributes. Thus, when the input data contain many attributes which should be processed, they are converted to a new and extra compact attribute space [58] .

Feature extraction approaches are used to extract features from datasets whose formats, such as text, are not applicable to machine learning algorithms. The extracted text features thus come in a particular format that matches these algorithms. Term frequency–inverse document frequency (TF-IDF) representation is one of the major approaches used in feature extraction because it considers text as a series of characters that are converted to numerical attributes' vectors with a fixed size taken via machine learning algorithms instead of text documents with different lengths. Features are extracted from documents, and weights related to each feature define the importance of the document features; a document is given as:

Dj = (w_{1j}, w_{2j}, w_{3j}, w_{4j},…,w_{nj}), where wij is the weight of feature i in document j, indicating the significance and relevance of the feature [58,59].

2.7.1 Feature Extraction with TF-IDF

TF-IDF is a statistical method for evaluating the significance of a word in given documents. Term frequency (TF) refers to how many times a given term appears in a document. Inverse document frequency (IDF) measures the weight of a word in a document, that is, whether the word is common or rare in the entire document. The TF-IDF intuition follows that the terms appearing frequently in a document are less important than the terms that are rarely used. The TF-IDF method uses the vector space modelling technique for text document representation. TF-IDF is used in document classification, text summarisation and recommender systems among other use cases [58,60].

2.7.1.1 Term Frequency (TF)

TF(t,d) is the total number of times a given term (t) appears in document (d) against the total number of all words in the document. If the frequency of a given term increases in a document, then its TF also increases. The content, format and length of text are different, these factors influence the TF value, and the method generally used to deal with this problem is normalisation. In practical applications, classification results are influenced if a text feature includes many stop words (the, an, my…) and the high-frequency occurrence of these words increases their TF weights. In conclusion, TF results greatly depend on the removal of stop words. TF is calculated as follows:

$$\textbf{Term Frequency (TF)} = \frac{F_{ij}}{F_{dj}} \qquad \textbf{(2.1)}$$

where F_{ij} is the total number of occurrences of term i in document j. Fd_j is the total number of terms occurring in document j [58,59,60].

2.7.1.2 Inverse Document Frequency (IDF)

IDF(t,D)(where D represent term(t) appears in corpus documents) is a measure of how much information a word provides. It measures the weight of a given word in an entire document. IDF shows how common or rare a given word is across all documents.

One of the most utilised approaches in feature weighting systems is the IDF approach. It is used to evaluate a feature’s significance in a group of documents and is thus different from the TF approach that estimates a feature’s significance in a single document. IDF is calculated as follows:

$$IDF = \log \frac{N}{Ni} \qquad (2\,.2)$$

where

N: the total number of documents in a group of documents,

Ni: the number of documents in the group in which word i occurs [58,59].

2.7.1.3 Term Frequency–Inverse Document Frequency

TF-IDF is the product of term frequency tf (t,d) and inverse document frequency idf(t,D). TF-IDF describes a numerical series that shows the importance of a term (word) to a document or corpus. TF-IDF has many advantages over TF and IDF feature extraction. For instance, ‘the’ is commonly used in many documents, and the word has a high TF. However, the IDF of the word ‘the’ is low. Hence, with TF and IDF considered, the word should be given a low weight. It is calculated as follows:

$$\boldsymbol{TF\_IDF = TF * IDF} \qquad (\mathbf{2.3})$$

where

TF: The number of times word i appears in a document.

IDF: The inverse document frequency of the word across a set of documents[58,59,60].

2.8 Feature Selection

Feature selection is a core concept in machine learning that seriously impacts model performance. Specifically, the data features used to train machine learning models greatly influence performance. In feature selection, the task is to select a subset of features from a set of original features that offer unchanged meaning with no loss in information. Having irrelevant features in data can decrease the accuracy of models and cause the model to learn on the basis of irrelevant features. Feature selection is an important preprocessing stage in various research that results in highly precise results [63,62,61]. Various advantages are related to feature selection [61,64,65]:

1. Feature selection removes redundant, noisy and irrelevant data and thus leads to reduced run time and storage media requirements.
2. It improves accuracy. A reduction in misleading data equates to an improvement in modelling accuracy.
3. It reduces training time. The use of few data points reduce algorithm complexity and hastens algorithm training.

2.8.1 Feature Selection Methods

Feature selection involves three methods, namely, filter, wrapper and embedded methods. The filter method serves as a preprocessing step to rank the features. The highly ranked features are selected and applied to a predictor. In the wrapper method, the feature selection criterion is the performance of the predictor, i.e. the predictor is wrapped on a search algorithm which finds a subset with the highest predictor performance. The embedded method combines the qualities of the filter and wrapper methods and includes variable selection as part of the training process without splitting the data into training and testing sets [67,66,61].

2.8.1.1 Filter Method

The filter method for feature selection applies a statistical measure to assign a score to each feature. The features are ranked according to their scores and are either selected to be retained or removed from the dataset, as shown in Figure 2.5. Filter methods are often univariate methods that consider features independently or as dependent variables; some examples of filter methods include the chi-square test, information gain and correlation coefficient scores [67,61].

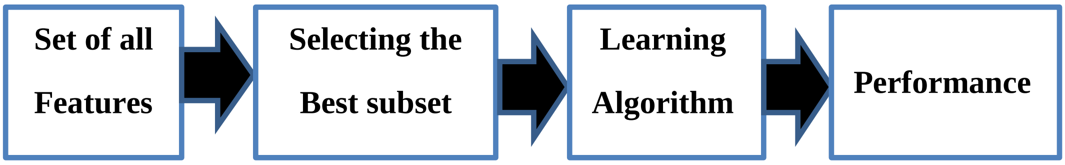

Figure 2.5: Filter Method [67]

2.8.1.1.1 Chi-Square Test For Feature Selection

The chi-square (x^2) statistical test has been widely accepted as a statistical hypothesis test to evaluate the dependency among two variables. In NLP, the chi-square test is often applied to test the independence between the occurrence of a term and the occurrence of a class. It is often used as a feature selection method in NLP [68,69].

To calculate X^2, we suppose that we have a set of training instances that belong to positive and negative classes. In calculating the \chi^2 score of a feature X, the contingency table (Table 2.2) can be used to illustrate the idea.

Table 2.2: Contingency Table

	Positive class (c)	**Negative class**	**Total**
Feature X occurs(t)	A	B	A+B
Feature X does not occur	C	D	C+D
Total	A+C	B+D	N

A: numbers of positive instances that contain feature X

B: numbers of negative instances that contain feature X

C: numbers of positive instances that do not contain feature X

D: numbers of negative instances that do not contain feature X

N: total number of instances (total number of documents in the corpus)

A + B : number of instances that contain feature X

C + D : number of instances that do not contain feature X

A + C : number of positive instances

B + D: number of negative instances

The following equation calculates the chi square (x^2) using Table (2.2):

$$\mathbf{X^2\ (t,c)=\frac{N(AD-BC)^2}{(A+C)+(B+D)+(A+B)(C+D)}} \quad \text{............(2.4)} \quad \mathbf{[69]}$$

A high value of x^2 indicates that term t and class c are dependent, thereby making term t a useful feature given that the occurrence of t indicates that the document is likely to be seen in class c. Utilising the property of x^2 with high x^2 values of term t indicates a high likelihood of occurrence in class c. The key aspect of our discovery is that words with high x^2 statistics tend to be the keywords for class identification. Thus, we use the chi-square statistical test to select the lexicon that particularly caters to the specific class identification task of short sentences [71,70,68,63].

2.9 Artificial Neural Network (ANN)

ANN is a branch of artificial intelligence based on the intelligence of the human brain. The concept of ANN was inspired by human biology and the way neurons of the human brain function together to understand inputs from human senses. Neurons are linked to one another by axons and dendrites, and the connecting regions are referred to as the synapses between axons and dendrites [72].

Figure 2.6 shows a biological neuron cell. Synaptic connection strengths often change in response to external stimuli. This change is reflects the process of learning in living organisms. This biological mechanism is simulated in ANN containing computational units known as neurons. The computational units are connected by weights that serve the same role as the strengths of synaptic connections in biological organisms. Each input into a neuron is scaled by a weight that affects the computed function in that unit. ANN calculates a function of the inputs by propagating the calculated values from the input neurons to the output neurons and using the weights as intermediate parameters. Learning takes place by changing the weights connecting the neurons. External stimuli are required for learning in biological organisms, and training data containing examples of input–output pairs of the learning function provide the external stimulus in ANNs [72, 73]

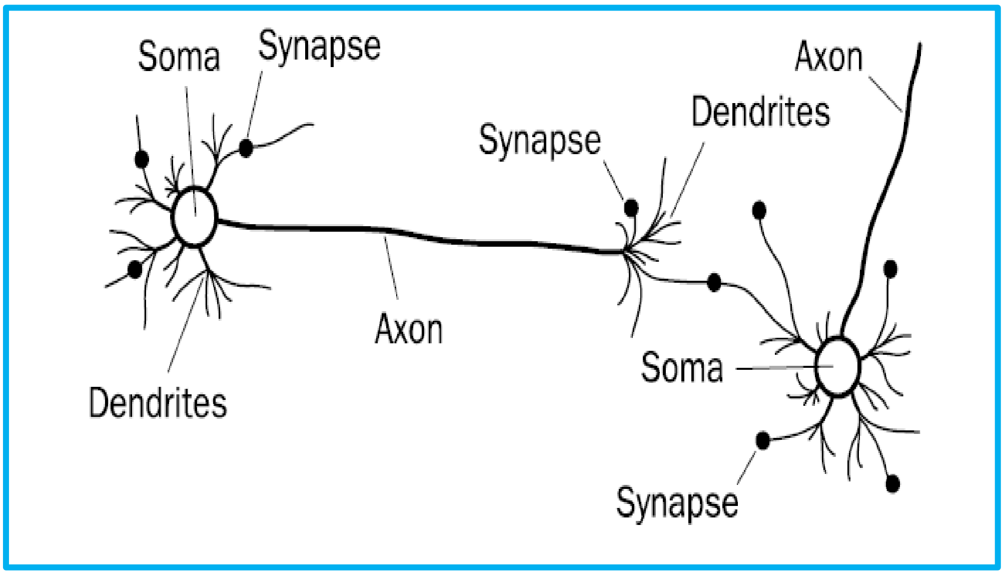

Figure 2.6: Biological Neuron Cell [73]

Machine learning algorithms that use neural networks generally do not need to be programmed with specific rules that define what to expect from the input. The neural network learning algorithm instead learns from processing many labelled examples (i.e. data with ‘answers’) that are supplied during training and then uses the answer key to learn the required characteristics of the input to construct an accurate output. Once a sufficient number of examples have been processed, the neural network can begin to process new, unseen inputs and successfully return accurate results. When additional examples and a variety of inputs are available, the results typically become increasingly accurate because the program learns with experience [73].

Neural networks are tools and approaches used in machine learning algorithms. A neural network itself may be used as a piece in many different machine learning algorithms to process complex data inputs into a space that computers can understand. Neural networks are applied to many real-life problems, including speech and image recognition, spam email filtering, finance and medical diagnosis to name a few [72,74].

Figure 2.7 describes a model of an artificial neuron. The neural model is made out of the following [72,73,75]:

1. An arrangement of synapses or neural connections; each is described by weight or strength. Each input that connects to a neuron is multiplied by weight *w*.

2. Adding a bias *bk* for summing the inputs weighted for each neuron. The objective of the added bias is to increase or decrease the net inputs of the activation function depending on whether it is positive or negative.

3. Limiting the amplitude via an activation function *ϑ(.)* for each neuron output.

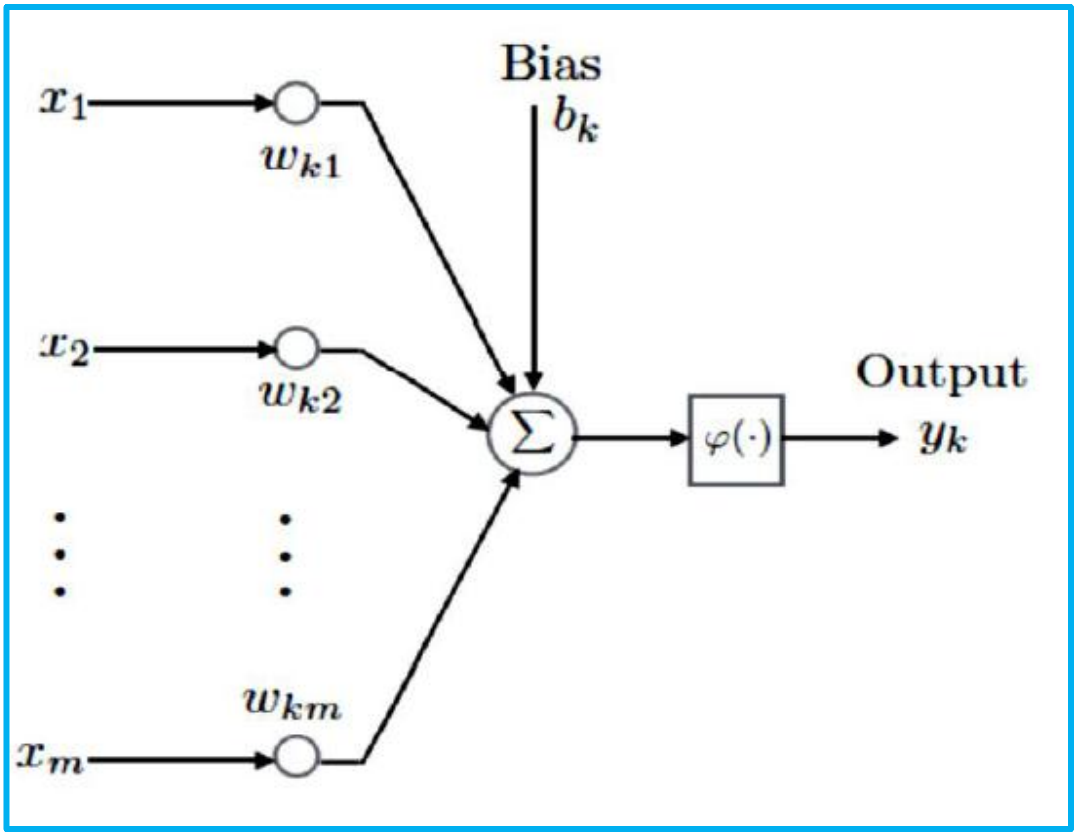

Figure 2.7: Artificial Neural Network [72]

The mathematical expression of the neural network in Figure 2.7 is described in the following equation:

$$yk = \vartheta\left(\sum_{j=1}^{m} wkj\ xj + bk\right) \ldots\ldots\ldots\ldots\ldots (2.5) \qquad [72]$$

where ***x1***, .., ***xm*** represents the inputs and ***w1***, .., ***wm*** represents the weight of neuron ***k***; ***yk*** represents the outputs of the neuron, ***bk*** is denoted as the bias, and ***ϑ (.)*** represents the activation function.

2.9.1 Architecture of Artificial Neural Network

The simplest neural network is called perception. This neural network includes a single layer of input and a node of output. As illustrated in Figure 2.8, a single layer network is composed of one layer of weights connected from the inputs to the outputs. The input layer transmits the data to the output layer, and all computations are completely visible to the user. A single layer is used to solve noncomplex computation problems [76,73].

