ENSEMBLE CLASSIFIER BASED WEB SPAM DETECTION IN IoT ENVIRONMENT

A PROJECT REPORT

Submitted by

JENITLIN B	**960318104019**
PRATHEBA C	**960318104025**
SANDHYA T	**960318104028**
BINDHU R	**960318104301**

In partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

BETHLAHEM INSTITUTE OF ENGINEERING,

KARUNGAL

ANNA UNIVERSITY: CHENNAI 600 025

JUNE 2022

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report titled "**ENSEMBLE CLASSIFIER BASED WEB SPAM DETECTION IN IoT ENVIRONMENT**" is the bonafide work of **"JENITLIN B (960318104019), PRATHEBA C (960318104025),SANDHYA T (960318104028), BINDHU R (960318104301)"** who carried out the project work under my supervision.

SIGNATURE

Mr.P.Libin Jacob

HEAD OF THE DEPARTMENT
Assistant Professor
Computer science and Engineering
Bethlahem Institute of Engineering
Karungal, Kanyakumari Dist.

SIGNATURE

Mrs.W.V.Vinisha

SUPERVISOR
Assistant Professor
Computer science and Engineering
Bethlahem Institute of Engineering
Karungal, Kanyakumari Dist.

Submitted for the Project Viva-Voce Examination held on ……………

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

First and foremost, we consider it is the cardinal duty to thank **ALMIGHTY GOD** for his grace and blessings throughout the project.

We express our heart thanks to our Chairman **Shri. N.GERALD SELVARAJA**, for providing full facilities and Technical environment to start this project work.

We express our heartfelt gratitude to the Director **Er.T.ISAN**, for our Institute for his constant support. We express our heartfelt gratitude to the Principal **Dr. JERALD JEBAKUMAR** of our Institute for his constant support.

We extent our thanks to ever loving H.O.D **Mr.P.Libin Jacob,** Assistant Professor, Department Of Computer Science And Engineering, for rendering his full supports both mentally and technically by encouraging us at all times we needed it.

We would like to express our whole hearted thanks to our Project Internal Guide **Mrs. W.V.Vinisha,** Assistant Professor, and Department Of Computer Science And Engineering**,** to meet all challenges to come up with our project victoriously.

We extent our sincere thanks to all the Teaching and Non-Teaching Staff Members in Department of Computer Science And Engineering for their valuable help and supports.

ABSTRACT

From the last few years, Internet of Things has revolutionized the entire world. In this, various smart objects perform the tasks of sensing and computing to provide uninterrupted services to the end users in different applications such as smart transportation, e-healthcare to name a few. However, while accessing data from the Internet, web spam is one of the challenges to be handled. The major cause for the failure of IoT devices is due to the attacks, in which web spam is more prominent. There seems a requirement of a technique which can detect the web spam before it enters into a device. Motivated from these issues, an Ensemble Classifier based web spam detection technique is proposed. Random Forest, KNearest Neighbor, Support Vector Machine and Decision Tree classifiers are used in the Ensemble classifier. Each classifier produces the quality score of the webpage. These quality scores are then ensemble to generate a single score, which predicts the spam of the web page. The Principal Component Analysis (PCA) based feature extraction approach and Pearson Correlation Coefficient (PCC) based feature selection approach is utilized in the proposed approach.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

SVM	Support Vector Machine
LSTM	Long short-term memory
AI	Artificial Intelligence
NB	Naïve Bayes
TF-IDF	Term Frequency-Inverse Document Frequency
ML	Machine Learning
NLP	Natural Language Processing
LR	Logistic regression
RF	Random Forest
DL	Deep Learning
CNN	Convolution neural network
RNN	Recurrent Neural Network
ABC	Artificial Bee Colony
NLTK	Natural Language Tool Kit
BoW	Bag of words
CBOW	continuous Bag of words

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Spam can be defined as an unsolicited and unwanted message sent electronically by a sender that has no current relationship with the recipient. There exist several subsets of electronic spam. Indeed, spam message can be sent over multiple communication channels, such as e-mail, SMS, social networks or shopping online platforms. E-mail spam consumes users’ time, as users must identify and remove undesired messages; it also takes up limited mailbox space and buries important personal e-mails. Meanwhile, SMS spam is typically transmitted over a mobile network. Recently, social network spam has received increased attention from both researchers and practitioners due to both the considerable number of spammers and the potential negative effects of social network spam on convenience and understanding of all the followers.

Review volume and review valence have been reported to be significant determinants of retail sales in a metaanalysis of more than empirical studies. This is particularly relevant for high-involvement products that can only be reviewed upon consumption. Consumers’ experience of product use is therefore an important assumption. As shown in a recent survey, more than 80% of consumers trust online reviews as much as they trust personal recommendations. This is why a considerable attention is given to spam filtering in the above communication channels. Spam messages can be filtered either manually or automatically. Obviously, manual spam filtering by identifying spam message and removing it is a time-consuming task. Moreover, spam messages may contain a security threat, such as links to phishing web sites or servers hosting malware. Therefore, over a number of decades researches and practitioners have worked on improving automatic spam filtering algorithms. Machine learning techniques are particularly known to be highly accurate in detecting spam messages.

The main concept of the machine learning algorithms is to build a word list and assign a weight to each word accordingly. However, spammers tend to include common legitimate messages into the spam message in order to decrease the probability of being detected. There is a number of existing machine learning algorithms applied to spam filtering, such as neural networks (NNs), support vector machines (SVMs), Naïve Bayes (