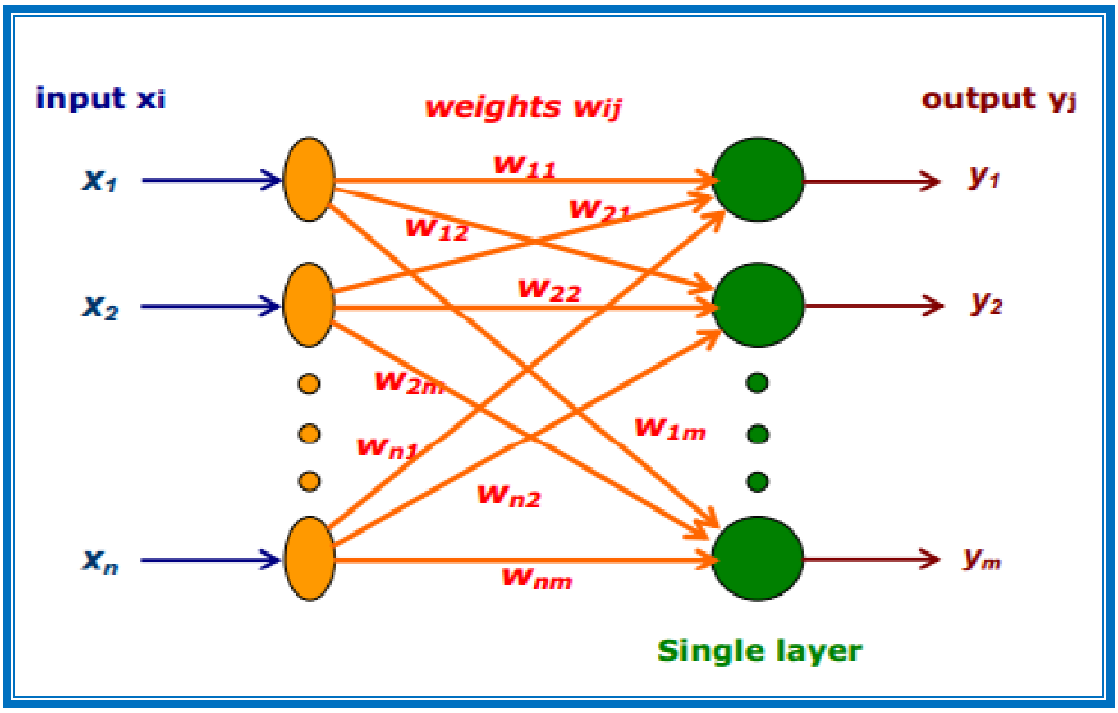

Figure 2.8: Single Layer Neurons [76]

Multilayer neural networks consist of an input layer, one or more hidden layers and an output layer (Figure 2.9). Any node from the input layer is connected to the node from the hidden layer, and every node from the hidden layer is connected to a node in the output layer. Each connection has several weights. The input layer represents the raw information fed into the network. Each single input to the network is multiple and is transmitted down to the nodes in the hidden layer. The hidden layer takes the data from the input layer. It uses input values and modifies them using a weight value. The new value is then transmitted to the output layer, where it changed by several weights according to the relationship between the hidden layer and the output layer. The output layer processes the information received from the hidden layer and produces an output. The output is then processed by the activation function [77,76,73,72].

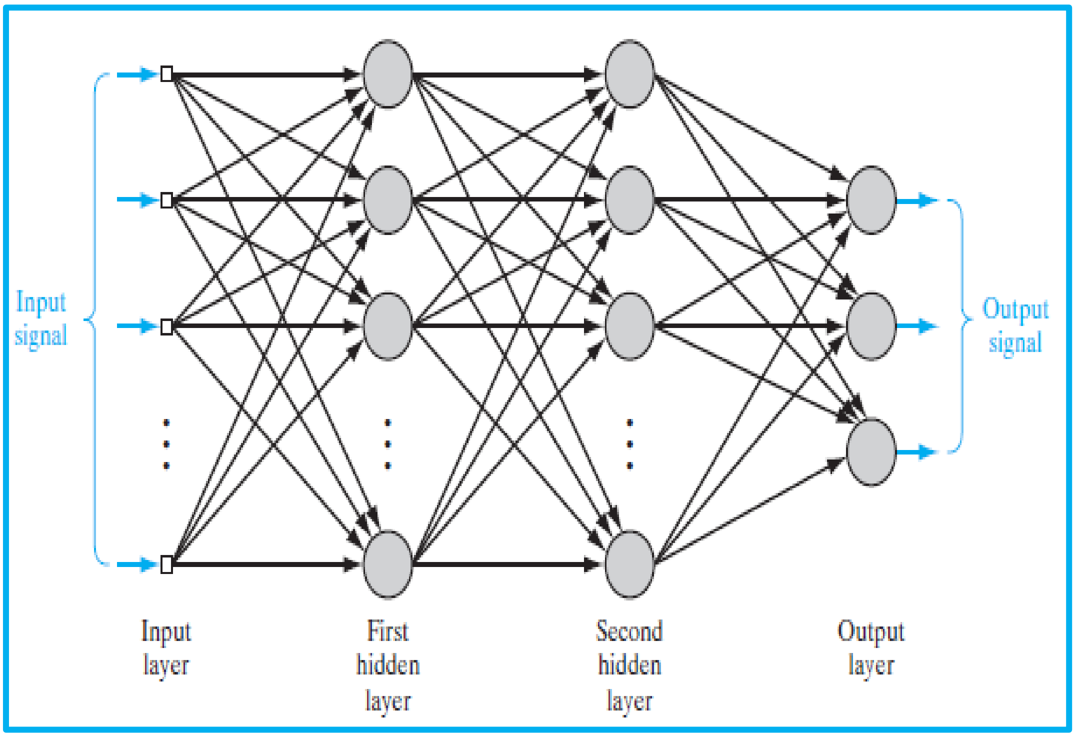

Figure 2.9: Multilayer Neurons [77]

2.9.2 Activation Functions

The activation function is used to transform the activation level of a unit (neuron) into an output signal. It is used to determine the output of the neural network, such as ‘yes’ or ‘no’. It maps the resulting values between 0 to 1 or −1 to 1 (depending upon the function). The process of an ANN is to sum up the output of the connected weights and the input signal and then produce an output or activation function.

Activation functions are an extremely important feature of ANNs. They basically decide whether a neuron should be activated or not, that is, whether the information that the neuron is receiving is relevant for the given information or should be ignored.

Many types of activation functions have been proposed over the years. However, the best practice confines the use to only a limited type of activation functions. Several common activation functions are listed in Table (2.3) [78,72,73].

Table 2.3: Types of Activation Functions.

Activation function	Description	Equation
Linear	A straight line function where activation is proportional to input.	$f(x) = \mathrm{x}$ (2.6)
Sigmoid	Transforms the input, which can have any value between 0 and 1	$\boldsymbol{f}(\boldsymbol{x}) = \frac{\mathbf{1}}{\mathbf{1}+\boldsymbol{e}^{-\boldsymbol{z}}}$ (2.7)
ReLU	Provides the same benefits as Sigmoid but with better performance.	$F(x) = \max(0,x)$ (2.8)
Step	Used in decision-making neurons for classification and pattern recognition.	$f(x)\begin{cases}1 & IF\ \ X\ \geq\ 0\\ 0 & IF\ \ X\ <\ O\end{cases}$ (2.9)
Tanh	It is bound to the range (-1, 1).The gradient is stronger for tanh than sigmoid (derivatives are steeper).	$\tanh(\mathrm{x}) = \frac{2}{1+e^{-2x}} - 1$ (2.10)
Sign	Used in decision-making neurons for classification and pattern recognition.	F(x)=$\begin{cases}+1 & IF\ X\ \geq\ 0\\ -1 & IF\ X<\ O\end{cases}$ (2.11)

2.9.2.1 Linear Activation Function

Linear activation takes the inputs, multiplies them by the weights for each neuron and creates an output signal proportional to the input. Figure 2.10 illustrates the plot of the linear activation function [78]. The linear activation function is defined as follows:

Equation: $\mathbf{f(x) = x}$ **………….. (2.6)**

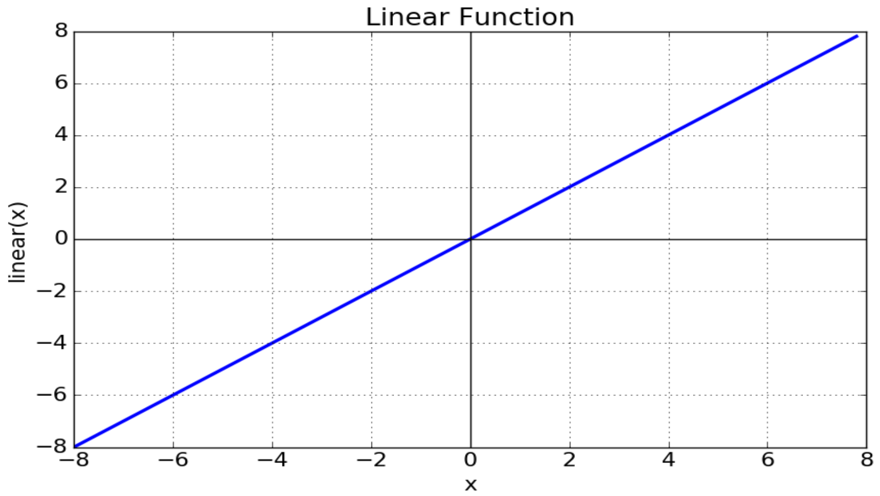

Figure 2.10: Linear Activation Function [78]

2.9.2.2 Sigmoid or Logistic Activation Function

A sigmoid takes a real value as input and outputs another value between 0 and 1. It is easy to deal with, and it embodies the good properties of activation functions, including their nonlinearity, continuously differentiability and fixed output range. As the probability of anything exists only between the range of 0 and 1, the sigmoid is the right choice [79]. Figure 2.11 illustrates the plot of the sigmoid activation function. It is defined as

Equation: F(X) = $\frac{1}{1+e^{-z}}$ **……………… (2.7)**

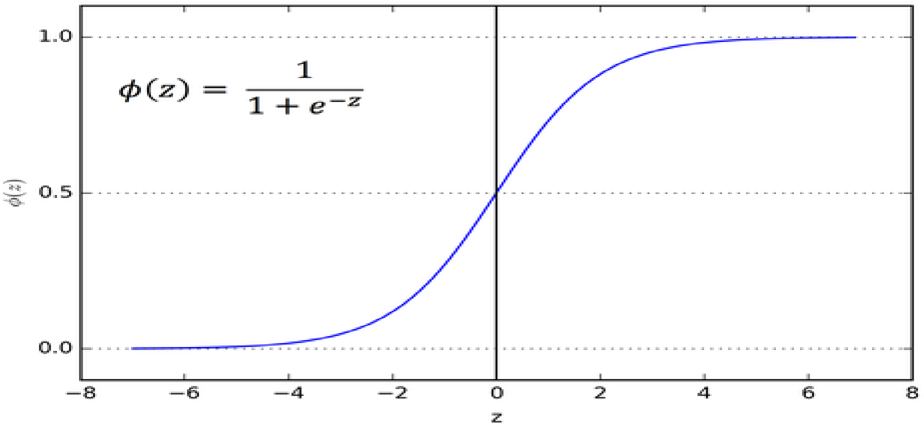

Figure 2.11: Sigmoid Activation Function [79]

2.9.2.3 Rectified Linear Unit (ReLU)

The rectified linear unit (ReLU) is the most widely used activation function in the world. It is used in almost all CNNs or deep learning approaches and provides the same benefits as a sigmoid but with better performance. The ReLU is less computationally expensive than the sigmoid because it involves simpler mathematical operations [79]. Figure 2.12 illustrates the plot of the ReLU activation function. It is defined as follows:

Equation: F(x) = max (0,x) ……………(2.8)

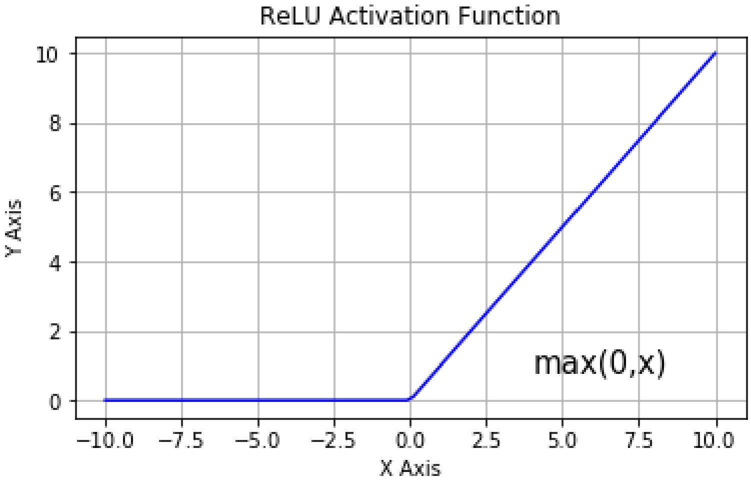

Figure 2.12: ReLU Activation Function [79]

2.9.2.4 Threshold or step activation function

This function is the simplest and can be considered as a yes or no function. If the value of z is above the threshold value, then the activation is set to 1 or yes, and the neuron is fired. If the value of z is below the threshold value, then activation is set to 0 or no, and the neuron is not fired, it is useful for binary classification [78]. Figure 2.13 illustrates the plot of the threshold activation function. It is defined as follows:

$$\textbf{Equation: } f(x) = \begin{cases} \mathbf{1} & \boldsymbol{IF \;\; X \;\geq\; 0} \\ \mathbf{0} & \boldsymbol{IF \;\; X \;<\; O} \end{cases} \quad \text{……………………(2.9)}$$

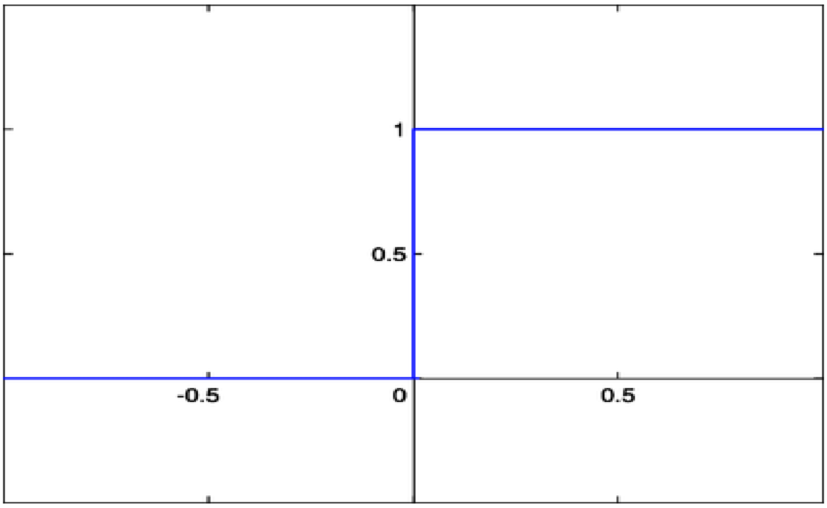

Figure 2.13: Threshold or step activation function [78]

2.9.2.5 Tanh activation function

It is nonlinear in nature, so we can stack layers. It is bound to the range (-1, 1).The gradient is stronger for tanh than sigmoid (derivatives are steeper).In practice, optimization is easier in this method hence in practice it is always preferred over Sigmoid function. And it is also common to use the tanh function in a state to state transition model (recurrent neural networks). Figure 2.14 illustrates the plot of the Tanh activation function. It is defined as follows:

$$\textbf{Equation}: \ \tanh(\mathbf{x}) = \frac{2}{1+e^{-2x}} - 1 \ldots\ldots\ldots\ldots\ldots\ldots..(2.10)$$

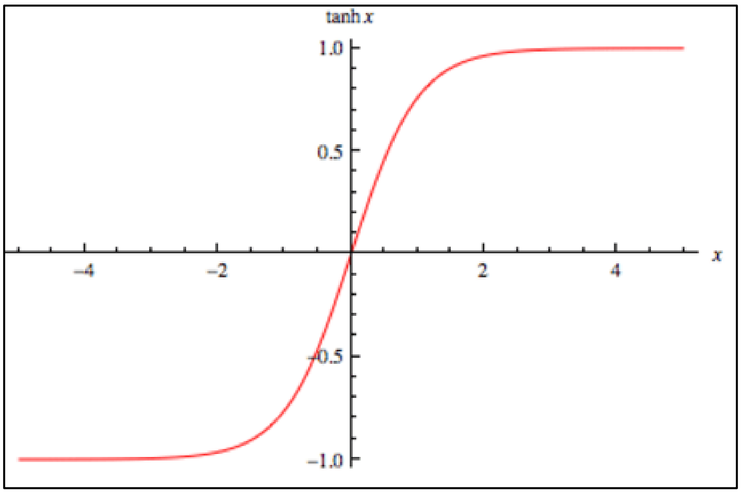

Figure 2.14: Tanh activation function [78]

2.9.3 Learning Algorithm

The learning algorithm involves the process of extracting patterns appropriate for application in a new situation. The main aim is to adapt the network to the specific input–output transformation process.

Backpropagation is the essence of neural net training. It is the method of fine-tuning the weights of a neural net on the basis of the error rate obtained in the previous epoch (i.e. iteration). The appropriate tuning of the weights reduces the error rates and helps generate a reliable model by increasing its generalisation [80,81].

The backpropagation algorithm is an example of a supervised learning method; it is used at each layer to minimise the error between the desired output and the actual output. It is a standard method for training ANNs. This method calculates the gradient of a loss function with respect to all the weights in the network. Before starting the backpropagation learning process, the following parameters are required [80,81.82]:

- set of training patterns, input and target.
- a value for the learning rate.
- a criterion that terminates the algorithm.
- a methodology for updating weights.
- selecting the activation function.
- initial weight values (typically small random values).

The basic backpropagation algorithm consists of the following main steps [80,81,82]:

1. Input X arrives through the preconnected path.
2. The input is modelled using real weights W. The weights are usually randomly selected.
3. The output is calculated for every neuron from the input layer, to the hidden layers and to the output layer.
4. The error in the outputs is calculated.

$$\textbf{Error = (Actual output - Desired output)} \qquad \textbf{(2.11)}$$

5. The weights are adjusted from the output layer to the hidden layer to reduce the error.
6. The process is repeated until the desired output is achieved.

Figure 2.15 shows the backpropagation.

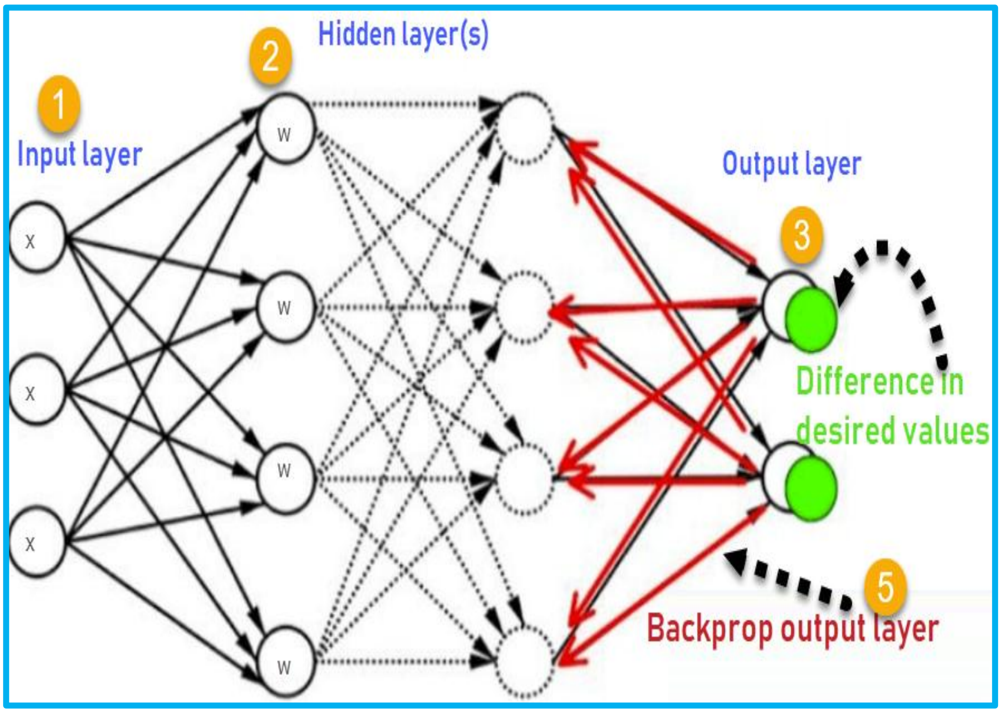

Figure 2.15: Backpropagation

2.9.4 Multilayer Perceptron (MLP)

The MLP is a feedforward neural network with one or more hidden layers. It consists of a number of neuron layers. The first layer represents the input signals to the neural network, the second layer is the hidden layer that confirms the nonlinearity of the model, and the last layer contains the outputs of the network. In the simplest arrangement, each layer in the hidden section is connected to the previous layer, and the next layer finally produces an output vector from the previous layers. The output of a layer is computed by applying the neuron activation function for all neurons on the layer, as shown in the following equation:

$$Yl=f(Wl*yl + bl)\ldots\ldots\ldots\ldots\ldots \text{(2.12)} \qquad [72]$$

where W represents the weight assigned to each pair of neurons from the layer input to the hidden layer or from the hidden layer to the output layer. b is a vector of bias terms for each neuron in the layers, y is the input from the previous layer, l refers to the index of the layer, and f represents the activation function. Equation (2.12) is applied from the first layer up to the last layer (output layer) in sequence (feedforward) [83,82,77,76]. The steps of MLP are as follows:

1-Initialisation: - Set initial weights *w1; w2; ...; wn* and threshold *θ* to random numbers.

2-Activation: **-** Activate the perceptron by applying inputs x1*(p) , x2(p), ……, xn (p)* and desired output Y*(p).* Calculate the actual output at iteration *p = 1*.

$$\boldsymbol{y(p) = RELU \sum_{i=1}^{n} xi(p)\ .wi(p) \dots\dots\dots\dots..(2.13)} \qquad [72]$$

where *n* is the number of perceptron inputs and RELU is a step activation function.

3-Weight training: Update the weights of the perceptron.

$$\boldsymbol{wi(p+1) = wi\ (P) + \Delta wi(p)\ \dots\dots\dots\dots(2.14)} \quad [72]$$

The weight correction is computed by the delta rule as follows:

$$\boldsymbol{\Delta wi(p) = \alpha\ .xi\ (p).e(p) \dots\dots\dots\dots.(2.15)} \qquad [72]$$

where $\boldsymbol{\alpha}$ represent learning rate, xi represent inputs and e represent error rate.

4- Iteration: Increase iteration p by 1, go back to Step 2, and repeat the process until convergence.

2.10 Performance Measures

Different performance metrics are used to evaluate different machine learning algorithms. The metrics adopted to evaluate any machine learning model is important. The choice of metrics influences how the performance of machine learning algorithms is measured and compared. The confusion matrix is one of the most intuitive and easiest metrics used for finding the correctness and accuracy of models between predicted classes and actual classes. It is used for the classification problem in which the output can be of two or more class types [84,85]. Table (2.4) presents the confusion matrix.

Table 2.4: Confusion Matrix

		Predicted	
Actual		**Positive**	**Negative**
	Positive	**True Positive TP**	**False Negative FN**
	Negative	**False Positive FP**	**True Negative TN**

The terms associated with the confusion matrix are as follows [84,85]:

1. **True Positives (TP):** True positives are the cases in which the actual class of a data point is 1 (true) and the predicted class is also 1 (true).
2. **True Negatives (TN):** True negatives are the cases in which the actual class of a data point is 0 (false) and the predicted class is also 0 (false).
3. **False Positives (FP):** False positives are the cases in which the actual class of a data point is 0 (false) and the predicted class is 1 (true). The case is false because the model predicts incorrectly and positive because the class predicted is a positive one.
4. **False Negatives (FN):** False negatives are the cases in which the actual class of a data point was 1 (true) and the predicted class is 0 (false). The case is false because the model predicts inaccurately and negative because the class predicted is negative.

Several standard terms representing the performance criteria have been defined, and the two class matrices can be defined as follows [84,85]:

1- **Accuracy:** Accuracy in classification is the number of correct predictions made by the model over all predictions. Accuracy can be defined as follows (Equation(2.16)):

$$Accuraacy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (2.16)$$ [85]

2- **Precision:** It is calculated as the number of true positives divided by the total number of elements labelled as belonging to the positive class. Precision can be defined as follows (Equation (2.17)):

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \qquad (\mathbf{2.17}) \qquad [85]$$

3- **Recall (Sensitivity):** It is calculated as the number of true positives divided by the total number of elements that actually belong to the positive class. Recall can be defined as follows (Equation (2.18)).

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \qquad (\mathbf{2.18}) \qquad [85]$$

4- **Error rate (ERR):** It is calculated as the number of all inaccurate predictions divided by the total number of the predictions. ERR can be defined as follows (Equation (2.19)): $\boldsymbol{ERR} = \frac{\boldsymbol{FP+FN}}{\boldsymbol{TP+TN+FP+FN}}$ $(\mathbf{2.19})$ [85]

5- **F1 score:** In the statistical analysis of binary classification, the F1 score (F-score or F-measure) is a measure of a test's accuracy. It considers the precision p and recall r of the test to compute the score. The F1 score is the harmonic average of precision and recall. An F1 score reaches its best value at 1 (perfect precision and recall) and its worst value at 0. The F1 score can be defined as follows (Equation (2.20)):

$$\mathbf{F1\ score} = \mathbf{2} * \frac{\mathbf{precision} * \mathbf{recall}}{\mathbf{precision} + \mathbf{recall}} \qquad (\mathbf{2.20}) \qquad [85]$$

2.11 Chapter Summary

This chapter provides a background of the work related to Twitter and movie reviews SA. In particular, the chapter discusses the use of SA, social networks, Twitter, preprocessing stages, feature extraction techniques, feature selection techniques and ANN. In this work, TF-IDF is used in document extraction, and the chi-square technique is used for feature selection. For the SA of data (positive and negative), MLP is adopted.

Chapter Three
Proposed System

3.1 Introduction

SA, also referred to as opinion mining, is an approach of NLP. It involves the use of data mining, machine learning, information retrieval and artificial intelligence to mine text for subjective information and sentiment to identify positive, negative or neutral opinions, emotions and evaluations. This chapter describes the architecture and methodology of our SA system. This chapter also describes all stages of the proposed system. The algorithms and techniques used at each stage are explained. Each stage of the application of the SA system includes the necessary steps and algorithms and depends on the results of the previous stage. The implementation details are illustrated in this chapter.

3.2 Proposed System

In this section, we describe the detailed components of the proposed system for SA. As shown in Figure 3.1, the proposed system operates in two phases, namely, training and test phases. The purpose of the training phase is to build the classification model to distinguish between positive and negative documents based on the input labelled documents collected. In the test phase, the trained classification model assigns positive or negative labels to the new unlabelled documents. The system comprises five steps: preprocessing, feature extraction, feature selection, classification model for SA and system evaluation.

In the first step, the main objective is to use NLP techniques to process the input document text and make it suitable for accurate feature extraction in the next step. In the second step, several features need to be extracted to represent the input document text. In this step, we present the different types of features used in the proposed system by using TF-IDF. In the third step, we use the chi-square test as a feature selection technique. The goal is to select a subset of features from the set of original features that offers unchanged definition with no loss of information. In the fourth step, we build a classification model that can differentiate efficiently between positive and negative labelled documents in the training phase. Several neural network algorithms can be used in building such a system. In the proposed system, we implement an MLP classifier. MLP is commonly used and have great success in text classification problems. Finally, in the fifth step, system performance is evaluated using several metrics.

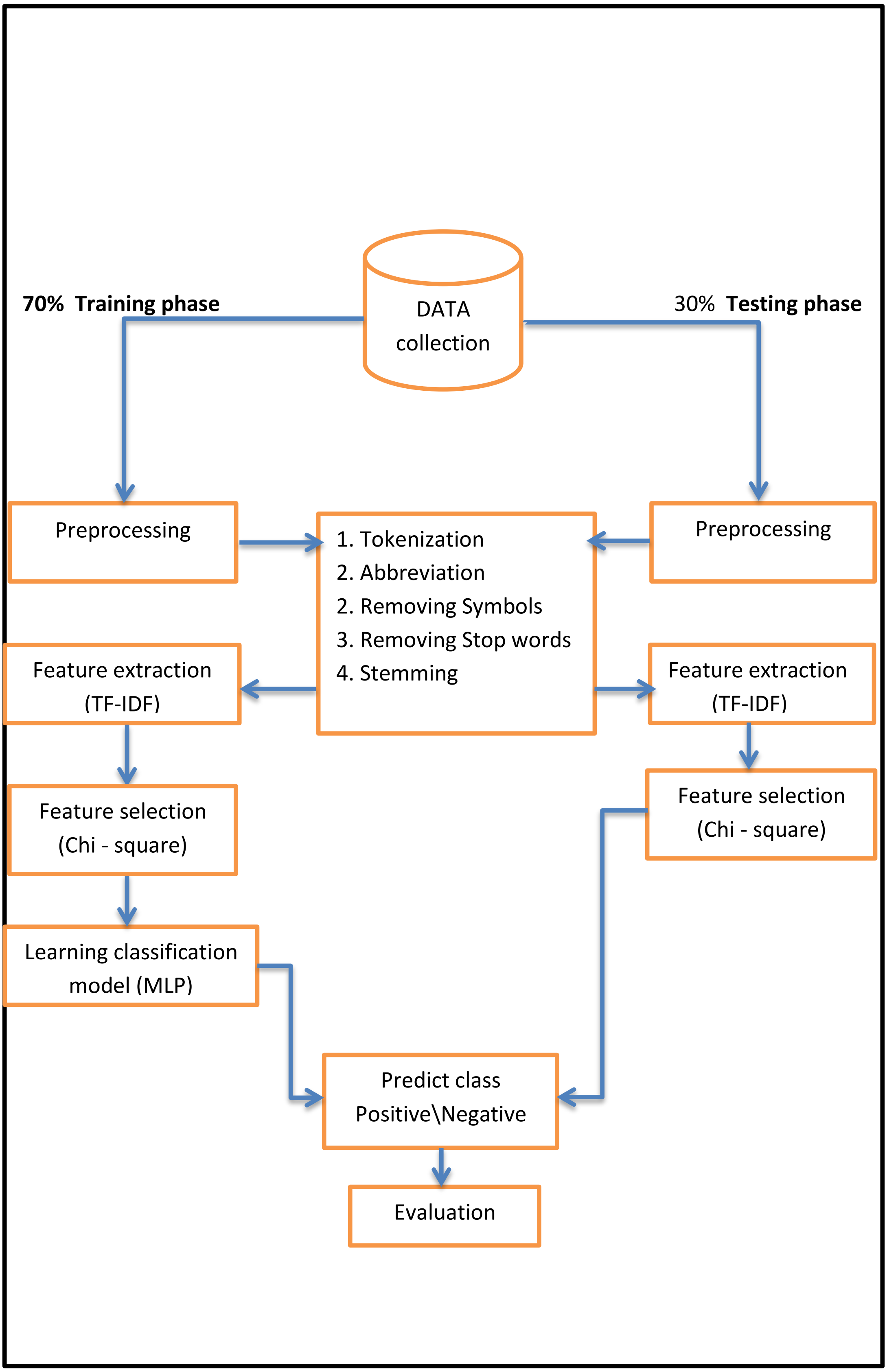
70% Training phase
DATA collection
30% Testing phase
Preprocessing
1. Tokenization
2. Abbreviation
2. Removing Symbols
3. Removing Stop words
4. Stemming
Preprocessing
Feature extraction (TF-IDF)
Feature extraction (TF-IDF)
Feature selection (Chi - square)
Feature selection (Chi - square)
Learning classification model (MLP)
Predict class Positive\Negative
Evaluation

Figure 3.1:The Proposed SA System

3.3 Preprocessing

Preprocessing is the first step of the proposed system. Text preprocessing is an important stage in the SA of the data on Twitter and movie reviews. A tweet is a short message full of noise, such as irrelevant symbols, misspellings, emoticons, stop words and slang words; some of these characteristics apply to movie review data. Such noisy characteristics typically affect the performance of SA system. Thus, in this work, preprocessing methods are applied prior to feature extraction and selection using specific machine learning algorithms. The most general view for preprocessing is shown in Figure (3.2).

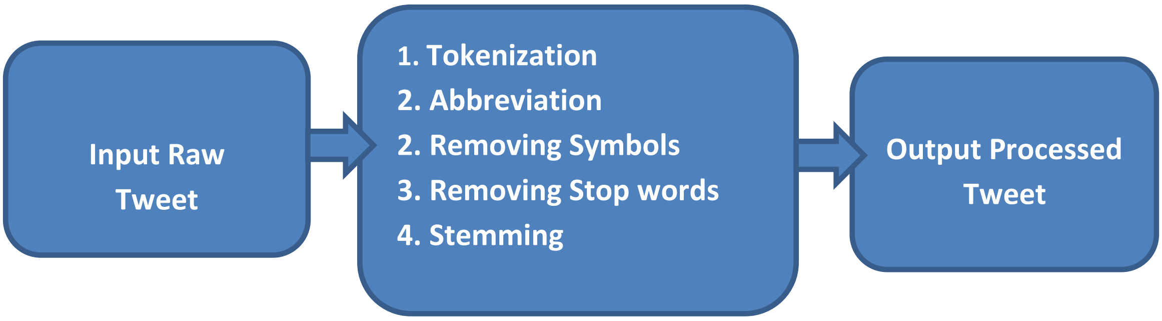

Figure 3.2: Overview of data preprocessing

After collecting many raw datasets, it has been process the document text. A number of preprocessing steps is applied to standardise the dataset and filter the noise from the data to reduce its size. We initially perform preprocessing on all documents.

3.3.1 Tokenization

This process is important for data preprocessing. In tokenisation, the text document is separated into tokens using a space to split each word from another. Tokens might be numbers, symbols and words. Herein, to tokenise the words, the NLTK tokenisation package is used. Selecting a tokeniser depends on the characteristic of the data and the language. In this work, a tokenising method is created to tokenise the words using the document tokeniser module for processing document terms. The algorithm for word tokenisation using the document tokeniser is shown in Algorithm (3.1).

Algorithm 3.1: Word Tokenisation

```
Input:  Documents
Output: Tokenised Word
For each   Document
        Add the state of tokenization
        Tokenise the word passed to document tokeniser method and
        append the tokenised word
Return Tokenised Word
```

In table (3.1) shows the document after processing by tokenisation.

Table 3.1: Example of Word Tokenisation

no	Original Document	Word Tokenisation
1	is so sad for my APL friends	['is', 'so', 'sad', 'for', 'my', 'APL', 'friends']
2	Fifa ticketing just times out. Seems like my 2010 ticket chances are getting less and less	['Fifa', 'ticketing', ' just', 'times', 'out', ' Seems', ' like', 'my', '2010', 'ticket', 'chances', 'are', 'getting', 'less', ' and', 'less']

3.3.2 Removing Symbols and Repeated Characters

We also identify several features that could affect the experiment results, and they include ‘http://t.co/NxSE1HYTHj’ or (: \ | [ ] ; : {} - + ( ) < > ? ! @ # % *,). Many documents contain nonalphabetic symbols, such as ‘@’ or ‘#’, as well as active web links. The word immediately following the ‘@’ symbol indicates a username, which we filter out entirely. The username is deemed to add no sentiment value to the text.

To express their strong feelings, people often use words with multiple characters. For example, in ‘LOVEEEEE’, the many ‘E’ characters used are unnecessary and should therefore be eliminated. The algorithm for removing document symbols and repeated characters is demonstrated in Algorithm (3.2).

Algorithm 3.2: Removing Document Symbols

```
Input: Document
Output: clean_document

clean_document = empty list [ ]
For each word in Document
    Convert the word characters to lowercase alphabet
    Remove URL using regular expression.
    Remove special characters
    Remove the numbers.
    Look for repetitions of two or more characters and replace with
    the character itself.
    Append word in clean_document
Return clean_document
```

Table 3.2:Example of Removing Symbols and Repeated Character

No	Input document without Removing of symbols and repeated characters	Output with Removing of symbols and repeated characters
1	(must i say more?)	must i say more
2	A dog riding the bicycle http://bit.ly/gvMzD	A dog riding the bicycle
3	my life is very lazzzzyyyy!!!!!!!	my life is very lazy

Table (3.2) shows the documents after removing the symbols and repeated characters.

3.3.3 Abbreviation

The number of abbreviations that people use on popular social media channels is astonishing. People regularly use and create new abbreviations on Twitter because of the 140-character limit. These platforms are based on short communication. Thus, people rely on abbreviations, especially for common phrases.

The abbreviations dictionary has been developed manually and contains a large number of abbreviations. The abbreviations dictionary helps expand the tweets and improve the overall sentiment score. The algorithm for abbreviations is shown in Algorithm (3.3).

Algorithm3.3: Abbreviations Dictionary

```
Input: word
Output: word with full name
Slang = list of words Abbreviation [  ]
    if "word" found in Slang list then
        return the word with full name
    else
        return word
    End
```

3.3.4 Removing Stop Words

In processing natural language, some of the words with high occurrence in documents are stop words (e.g. ‘on’, ’the’, ‘am’ and ‘are’). They only have little emotional meaning and do not affect the opinion and sentiment score when applied to lexical resources.

A predefined list of stop words is used to identify and remove these words from any document. The list contains all possible stop words in the English language. The list also contains many categories, namely, adverbs, conditional pronouns, interrogative pronouns, prepositions, referral names, relative pronouns, transformative verbs and verbal pronouns. Therefore, using stop word removal in the SA process is necessary in any system. Removing stop words from a document leads to improved accuracy. The algorithm for removing stop words is demonstrated in Algorithm (3.4).

Algorithm 3.4: Removing Stop Words

```
Input: Document
Output: Document without stop words

Stop_list = List of stop word [ ]
For each word in document
If word found in stop_list
    Remove();
Else
    Return word
End
```

The results obtained after the removal of stop words from a document are shown in Table (3.3).

Table 3.3: Example of Removing Stop Words

No	Document Tokens without filtering stop words	Document Tokens with filtering Stop Words
1	[**'is', 'so',** 'sad', **'for', 'my'**, **'APL',** 'friend']	['sad', 'friends']
2	['Fifa', 'ticketing', ' **just'**, **'times', 'out',** ' Seems', ' like', **'my', '2010'**, 'ticket', 'chances', **'are',** 'getting', 'less', ' **and'**, 'less']	['Fifa', 'ticketing',' Seems', ' like', 'ticket', 'chances', 'getting', 'less']

3.3.5 Stemming

The stemming of words produce a normalised form of a word in the text. Stemming is a technique that obtains the root (base) of a word in a text. It normalises the word by removing the suffix from the word, thereby providing the root meaning of the word. Many stemming algorithms are openly available for use in word stemming. In this approach of data preprocessing, the Porter Stemmer algorithm is used for stripping the suffixes from the words to retrieve the appropriate meaning from the text. Porter Stemmer algorithm is produces the best output as compared to other stemmers and it has less error rate [57,55].

To implement this functionality for stemming words, we use the Porter Stemmer algorithm available in the Python NLTK stem package. Porter Stemmer stems the word character by character, removes the suffix and provides the base meaning of the word. During the stemming process in this system, the word is stemmed, and the root meaning of the word is returned. To achieve accuracy in SA, we only stem the words whose length is greater than two. Words such as ‘a’, ’is’ and ‘OH’ are not taken into consideration when applied to the sentiment dictionary for obtaining word polarity. The algorithm for word stemming is demonstrated in Algorithm (3.5).

Algorithm 3.5: Porter Stemmer

```
Input: words
Output: stemmed_words

Stemmed_word = empty list [ ]
For each word in words
    If length of word greater than 2
        Method call for stemming the word using Porter Stemmer object.
        Append Stemmed_word
Return Stemmed_word
```

In this algorithm, the word whose length is greater than two is stemmed, and the word in the variable type list then will return stemmed word list. The returned stemmed word comes in a generic form and can be used in the further steps of the NLP task. Furthermore, word stemming and lemmatising provide a common base form of a word by removing the suffix from the word to provide the dictionary meaning of the word. For example, the word ‘cars’ becomes ‘car’ by stemming the letter ‘s’. In addition, words such as ‘consult’, ‘consultant’, ‘consulting’, ‘consultantative’ and ‘consultants’ are stemmed to the single word ‘consult’, which is the generic form of the words. This approach provides a generic sentiment score and helps to evaluate accurate sentiment polarity for the textual document. The complete result obtained using this algorithm is shown in Table (3.4).

Table 3.4: Example of Word Stemming

no	Document Tokenisation Without Stemming	Document Tokenisation With Stemming
1	['sad', '**friends**']	['sad', 'friend']
2	['Fifa', **'ticketing'**,' **Seems'**, ' like', 'ticket', **'chances'**, **'getting'**, 'less']	['Fifa', 'ticket',' Seem', ' like', 'ticket', 'chance', 'get ', 'less']

3.4 Feature Extraction Using TF-IDF

For each preprocessed document, a feature vector is constructed. For the purpose of representing a document as an array of features, the document should be transformed from full text to document vector. The TF-IDF model is utilised to compute the frequency of each word or extract features from documents. Thus, a collection of text documents might be presented via a matrix which displays the association of each word within the document via weights. Each feature related to a weight decides the significance of the feature in the document, such as document D, which is defined as (w1, w2,w3 , w4 , ...,wn), where wi is the weight of feature i in document D. In Algorithm (3.6), the weight vector representation for the document is described. Document representations have TF-IDF weight feature vectors. In this algorithm, for each preprocessed document, a vector of a feature is constructed, and the TF is computed, representing the number of times each word appears in each document.

Algorithm 3.6: Document Representations with TF-IDF Weight Feature Vector

```
Input: Stem-List
Output:  Feature vector
Begin
Step1: TF computing
    For each words in Stem-List
        Compute the TF using equation (2.1)
    Next
Return TF
Step2: IDF computing
    For each word in Stem -List
        Compute the IDF using equation (2.2)
    Next
Return IDF
Step3:TF-IDF computing
    For each word in Stem -List
        Compute TF-IDF with the use of equation (2.3)
    Next
Return Feature vector
End
```

3.5 Feature Selection Using Chi Square

The next important step in text classification after preprocessing and feature extraction is feature selection. It is also known as attribute selection. The main goal of feature selection is to select a subset of features from the set of original features by removing the irrelevant and redundant features from the original dataset. The accuracy of SA depends not only on the classification algorithm but also on the feature selection method. The selection of irrelevant and inappropriate features may lead to inaccurate results. The solution to this problem is feature selection to improve the efficiency and accuracy of SA. Feature selection reduces the dimensionality of a dataset, increases the learning accuracy and improves result comprehensibility.

Even after preprocessing and feature extraction, the number of features remains high. To address this issue, several methods have been developed to reduce the input space by selecting a subset of features that might lead to enhanced classification. We use chi-square statistics (X^2), as has been explained as outlined in chapter 2 (section 2.7.1.1.1), which is one of the most widely used metrics for feature selection. Algorithm (3.7) illustrates the steps of feature selection process by using Chi Square.

Algorithm 3.7: Chi Square Algorithm

```
Input:   Feature vector
Output:  Best feature
For each word in feature vector
    Step 1
        A- Calculate numbers of positive instances that contain feature X
        B- Calculate numbers of negative instances that contain feature X
        C - Calculate numbers of positive instances that do not contain feature X
        D-Calculate numbers of negative instances that do not contain feature X
    Step 2
        Applying equation (2.4)
Return Best feature
```

3.6 Classification of Data

Several types of classifiers are used for machine learning techniques to identify the sentiments in data. Machine learning offers a solution to the sentiment classification problem. We use a neural network classification method based on polarity that uses a set of positive and negative documents in SA. Polarity is given by the ratio of the probability of a word appearing in a set of positive or negative statements which makes the word positive or negative.

Neural networks have emerged as an important tool for document classification. The feature weights of a neural network are loaded into input nodes. The activation of the nodes is propagated forward through the network, and the final values of the output nodes determine the categorisation decisions. Neural networks are trained by backpropagation, in which the training documents are loaded into the input nodes. If a misclassification occurs, then the error is propagated back through the network, the link weights are modified to minimise the error.

The process of a neural network contains two stages, namely, training and testing. The general neural network design process comprises seven primary steps:

- Collect data representing features
- Create the network
- Form the network
- Initialise the weights and biases
- Train the network
- Validate the network (test the network)
- Evaluation

3.6.1 MLP Classifier

One of the popular neural networks for SA application is the MLP network. The network builds its predictive model using a set of data samples. The MLP classifier network consists of inputs layers, one or more hidden layers and an output layer. The connections between the layers are called weights W, which are normally defined between 0 and 1. The advantages of using MLP in our proposed system is mainly minimising the error function. The minimisation of this error leads to the optimal values of the weights. In addition to that MLP is commonly used and have great success in text classification problems

Classification is identifying to which category an object belongs. The plan is to build a classifier with a set of training data and labels. In this case, a classifier that is trained on our ‘positive’ or ‘negative’ labelled document corpus is constructed. The classifier can label future documents as ‘positive’ or ‘negative’ on the basis of the document’s attributes or features. The data has been split into training and test sets. We also maintain separate positive and negative prelabelled datasets for checking their classification in the test set. Then, the training data are fed to the MLP classifier, which learns to classify and use backpropagation to modify the weights.

In this thesis, the MLP classifier based on a feedforward ANN in the current implementation uses backpropagation to learn the model and modify the weights. It has several layers, and each layer has a weight matrix W, a bias vector b and an output vector. The inputs (x_1,x_2 ,x_3..x_m) and connection weights (w_1,w_2,w_3..w_m) are typically real values. The output from the neural network is a single value, which we pass through the nonlinear ReLU activation function to squish it in the range [0, 1]. The ReLU function is defined by the output from the neural network and provides the probability 0 (negative) and 1 (positive).The general steps for MLP are as follows:

- Step 1: **Initialisation.** Set the initial values of the weights randomly. The weights should be adjusted to minimise the error. If the output is correct, then the weights are unchanged.
- Step 2: **Activation**. Calculate the actual outputs of the neurons. This stage works under the hidden layer.

$$\boldsymbol{y(p) = RELU \sum_{i=1}^{n} xi(p)\ .wi(p)} \ldots\ldots\ldots\ldots..(\mathbf{2.13})$$

- Step 3: **Weight training**. If the output is correct, then the weights are unchanged. However, if the output is incorrect, then the weights w_i must be changed.
- Step 4: **Iteration**. Increase the iteration and repeat the process until the selected error criterion is satisfied.

The algorithm for the MLP classifier is demonstrated in Algorithm (3.8).

Algorithm 3.8: Multilayer Perceptron

```
Input : X  = ( $x_1,x_2,x_3,\dots,x_n$), where x ∈ feature vector, n is no of input
            y = actual class
Output : predicate (positive(1)) or (negative (0)).

Weight: W = ($w_1,w_2,w_3,\dots,w_m$), initial random
Repeat
    For each training vector pair (x, y)
        Compute the output $\boldsymbol{y'}$ when x is the input
            $y' = \sum_{i=1}^{n} xi.wi + b$   (where b is bias)
            If y ≠ y'  then
                update weight w´ according to
                w´ (i) =w(i) + α (y-y') x  (where α is learning rate)
            End if
  Until y'= y for all training vector pairs
```

3.7 Performance Evaluation Methods

Statistical analysis is important in measuring the performance of SA systems. The accuracy of SA represents the basic parameter to check system performance. Whilst SA is considered a classification problem, system performance could be evaluated in terms of many parameters. Four parameters are used to check the performance of the proposed system: accuracy, precision, recall and F1 score, as has been explained as outlined in chapter 2 (section 2.9).

Chapter Four

Results and Discussion

4.1 Introduction

Social media have garnered immense popularity. Data from social networking services (SNSs) can be used for a number of objectives, such as prediction or SA. SA has emerged as an interesting research field. It combines NLP techniques with data mining approaches for building such systems. Twitter and movie review platforms are types of SNS with large data and user posting. However, handling such a large amount of unstructured data is difficult, especially in the context of Twitter. Machine learning is required for handling such large data.

This chapter presents an analysis and evaluation of the results obtained by the proposed system discussed in the previous chapters. The proposed system is tested to determine its effectiveness. In addition, the chapter presents the comparisons between the results of the proposed system and those presented in the literature on SA. The proposed system is implemented to detect positive and negative opinions in four stages, namely, preprocessing, feature extraction(TF_IDF technique) , feature selection(chi-square technique) and classification model for SA (MLP). In the classification phase, the MLP classifier is implemented to achieve accurate results. The implementation of the proposed SA system is tested on Twitter and movie review datasets. In table 4.1 summarises the specifications of the environment used to build and run the proposed system.

Table 4.1: Specifications of Environment for the Proposed System

NO.	Operating System	Windows 7 Ultimate
1	System type	64 bit operating system
2	Memory	4.00 GB
3	Processor	Intel Core i7
4	Development tool	Python 3.4
5	Used Libraries	NLTK

4.2 Tools

The system is written in Python, a powerful programming language. The data structure and object-oriented programming in Python provide programmers with an effective . Its interactive interpreter permits the direct coding of a program and allows access to many standard libraries and resources that are freely available on the web to fulfil the requirements for application development. The most widely used library in the program is the natural language toolkit (NLTK), which is written in Python. Therefore, it is easy to use within Python. Another library is the Python Twitter Tools, which offers a simple way to interact with the Twitter API from Python. It is also the most suitable language for scripting and application development because of its sophisticated syntax and dynamic tying; it is equally interesting to use in processing linguistic data. During the course of this research, the linguistic data are processed on Python version 3.4 using NLTK, which is most compatible with this version of Python.

4.3 Dataset Description

For the purpose of training and testing of the system, two different dataset corpus have been used. The main Objective for using two data sets in this thesis, was to show the difficulties of guessing the sentiment in short and often ungrammatical English texts as in Twitter dataset, as opposed to a relatively long and well established English texts as in movie reviews dataset. These two dataset corpus are described below:

4.3.1 Twitter Dataset

In this research, machine learning is used to classify sentiments and Twitter datasets from Kaggle[1] which were crawled and labelled as positive/negative. The data are in the form of CSV files with 'tweets' and their corresponding sentiments. The training dataset is a CSV file of type item_id, sentiment and sentiment text, where the item_id is a unique integer identifying the tweet, the sentiment is either 1 (positive) or 0 (negative) and the sentiment text is the tweet. Similarly, the test dataset is a CSV file of type item_id and sentiment text. The dataset is a mixture of words, emoticons, symbols, URLs and references to people as seen on Twitter. Words and emoticons contribute to predicting sentiments, whereas URLs and references to people do not. This dataset contains 1,494 tweets split into 749 negative tweets and 745 positive tweets, and the data are split into 70% training dataset and 30% test dataset. The length factor of feature is 3008 for the Twitter dataset.

4.3.2 Movie Reviews Dataset

The dataset of movie reviews was collected from the archives of the IMDb web portal[2] and all written in English languages. The structure of movie reviews dataset is the same structure of twitter dataset previously mentioned. In this dataset, we also apply all the preprocessing steps applied to the Twitter dataset. This dataset contains 400 document split into 200 negative document and 200 positive document, and the data are split into 70% training dataset and 30% test dataset. The length factor of feature is 7,511 for the dataset of movie reviews.

4.4 Configuration of Classifier

The configuration of fully connected layers in terms of number of layers and nodes is tested on several models beginning with two layers and ending with four layers. The maximum dimension of each layer is 2048 nodes, and the minimum one is 16 nodes**.** Initially, the model is tested on the Twitter dataset without feature selection, and then the model is tested with feature selection. The parameters used in the ANN are shown in Table 4.2.

[1] https://www.kaggle.com/
[2] http://reviews.imdb.com/Review

Table 4.2: Parameters of Artificial Neural Network

parameter	value	Details
Hidden layer	2-4 hidden layers	The element represents the number of neurons in the hidden layer.
Activation	ReLU	We use ReLU activation function for the hidden layer.
Alpha	0.0001	Penalty regularisation term
Learning rate	0.001	The initial learning rate used. It controls the step size in updating the weights.
Max iteration	100 iteration	Maximum number of iterations. The solver iterates until convergence (determined by 'tol') or this number of iterations.
Tolerance	1e-4	Tolerance for the optimisation. When the loss or score is not improving by at least 'tol' for 'n_iter_no_change' consecutive iterations, unless 'learning_rate' is set to 'adaptive', convergence is considered to be reached and training stops.
Random state	Select number from [0 – 1]	If int, random_state is the seed used by the random number generator.

In table (4.2) describe all parameter that uses in MLP classifier. we used 2-4 hidden layer. We obtained the best performance for the system when using three hidden layers on Twitter dataset and two hidden layers in movie reviews dataset. Also,we used RELU activation function ,the best system performance we got when we used the RELU activation function. Maximum number of iterations was 200 iteration until reaching optimum values for weights. The rest of the parameters (Alpha, Learning rate, Tolerance, Random state) were the best performance of the system according to the values chosen in the above table.

4.4.1 Applying the System Using All Features

Firstly, the system is trained by using the Twitter dataset without feature selection. In the MLP classifier, backpropagation is used to modify the weights, and the activation function is the ReLU function which assigns '1' as positive and '0' as negative. The learning rate starts with 0.001 and can be changed in the program (alpha = 0.0001, maximum iterations = 100, tolerance = 1e-4).

In **Table 4.3**, each row represents the complete model configurations and results obtained from this model, and the columns represent the following. The first column is the model number representing the sequence of the model in the experiment. The second column represents the number of fully connected layers in the model in the two layers. The third column represents the number of nodes in each layer and is between (2048–16). The fourth column represents the 'accuracy of the model', and it is equal to the number of accurate predictions made by the model over all types of predictions. The fifth column represents the 'precision' of the model, and it is equal to the number of true positives divided by the total number of elements labelled as belonging to the positive class. The sixth column represents the 'recall' of the model, and it is equal to the number of true positives divided by the total number of elements that actually belong to the positive class. The seventh column represents the 'F1-score' of the model, and it is equal to the harmonic average of precision and recall.

Table 4.3: Two-Layer Model Using All Features

Model No.	No. of Layers	No. of Nodes	Accuracy	Precision	Recall	F-score
1	2	1024-1024	0.6851	0.69	0.69	0.68
2	2	512-512	0.6620	0.70	0.66	0.65
3	2	256-256	0.6782	0.69	0.68	0.68
4	**2**	**128-128**	**0.6597**	**0.67**	**0.66**	**0.66**
5	2	64-64	0.6643	0.66	0.66	0.66
6	2	32-32	0.6898	0.70	0.69	0.68
7	2	16-16	0.6620	0.68	0.66	0.66
8	2	2048-1024	0.6620	0.67	0.66	0.66
9	2	1024-512	0.6782	0.68	0.68	0.68
10	2	512-256	0.6851	0.69	0.69	0.69
11	**2**	**256-128**	**0.6921**	**0.69**	**0.69**	**0.69**
12	2	128-64	0.6712	0.67	0.67	0.67

13	2	64-32	0.6666	0.67	0.67	0.67
14	2	32-16	0.6643	0.67	0.66	0.66

The results in **Table 4.3** show that the values of accuracy, precision, recall and F1-score are convergent, indicating the presence of stability in the network. The worst accuracy is **0.6597** in model 4, and the best accuracy is **0.6921** in model 11. The accuracy is not high because the feature selection technique did not applied.

Figure 4.1 shows the evaluation metrics of the best results when using two layers without feature selection.

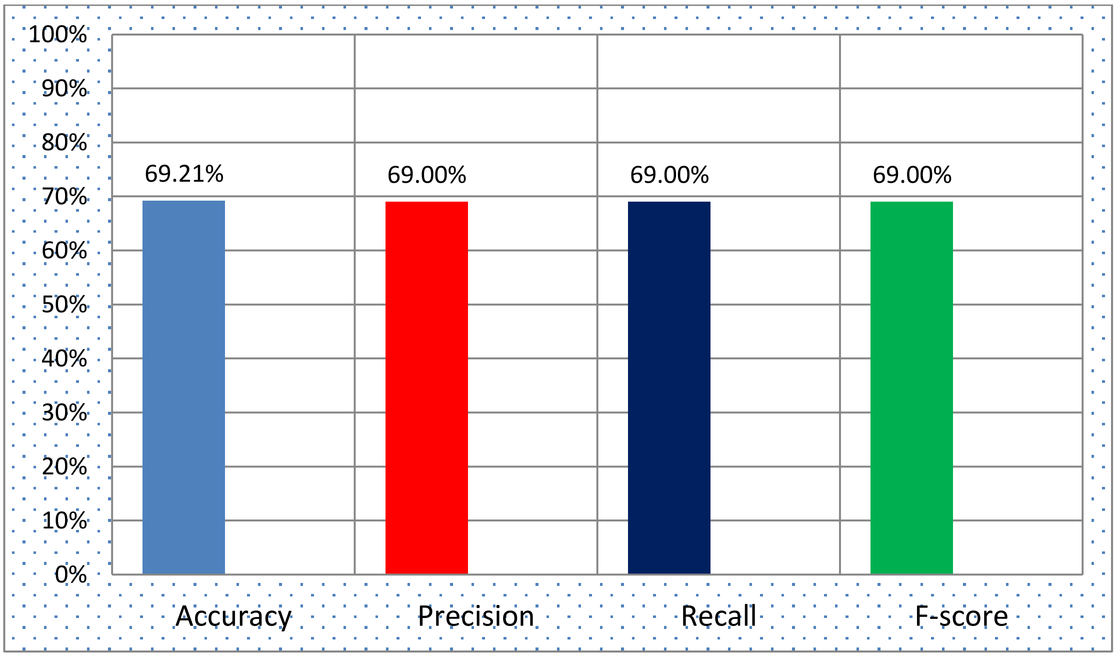

Figure 4.1: Evaluation Metrics for Two Layers Using All Features for Model 11

In **Table 4.4**, the rows represent a set of training rounds of a set of models. The second column represents the number of fully connected layers in the model, which comprises three layers.

Table 4.4: Three-Layer Model Using All Features

No.	No. of Layers	No. of Nodes	Accuracy	Precision	Recall	F-score
1	3	2048-2048-2048	0.6643	0.67	0.66	0.66
2	3	1024-1024-1024	0.6643	0.67	066	0.66
3	3	512-512-512	0.6620	0.67	0.66	0.66
4	**3**	**256-256-256**	**0.6875**	**0.69**	**0.69**	**0.69**
5	3	128-128-128	0.6736	0.68	0.67	0.67
6	3	64-64-64	0.6712	0.68	0.67	0.67
7	3	32-32-32	0.6643	0.70	0.66	0.65
8	**3**	**16-16-16**	**0.6481**	**0.65**	**0.65**	**0.64**
9	3	2048-1024-512	0.6666	0.67	0.67	0.67
10	3	1024-512-256	0.6759	0.69	0.68	0.67
11	3	512-256-128	0.6851	0.69	0.69	0.68
12	3	256-128-64	0.6851	0.69	0.69	0.68
13	3	128-64-32	0.6620	0.67	0.66	0.66
14	3	64-32-16	0.6736	0.68	0.67	0.67

The results in **Table 4.4** show that the values of accuracy, precision, recall and F1-score are convergent, indicating the presence of stability in the network. The worst accuracy is **0.6481** in model 8, and the best accuracy is **0.6875** in model 4.

Figure 4.2 shows the evaluation metrics of the best results when using three layers without feature selection.

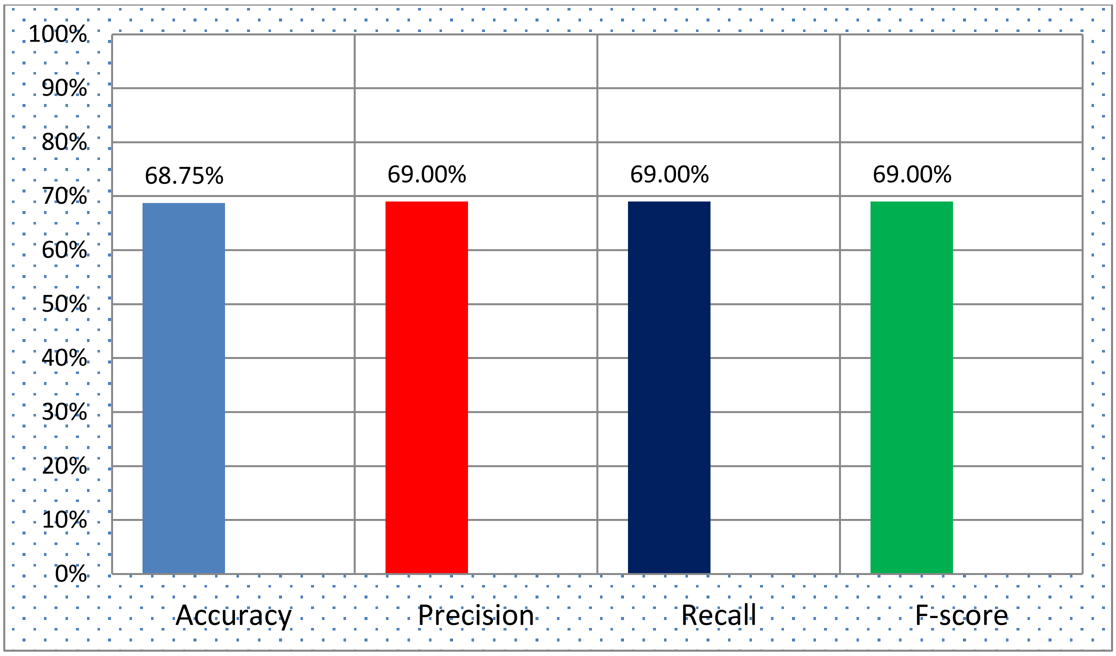

Figure 4.2: Evaluation Metrics for Three Layers Using All Features for Model 4

In **Table 4.5**, the rows represent a set of training rounds for each model. The second column represents the number of fully connected layers in the model, which comprises four layers.

Table 4.5: Four-Layer Model Using All Features

No.	No. of Layers	No. of Nodes	Accuracy	Precision	Recall	F-score
1	**4**	**2048-2048 2048-2048**	**0.6527**	**0.65**	**0.65**	**0.65**
2	4	1024-1024-1024-1024	0.6712	0.71	0.67	0.66
3	**4**	**512-512-512-512**	**0.6805**	**0.68**	**0.68**	**0.68**
4	4	256-256-256-256	0.6759	0.68	0.68	0.68
5	4	128-128-128-128	0.6550	0.66	0.66	0.66
6	4	64-64-64-64	0.6666	0.67	0.67	0.67
7	4	32-32-32-32	0.6550	0.67	0.66	0.65

8	4	16-16-16-16	0.6550	0.66	0.66	0.65
9	4	2048-1024-512-256	0.6643	0.66	0.66	0.66
10	4	1024-512-256-128	0.6574	0.66	0.66	0.66
11	4	512-256-128-64	0.6597	0.67	0.66	0.65
12	4	256-128-64-32	0.6736	0.68	0.67	0.67
13	4	128-64-32-16	0.6527	0.67	0.65	0.65

The results in **Table 4.5** show that the values of accuracy, precision, recall and F1-score are convergent, indicating the presence of stability in the network. The worst accuracy is **0.6527** in model 1, and the best accuracy is **0.6805** in model 3, which thus requires little training time and we need to apply feature selection technique to improve system performance.

Figure 4.3 shows the evaluation metrics of the best results when using four layers without feature selection.

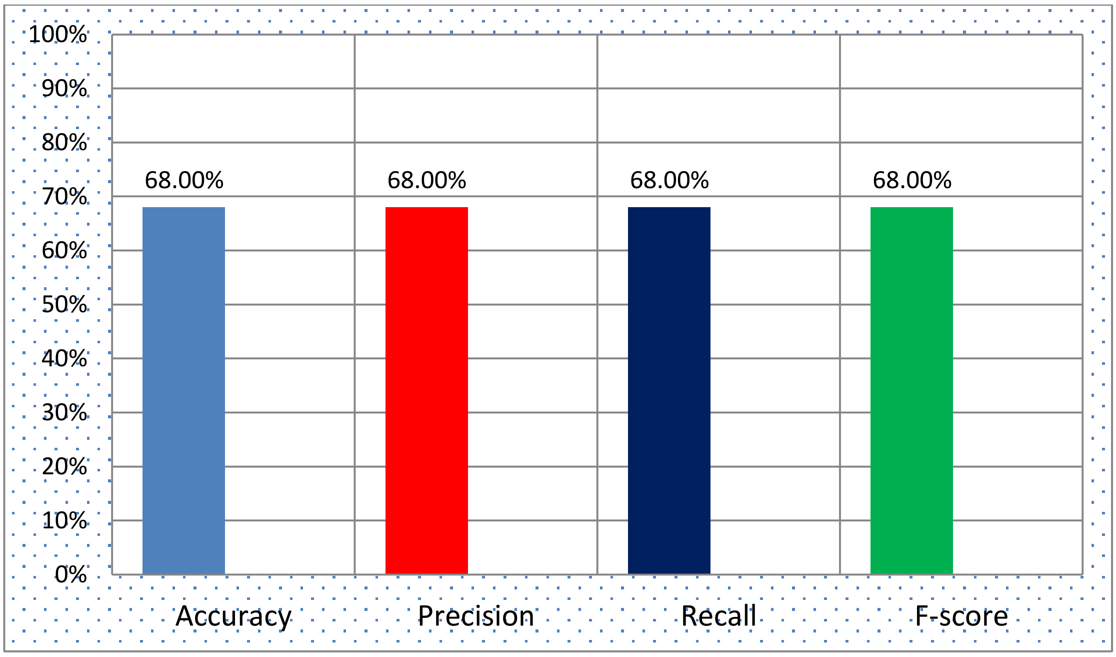

Figure 4.3: Evaluation Metrics for Four Layers Using All Features for Model 3

The results in **Tables 4.3–4.5** show that the best configurations in the two layers are as follows. The first layer consists of 256 nodes, and the second layer is made up of 128 nodes because the model has the highest accuracy using these configurations. However, the accuracy and other performance parameters of the system are still low. Thus, we need to apply the feature selection to increase accuracy and other performance parameters of the system (as shown in the next section).

4.4.2 Applying the System using Feature Selection Algorithm

After testing all configurations of the classifier component (fully connected layers) without feature selection, we test the model on the Twitter dataset with feature selection. In the MLP classifier, we use backpropagation to modify the weights. In the activation function, we use the ReLU function which assigns ‘1’ as positive and ‘0’ as negative. The learning_rate_init = 0.001 and can be changed in the program, in addition to alpha = 0.0001, max_iter = 100 and tol = 1e-4.

In **Table 4.6**, the rows represent a set of training rounds of a set of models. The columns are as follows. The first column is the model number, which represents the sequence of the model in the experiment. The second column represents the number of fully connected layers in the model, which comprises two layers. The third column represents the number of nodes in each layer and is approximately 2048-16. The fourth, fifth, sixth and seventh columns represent the accuracy, precision, recall and F1-score of each model, respectively. Finally, the eighth column represent the best feature selection.

Table 4.6: Two-Layer Model Using Feature Selection

No.	No. of Layers	No. of Nodes	Accuracy	Precision	Recall	F-score	Best feature
1	2	1024-1024	0.8333	0.84	0.83	0.83	506
2	2	512-512	0.8333	0.84	0.83	0.83	448
3	2	256-256	0.8333	0.84	0.83	0.83	570

4	2	128-128	0.8379	0.84	0.83	0.83	545
5	2	64-64	0.8425	0.85	0.84	0.84	542
6	2	32-32	0.8425	0.85	0.84	0.84	511
7	2	16-16	0.8402	0.85	0.84	0.84	563
8	**2**	**2048-1024**	**0.8310**	**0.84**	**0.83**	**0.83**	**553**
9	2	1024-512	0.8379	0.84	0.84	0.84	539
10	2	512-256	0.8449	0.85	0.84	0.84	574
11	2	256-128	0.8333	0.83	0.83	0.83	510
12	2	128-64	0.8402	0.84	0.84	0.84	590
13	2	64-32	0.8356	0.84	0.84	0.83	470
14	**2**	**32-16**	**0.8449**	**0.85**	**0.84**	**0.84**	561

The results in **Table 4.6** shown that there is an Significant improvement in system performance when feature selection operation has been done for all feature vectors by using chi square technique. The results show that the values of accuracy, precision, recall and F1-score are convergent, indicating the presence of stability in the network. The worst accuracy is equal to **0.8310** in model 8**,** and the best accuracy is equal to **0.8449** in model 14.

Figure 4.4 shows the evaluation metrics of the best results when using two layers with feature selection.

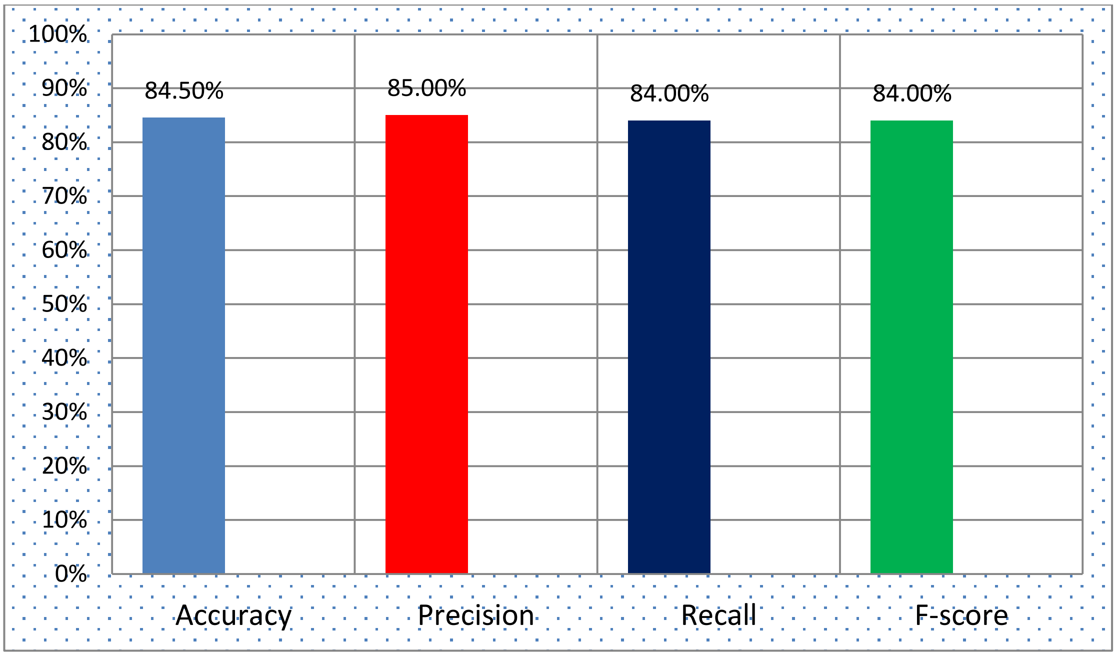

Figure 4.4: Evaluation Metrics for Two Layers Using Feature Selection for Model 14

In **Table 4.7**, the rows represent a set of training rounds of a set of models. The second column represents the number of fully connected layers in the model, which has three layers.

Table 4.7: Three-Layer Model Using Feature Selection

No.	No. of Layers	No. of Nodes	Accuracy	Precision	Recall	F-score	Best feature
1	**3**	**2048-2048-2048**	**0.8310**	**0.83**	**0.83**	**0.83**	564
2	3	1024-1024-1024	0.8310	0.83	0.83	0.83	581
3	3	256-256-256	0.8333	0.84	0.83	0.83	510
4	3	128-128-128	0.8379	0.84	0.84	0.84	545
5	3	64-64-64	0.8379	0.84	0.84	0.84	470
6	3	32-32-32	0.8402	0.85	0.84	0.84	590
7	**3**	**16-16-16**	**0.8495**	**0.86**	**0.85**	**0.85**	595
8	3	2048-1024-512	0.8310	0.84	0.83	0.83	579
9	3	1024-512-256	0.8333	0.84	0.83	0.83	572

10	3	512-256-128	0.8356	0.84	0.84	0.83	521
11	3	256-128-64	0.8379	0.84	0.84	0.84	586
12	3	128-64-32	0.8356	0.84	0.84	0.83	473
13	3	64-32-16	0.8402	0.85	0.84	0.84	570

The results in **Table 4.7**, we note that the best performance for our proposed system is obtained when we use three hidden layers with the feature selection technique. The results in **Table 4.7** show that the values of accuracy, precision, recall and F1-score are convergent, indicating the presence of stability in the network. The worst accuracy is **0.8310** in model 1 and the best accuracy is **0.8495** in model 7.

Figure 4.5 shows the evaluation metrics of the best results when using three layers with feature selection.

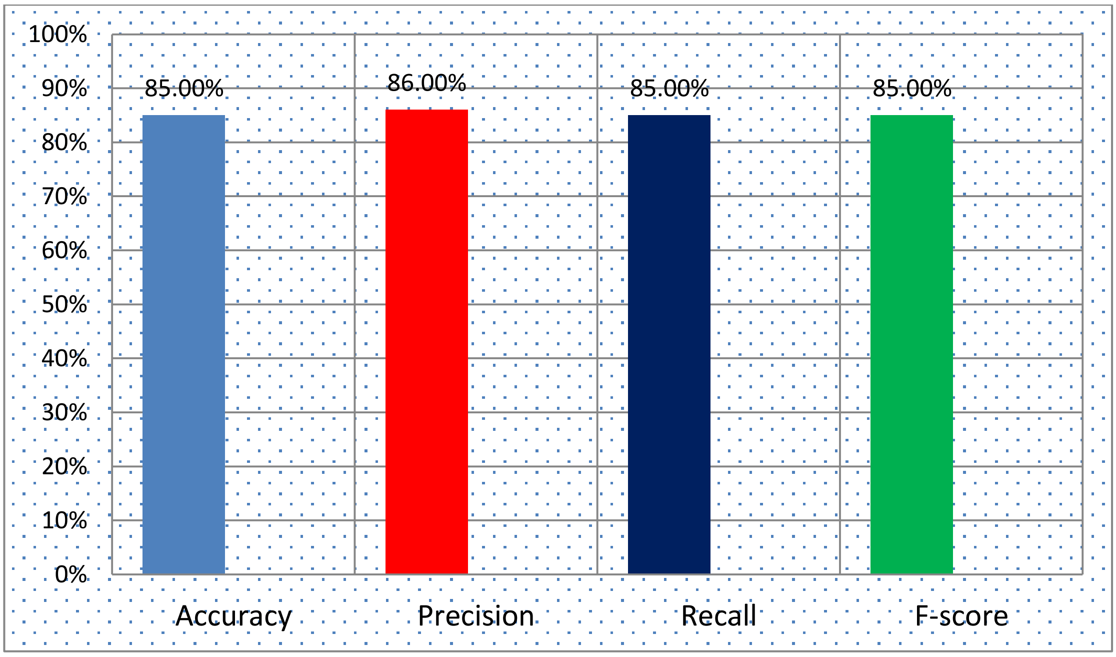

Figure 4.5: Evaluation Metrics for Three Layers Using Feature Selection for Model 8

In **Table 4.8**, the rows represent a set of training rounds of a set of models. The second column represents the number of fully connected layers in the model, which has four layers.

Table 4.8: Four-Layer Model Using Feature Selection

No.	No. of Layers	No. of Nodes	Accuracy	Precision	Recall	F-score	Best feature
1	**4**	**2048-2048 2048-2048**	**0.8310**	**0.83**	**0.83**	**0.83**	**597**
2	4	1024-1024-1024-1024	0.8310	0.84	0.83	0.83	589
3	4	512-512-512-512	0.8333	0.84	0.83	0.83	573
4	4	256-256-256-256	0.8333	0.84	0.83	0.83	587
5	4	128-128-128-128	0.8379	0.84	0.84	0.84	540
6	4	64-64-64-64	0.8379	0.84	0.84	0.84	458
7	**4**	**32-32-32-32**	**0.8402**	**0.85**	**0.84**	**0.84**	454
8	4	16-16-16-16	0.8379	0.84	0.84	0.84	504
9	4	512-256-128-64	0.8333	0.84	0.83	0.83	576
10	4	256-128-64-32	0.8402	0.85	0.84	0.84	535
11	4	128-64-32-16	0.8333	0.84	0.83	0.83	536

The results in **Table 4.8** show that the values of accuracy, precision, recall and F1-score are convergent, indicating the presence of stability in the network and the system is still maintaining its performance. The worst accuracy is **0.8310** in model 1, and the best accuracy is **0.8402** in model 7.

Figure 4.6 shows the evaluation metrics of the best results when using four layers with feature selection.

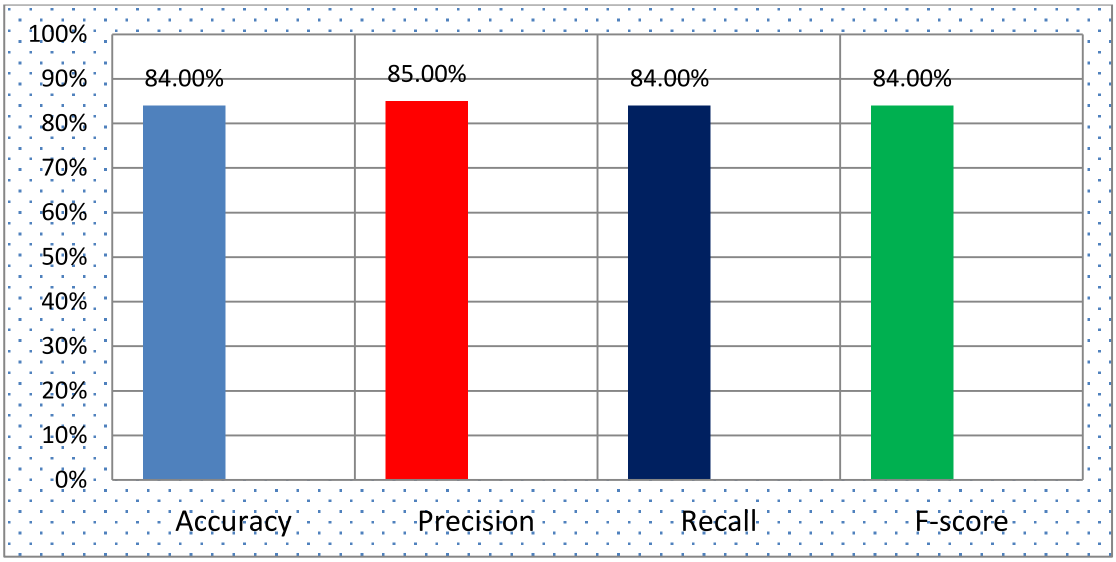

Figure 4.6: Evaluation Metrics for Four Layers Using Feature Selection for Model 7

Tables 4.6–4.8 show that using chi Square technique to the feature selection increases the performance and accuracy of the system and reduces training time.

The results show, the best architecture of the proposed system that has three layers for several reasons: the best accuracy was in the three layers. Also, the time consumed to build neural with a smaller number of layers and smaller number of nodes is less than it to build neural with more layers and more nodes in each layer. Another reason is the complexity of the models is less with a smaller number of layers and nodes.

The results show also that the best configuration is that in which the three layers each comprises 16 nodes because the model achieves the highest accuracy using this configuration. Thus, selecting the layers with a large number of nodes is not required because selecting the layers with a small number of nodes reduces the complexity of models in terms of the number of parameters and time.

4.5 Applying the System with Feature Selection Using the Movie Review Dataset

The second dataset used in this thesis is the movie review dataset, as described in Section (4.3.2). This dataset contains 400 document split into 200 negative document and 200 positive tweets; the data are split into 70% training dataset and 30% test dataset.

In **Tables 4.9-4.11**, the rows represent a set of training rounds of a set of models. The columns are as follows. The first column is the model number, which represents the sequence of the model in the experiment. The second column represents the number of fully connected layers in the model. The third column represents the number of nodes in each layer and is approximately 2048-16. The fourth, fifth, sixth and seventh columns represent the accuracy, precision, recall and F1-score of each model, respectively. Finally, the eighth column represent the best feature selection.

Table 4.9: Two-Layer Model Using Feature Selection

No.	No. of Layers	No. of Nodes	Accuracy	Precision	Recall	F-score	Best feature
1	2	2048-2048	0.9833	0.99	0.99	0.99	510
2	2	1024-1024	0.975	0.98	0.98	0.98	635
3	2	512-512	0.9833	0.99	0.99	0.99	610
4	2	256-256	0.9833	0.98	0.98	0.98	569
5	2	128-128	0.9916	0.99	0.99	0.99	640
6	2	64-64	0. 9916	0.99	0.99	0.99	614
7	2	32-32	0.9916	0.99	0.99	0.99	663
8	**2**	**16-16**	**0.9916**	**0.99**	**0.99**	**0.99**	**884**

The results in Table 4.9 shown that there is an improvement in system performance when feature selection operation has been done for all feature vectors by using chi square technique. The results show that the values of accuracy, precision, recall and F1-score are convergent, indicating the presence of stability in the network. The worst accuracy is equal to **0.975** in model 2, and the best accuracy is equal to **0. 9916** in model 8.

Figure 4.7 shows the evaluation metrics of the best results when using two layers with feature selection.

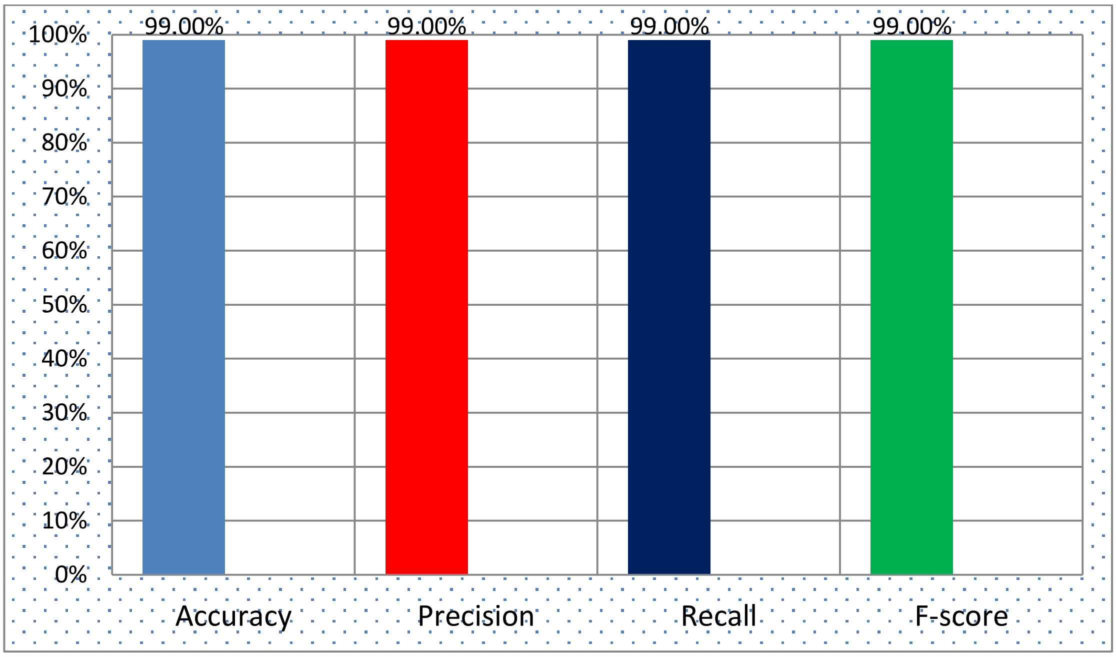

Figure 4.7: Evaluation Metrics for Two Layers Using Feature Selection for Model 8

Table 4.10: Three-Layer Model Using Feature Selection

No.	No. of Layers	No. of Nodes	Accuracy	Precision	Recall	F-score	Best feature
1	3	**2048-2048-2048**	**0.975**	**0.98**	**0.97**	**0.97**	**502**
2	3	1024-1024-1024	0.9833	0.98	0.98	0.98	568
3	3	512-512-512	0.9833	0.98	0.98	0.98	516
4	3	256-256-256	0.9833	0.98	0.98	0.98	485
5	3	128-128-128	0.9833	0.98	0.98	0.98	512
6	3	64-64-64	0.9916	0.99	0.99	0.99	546

7	**3**	**32-32-32**	**0.9916**	**0.99**	**0.99**	**0.99**	**647**
8	3	16-16-16	0.975	0.98	0.97	0.97	602
9	3	1024-512-256	0.975	0.98	0.97	0.97	479
10	3	512-256-128	0.9833	0.98	0.98	0.98	512
11	3	256-128-64	0.9833	0.98	0.98	0.98	513
12	3	128-64-32	0.9916	0.98	0.98	0.98	755
13	3	64-32-16	0.975	0.98	0.97	0.97	468

The results in **Table 4.10** show that the values of accuracy, precision, recall and F1-score are convergent, indicating the presence of stability in the network. The worst accuracy is **0.975** in model 1, and the best accuracy is **0.9916** in model 7.

Figure 4.8 shows the evaluation metrics of the best results when using three layers with feature selection.

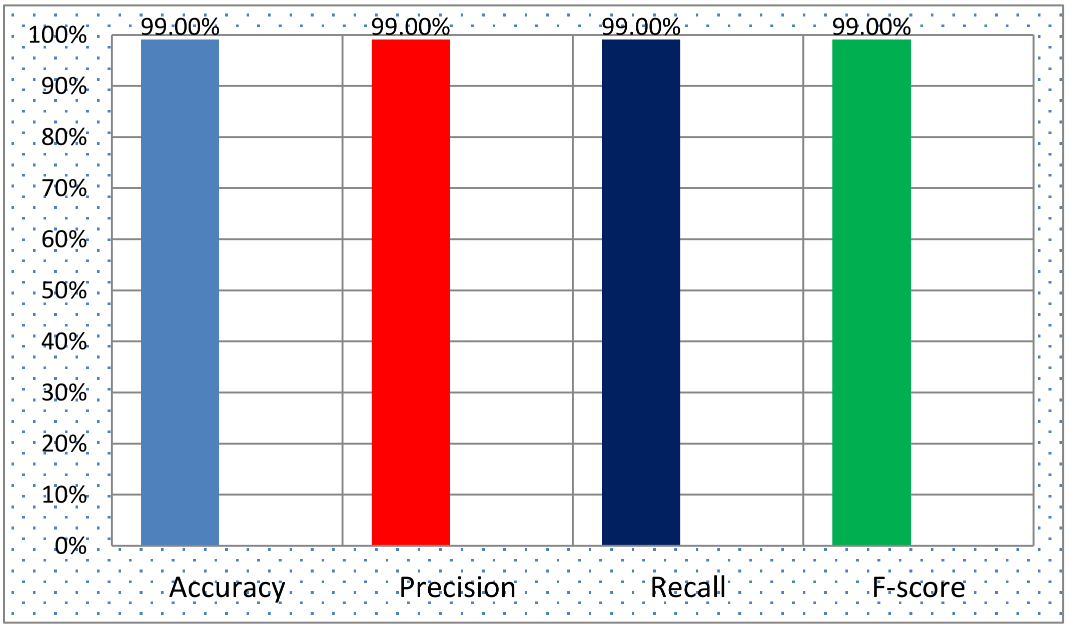

Figure 4.8: Evaluation Metrics for Three Layers Using Feature Selection for Model 7

Table 4.11: Four-Layer Model Using Feature Selection

No.	No. of Layers	No. of Nodes	Accuracy	Precision	Recall	F-score	Best feature
1	**4**	**2048-2048 2048-2048**	**0.975**	**0.98**	**0.97**	**0.97**	**500**
2	4	512-512-512-512	0.9833	0.98	0.98	0.98	473
3	4	256-256-256-256	0.9916	0.99	0.99	0.99	485
4	**4**	**128-128-128-128**	**0.9916**	**0.99**	**0.99**	**0.99**	**532**
5	4	64-64-64-64	0.9833	0.98	0.98	0.98	473
6	4	32-32-32-32	0.975	0.98	0.97	0.97	492
7	4	16-16-16-16	0.975	0.98	0.97	0.97	492
8	4	1024-512-256-128	0.975	0.98	0.97	0.97	452
9	4	512-256-128-64	0.9833	0.98	0.98	0.98	457

The results in **Table 4.11** show that the values of accuracy, precision, recall and F1-score are convergent, indicating the presence of stability in the network. The worst accuracy is **0.975** in model 1, and the best accuracy is **0.9916** in model 4.

Figure 4.9 shows the evaluation metrics of the best results when using four layers with feature selection.

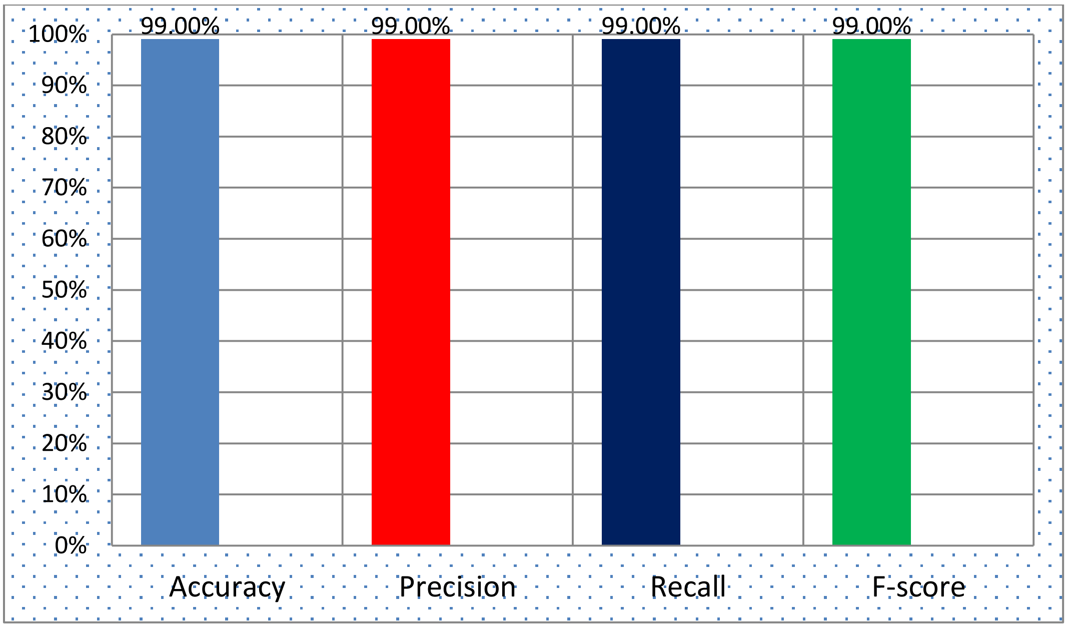

Figure 4.9: Evaluation Metrics for Four Layers Using Feature Selection for Model 5

The results in **Tables 4.9-4.11** show that the best configuration is that in which the two layers each comprises 16 nodes because the model achieves the highest accuracy using this configuration. Thus, selecting the layers with a large number of nodes is not required because selecting the layers with a small number of nodes reduces the complexity of models in terms of the number of parameters and time.

4.6 Discussion of Results

These results indicate that the proposed method, when used on Twitter dataset, achieves the best result and highest scores in terms of system's performance when using three layers. The best accuracy is 84.95%, and best results in precision, recall and F1-score are 85%, 84% and 84%, respectively. In addition, when using the movie review dataset, the proposed method achieves high the highest scores in system performance when using two layers. The best accuracy is 99%, and the best result in precision, recall and F1-score is 99%. These results indicate that we encountered difficulties in the Twitter dataset for the following reasons:

1- The size of compacted and the abbreviated statements generated by each tweet is limited (only 140 characters) which results in a small set of features, thus making the sentiment analysis is more difficult. However, with the movie reviews, the text length is unlimited which makes the set of features is large, thus the sentiment analysis and making a decision with the movie reviews will be easier than the decision with the sentiment analysis of twitter.

2- Use of slang in Twitter: These words are different from English words. Thus, a noisy and obscure dataset is obtained.

3- Twitter features: A tweet is a short message full of noise, such as irrelevant symbols, hashtags, user references, URLs, misspellings, emoticons and stop words. Such noisy characteristics frequently effect the performance of SA approaches.

4- User variety: Users express their opinions in various ways because of the wide popularity of Twitter; some use different languages in between, and others use repeated words or symbols to convey an emotion.

4.7 Result Comparison

Herein, the results of the proposed method are compared with those of other methods using Twitter and movie review datasets, as shown in Table 4.12.

Table 4.12:Comparison of Proposed System with Different Works

No	Method	Dataset	Accuracy	Year	Author
1	Using a feedforward neural network	Twitter	74.15%	2015	Duncan and Yanqing Zhang [28]
2	Using naive Bayes (NB) smoothing methods	Twitter	72.34%	2016	Indriani et al. [31]
3	Using deep feedforward neural network and multilayer perceptron (MLP)	Twitter	75.03% in first method. 67.45% in second method.	2017	Ramadhani and Goo [6]
4	Using machine learning classifiers such as Multinomial NB, logistic regression and SGD	Twitter	75%	2018	Elbagir et al. [34]
5	Convolutional neural networks (CNNs) and artificial neural networks	Twitter	81%	2019	Reddy et al. [37]
6	Using two machine learning methods: NB and SVM	Movie reviews	0.89.53% in first method. 0.94.06% in second method.	2015	Tripathy et al. [29]
7	Using three machine learning classifiers: NB, K-nearest neighbour and random forest	Movie reviews	81.45% in first method. 78.65% in second method. 55.30% in third method.	2017	Baid et al. [17]
8	Using four deep learning methods: CNN, LSTM,CNN-LSTM and MLP	Movie reviews	97,60 in first method. 96,57 in second method. 98,07 in third method. 78,27 in fourth method.	2018	Ay Karakuş et al. [35]
9	Using four deep learning methods: CNN, CNN-LSTM, LSTM, and MLP	Movie reviews	87.7% in first method. 89.2% in second method. 86.74% in third and fourth method.	2019	Ali et al. [38]
10	MLP	Twitter & Movie reviews	85% Twitter 99%Movie reviews	2019	The proposed system

Chapter Five

Conclusions and Future Work

5.1 Introduction

This chapter presents the conclusion of the results and the recommendations of the implemented SA system. The following section shows how the system achieved the desired goal of SA. Some conclusions and future works are mentioned in this chapter.

5.2 Conclusions

After implementing the proposed approach and obtaining the results, the following conclusions were found:

1. Based on a number of exterminates, it can be concluding that, the pre-processing stage is important in noise removal or data filtering for more accurate data, and less computational time.

2. The obtained results demonstrated that the Multilayer Perceptron Neural Networks is an efficient classifier to deal with huge datasets. In addition, the TF-IDF technique was able to manageable groups for processing by reducing the initial set of raw data.

3. Using a large number of hidden layers and nodes could provide highly accurate data.

4. Chi-square is an efficient technique could be use to reduce the features of the Twitter and movie review datasets and thereby limit the computational time of the system and improve system accuracy.

5.3 Future Work

This thesis has many research points that are worthy of further investigation. A number of suggestions for future work are provided as follows:

1. We can improve and train our model to handle a range of sentiments. Tweets do not always have positive or negative sentiments. At times, they may have no sentiment, i.e. they are neutral.

2. Given the multilingual messages on Twitter, an automatic classifier may be built for more than one language to predict sentiments for any language.

3. The MLP classifier was adopted in our work. The concept can be extended to other complex architectures, such as RNNs, CNNs and CNN-LSTM.

4. Another social media dataset may be used, and the data size may be increased.

References

[1] K. Francisco and M. Panguila, “**Sentiment Analysis on Social Media Data Using Intelligent Techniques,**” International Journal of Engineering Research and Technology, 2019.

[2] S. Joshi and D. Deshpande, “**Twitter Sentiment Analysis System**,” Int. J. Comput. Appl., vol. 180, no. 47, pp. 35–39, 2018.

[3] K. Mouthami, K. N. Devi, and V. M. Bhaskaran, “**Sentiment analysis and classification based on textual reviews**,” 2013 Int. Conf. Inf. Commun. Embed. Syst. ICICES 2013, pp. 271–276, 2013.

[4] M. Amajd, Z. Kaimuldenov, and I. Voronkov, “Text Classification with Deep Neural Networks,” International Conference on Actual Problems of System and Software Engineering, pp. 364-370,2017.

[5] L. Zhang, S. Wang, and B. Liu, “**Deep learning for sentiment analysis: A survey**,” Wiley Interdiscip. Rev. Data Min. Knowl. Discov., vol. 8, no. 4, Jul. 2018.

[6] R. Adyan Marendra, and H. S. Goo, "**Twitter sentiment analysis using deep learning methods,**" 2017 IEEE 7th International Annual Engineering Seminar (InAES), pp. 1-4, 2017.

[7] A. Kumar and T. M. Sebastian, “**Sentiment Analysis: A Perspective on its Past, Present and Future**,” Int. J. Intell. Syst. Appl., vol. 4, no. 10, pp. 1–14, 2012.

[8] A. Kumar and G. Garg, “**Systematic literature review on context-based sentiment analysis in social multimedia**,” Multimed. Tools Appl., 2019.

[9] S. Shahnawaz and P. Astya, “**Sentiment analysis: Approaches and open issues,**” Proceeding - IEEE Int. Conf. Comput. Commun. Autom. ICCCA 2017, vol. 2017-January, pp. 154–158, 2017.

[10] K. K. Uma and D. K. Meenakshisundaram, “**A Novel Optimizer Technique for Sentiment Polarity in Social Web Environment**,” International Journal of Applied Engineering Research, pp. 118–125, 2019.

[11] S. Siddiqui, A. A. Monem, and K. Shaalan, “**Sentiment analysis in Arabic**,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9612, pp. 409–414, 2016.

[12] V. A. and S. S. Sonawane, “**Sentiment Analysis of Twitter Data: A Survey of Techniques**,” Int. J. Comput. Appl., vol. 139, no. 11, pp. 5–15, 2016.

[13] V. VardanReddy, M. Maila, S. Sri Raghava, Y. Avvaru, and S. V. Koteswarao, "**A Survey on Analysis of Twitter Opinion Mining Using Sentiment Analysis**," Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol., pp. 537–542, 2019.

[14] M. Dorothy and S. Rajini, "**The Various Approaches for Sentiment Analysis: A Survey**," Int. J. Sci. Res., vol. 5, no. 1, pp. 44–46, 2016.

[15] T. P. Sahu and S. Ahuja, "**Sentiment analysis of movie reviews: A study on feature selection and classification algorithms**," Int. Conf. Microelectron. Comput. Commun. MicroCom 2016, 2016.

[16] S. J. S. Chowdary, S. W. Farooq, S. A. Siddique, and S. Dinesh, "**Sentiment Analysis of movie reviews using Microsoft Text Analytics and Google Cloud Natural Language API,**" vol. 5, no. 1, pp. 147–150, 2019.

[17] M. M and S. Mehla, "**Sentiment Analysis of Movie Reviews using Machine Learning Classifiers**," Int. J. Comput. Appl., vol. 182, no. 50, pp. 25–28, 2019.

[18] B. Lopez and X. Sumba, "**IMDb Sentiment Analysis**," pp. 2–6, 2019.

[19] M. T. Scholar and A. Pratap, "**A Survey on Various Sentiment Analysis Approaches and Its Challenges**," vol. 4, no. 3, pp. 463–466, 2018.

[20] S. Li, S. Y. M. Lee, Y. Chen, C. R. Huang, and G. Zhou, "**Sentiment classification and polarity shifting,**" Coling 2010 - 23rd Int. Conf. Comput. Linguist. Proc. Conf., vol. 2, no. August, pp. 635–643, 2010.

[21] A. Pak and P. Paroubek, "**Twitter as a corpus for sentiment analysis and opinion mining,**" Proc. 7th Int. Conf. Lang. Resour. Eval. Lr. 2010, vol. 5, no. 12, pp. 1320–1326, 2010.

[22] R. Passonneau, "**Sentiment Analysis of Twitter Data**," Proc. Work. Lang. Soc. Media (LSM 2011), no. June, pp. 30–38, 2011.

[23] R. Moraes, J. F. Valiati, and W. P. Gavião Neto, "**Document-level sentiment classification: An empirical comparison between SVM and ANN**," Expert Syst. Appl., vol. 40, no. 2, pp. 621–633, 2013.

[24] M. Karim and S. Das, "**Sentiment Analysis on Textual Reviews**," IOP Conf. Ser. Mater. Sci. Eng., vol. 396, no. 1, pp. 122–127, 2018.

[25] P. Gamallo and M. Garcia, "**Citius: A Naive-Bayes Strategy for Sentiment Analysis on English Tweets**," no. SemEval, pp. 171–175, 2015.

[26] R. Suchdev, P. Kotkar, R. Ravindran, and S. Swamy, “**Twitter Sentiment Analysis using Machine Learning and Knowledge-based Approach**,” Int. J. Comput. Appl., vol. 103, no. 4, pp. 36–40, 2014.

[27] L. Augustyniak, P. Szymánski, T. Kajdanowicz, and W. Tuliglowicz, “**Comprehensive study on lexicon-based ensemble classification sentiment analysis**,” *Entropy*, vol. 18, no. 1, pp. 1–29, 2015.

[28] B. Duncan and Y. Zhang, “**Neural networks for sentiment analysis on Twitter**,” Proc. 2015 IEEE 14th Int. Conf. Cogn. Informatics Cogn. Comput. ICCI*CC 2015, pp. 275–278, 2015.

[29] A. Tripathy, A. Agrawal, and S. K. Rath, “**Classification of Sentimental Reviews Using Machine Learning Techniques**,” Procedia Comput. Sci., vol. 57, pp. 821–829, 2015.

[30] M. Z. Asghar, S. Ahmad, M. Qasim, S. R. Zahra, and F. M. Kundi, **“SentiHealth: creating health-related sentiment lexicon using hybrid approach**,” Springerplus, vol. 5, no. 1, 2016.

[31] R. A. Ramadhani, F. Indriani, and D. T. Nugrahadi, “Comparison of Naive Bayes smoothing methods for Twitter sentiment analysis,” 2016 Int. Conf. Adv. Comput. Sci. Inf. Syst. ICACSIS 2016, pp. 287–292, 2017.

[32] Y. Zhou, M. Lan, and Y. Wu, “**ECNU at SemEval-2017 Task 4: Evaluating Effective Features on Machine Learning Methods for Twitter Message Polarity Classification,**” pp. 812–816, 2018.

[33] H. Han, X. Bai, and P. Li, “**Augmented sentiment representation by learning context information,**” Neural Comput. Appl., vol. 4, 2018.

[34] S. Elbagir and J. Yang, “**Sentiment analysis of twitter data using machine learning techniques and scikit-learn**,” ACM Int. Conf. Proceeding Ser., p. 57, 2018.

[35] B. Ay Karakuş, M. Talo, İ. R. Hallaç, and G. Aydin, “**Evaluating deep learning models for sentiment classification**,” Concurr. Comput. , vol. 30, no. 21, pp. 1–14, 2018.

[36] Murthy, Jamuna S., G. M. Siddesh, and K. G. Srinivasa, "**TwitSenti: A Real-Time Twitter Sentiment Analysis and Visualization Framework,"** Journal of Information & Knowledge Management, 2019.

[37] D. M. Reddy, D. N. V. S. Reddy, and D. Prema K V, “**Twitter Sentiment Analysis using Distributed Word and Sentence Representation**,” arXiv preprint arXiv:1904.12580, 2019.

[38] N. Mohamed Ali, M. M. A. El Hamid, and A. Youssif, “**Sentiment Analysis for Movies Reviews Dataset Using Deep Learning Models**,” Int. J. Data Min. Knowl. Manag. Process, vol. 09, no. 03, pp. 19–27, 2019.

[39] R. Bhatia, P. Garg, and R. Johari, **“Corpus Based Twitter Sentiment Analysis**,” Proceedings of 3rd International Conference on Internet of Things and Connected Technologies (ICIoTCT) ,pp. 26–27 , 2018.

[40] K. Chakraborty and S. Bhattacharyya, “Comparative Sentiment Analysis on a Set of Movie,” vol. 4, no. April, pp. 311–318, 2018.

[41] S. Grabner-Kräuter, “**Web 2.0 Social Networks: The Role of Trust**,” J. Bus. Ethics, vol. 90, no. SUPPL. 4, pp. 505–522, 2009.

[42] A. L. Traud, P. J. Mucha, and M. A. Porter, “**Social structure of Facebook networks**,” Phys. A Stat. Mech. its Appl., vol. 391, no. 16, pp. 4165–4180, 2012.

[43] A. Sarlan, C. Nadam, and S. Basri, **“Twitter sentiment analysis**,” Conf. Proc. - 6th Int. Conf. Inf. Technol. Multimed. UNITEN Cultiv. Creat. Enabling Technol. Through Internet Things, ICIMU 2014, pp. 212–216, 2015.

[44] H. A. Efstathiades, “**Extract Knowledge from Social Networks**,” Master thesis, Delft University of Technology, Netherlands, 2013.

[45] J. Spencer and G. Uchyigit, “Sentimentor: Sentiment analysis of twitter data,” CEUR Workshop Proc., vol. 917, no. June, pp. 56–66, 2012.

[46] N. A. Ghani, S. Hamid, I. A. Targio Hashem, and E. Ahmed, “**Social media big data analytics: A survey**,” Comput. Human Behav., vol. 101, pp. 417–428, 2019.

[47] M. Z. Asghar, F. M. Kundi, S. Ahmad, A. Khan, and F. Khan, “**T-SAF: Twitter sentiment analysis framework using a hybrid classification scheme**,” Expert Syst., vol. 35, no. 1, pp. 1–19, 2018.

[48] A. Chandra Pandey, D. Singh Rajpoot, and M. Saraswat, “**Twitter sentiment analysis using hybrid cuckoo search method,**” Inf. Process. Manag., vol. 53, no. 4, pp. 764–779, 2017.

[49] R. Patel, “Sentiment Analysis on Twitter Data Using Machine Learning,” PhD Thesis. Laurentian University of Sudbury, 2017.

[50] M. Ramzan and M. Ahmad, “**Evolution of data mining: An overview**,” Proc. 2014 Conf. IT Business, Ind. Gov. An Int. Conf. by CSI Big Data, CSIBIG 2014, 2014.

[51] M. Edison and A. Aloysius, “**Lexicon based Acronyms and Emoticons Classification of Sentiment Analysis (SA) on Big Data**,” Int. J. Database Theory Appl., vol. 10, no. 7, pp. 41–54, 2017.

[52] P. Mishra, R. Rajnish, and P. Kumar, "**Sentiment analysis of Twitter data: Case study on digital India,**" 2016 International Conference on

Information Technology (InCITe)-The Next Generation IT Summit on the Theme-Internet of Things: Connect your Worlds. IEEE, 2016.

[53] S. B. Moralwar and S. N. Deshmukh, "**International Journal of Computer Sciences and Engineering Engineerin g Open Access Different Approaches of Sentiment Analysis**," no. 3, 2015.

[54] M. Y. Al musaddar, "**Improving Arabic Light Stemming in Information Retrieval Systems**," MSC Thesis, Islamic University, Palestine ,vol. 1, 2014.

[55] H. Li, "**Sentiment Analysis and Opinion Mining on Twitter With Gmo Keyword**," MSC Thesis, North Dakota State University, 2016.

[56] S. Vijayarani, M. J. Ilamathi, M. Nithya, A. Professor, and M. P. Research Scholar, "**Preprocessing Techniques for Text Mining -An Overview**," International Journal of Computer Science & Communication Networks, vol. 5, no. 1, pp. 7–16, 2015.

[57] D. Virmani and S. Taneja, A text preprocessing approach for efficacious information retrieval, vol. 669. Springer Singapore, pp. 13–22, 2019.

[58] Iadh Ounis, "**Inverse Document Frequency**", Reference Work Entry Encyclopedia of Database Systems, pp 1570-1571, 2009.

[59] J. A. Cerón-Guzmánn, "**A Sentiment Analysis Model of Spanish Tweets. Case Study: Colombia 2014 Presidential Election**," MSC Thesis ,Univ. Nac. Colomb. …, p. 62, 2016.

[60] H. D. White, "**Bag of works retrieval: TF*IDF weighting of works co-cited with a seed**," Int. J. Digit. Libr., vol. 19, no. 2–3, pp. 139–149, 2018.

[61] E. Cantú-Paz, S. Newsam, and C. Kamath, "**Feature selection in scientific applications**," KDD-2004 - Proc. Tenth ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., pp. 788–793, 2004.

[62] G. Chandrashekar and F. Sahin, "**A survey on feature selection methods**," Comput. Electr. Eng., vol. 40, no. 1, pp. 16–28, 2014.

[63] M. Zareapoor and S. K. R, "**Feature Extraction or Feature Selection for Text Classification: A Case Study on Phishing Email Detection**," Int. J. Inf. Eng. Electron. Bus., vol. 7, no. 2, pp. 60–65, 2015.

[64] S. Bahassine, A. Madani, and M. Kissi, "**New stemming for arabic text classification using feature selection and decision trees**," J. Eng. Sci. Technol., vol. 12, no. 6, pp. 1475–1487, 2017.

[65] P. Kumbhar, "**A Survey on Feature Selection Techniques and Classification Algorithms for Efficient Text Classification**," Int. J. Sci. Res., vol. 5, no. 5, pp. 1267–1275, 2016.

[66] S. George K and S. Joseph, “**Text Classification by Augmenting Bag of Words (BOW) Representation with Co-occurrence Feature**,” IOSR J. Comput. Eng., vol. 16, no. 1, pp. 34–38, 2014.

[67] M. Drahansky et al., “**Variable Selection and Feature Extraction Through Artificial Intelligence Techniques**,” Intech, vol. i, no. tourism, p. 13, 2016.

[68] S. Kuang and B. D. Davison, “**Learning word embeddings with chi-square weights for healthcare tweet classification**,” Appl. Sci., vol. 7, no. 8, 2017.

[69] I. Sumaiya Thaseen and C. Aswani Kumar, “**Intrusion detection model using fusion of chi-square feature selection and multi class SVM**,” J. King Saud Univ. - Comput. Inf. Sci., vol. 29, no. 4, pp. 462–472, 2017.

[70] Dubey, A., Gupta, A., Raturi, N., and Saxena, P, "**Item-Based Collaborative Filtering Using Sentiment Analysis of User Reviews**," International Conference on Application of Computing and Communication Technologies. Springer, Singapore, pp. 77–87, 2018.

[71] N. Zainuddin and A. Selamat, “**Sentiment analysis using Support Vector Machine**,” I4CT 2014 - 1st Int. Conf. Comput. Commun. Control Technol. Proc., no. I4ct, pp. 333–337, 2014.

[72] H. Simon," **Neural Networks and Learning Machines**," 3/E. Pearson Education India, 2010.

[73] M. Negnevitsky, "**Artificial intelligence: A guide to intelligent systems**", Systems (2nd Edition), vol. 29, no. 1. 2005.

[74] Y. Jin, “**Artificial Neural Networks**,” pp. 73–91, 2003.

[75] S. Shanmuganathan, “**Artificial Neural Network Modelling: An Introduction**,” Springer, no. July, pp. 1–14, 2016.

[76] C. C. Aggarwal “**Neural Networks and Deep Learning**,” A Textbook, Springer International Publishing AG, part of Springer Nature 2018.

[77] A. Abraham, “**Artificial Neural Networks**,” Handbook of Measuring System Design, 2005.

[78] P.SIBI, S.A. JONES and P. Siddarth, “**Analysis Of Different Activation Functions Using Back Propagation Neural Networks**,” Journal of Theoretical and Applied Information Technology, vol. 47, no. 3, pp. 1264–1268, 2013.

[79] C. E. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “**Activation Functions: Comparison of Trends in Practice and Research for Deep Learning** ", Technical Report arXiv:1811.03378, 2018.

[80] J. Stefanowski, “**Artificial Neural Networks – Basics of MLP , RBF and Kohonen Networks**,” Institute of Computing Science, Lecture 13, Version 2010. 2010.

[81] M. Gullu, M. Yilmaz, and I. Yilmaz, “**Application of Back Propagation Artificial Neural Network for Modelling Local GPS / Levelling Geoid Undulations : A Comparative Study**,” J. Surv. Eng., vol. 133, no. May 2011, pp. 18–22, 2011.

[82] G. Batres-Estrada, “**Deep Learning for Multivariate Financial Time Series**,” p. 94, 2015.

[83] S. Chithore, “**Sentiment Analysis with Multilayer Perceptron using Emoticon Space Model**,” International Journal of Advance Engineering and Research, pp. 410–419, 2017.

[84] E. P. Costa, C. Postal, A. C. Lorena, R. S. Ad, C. Postal, and A. A. Freitas, “**A Review of Performance Evaluation Measures for Hierarchical Classifiers**,” Association for the Advancement of Artificial Intelligence, 2007.

[85] M. Hossin and M. N. Sulaiman, “**A Review On Evaluation Metrics For Data Classification Evaluations**,” International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.5, No.2, March 2015.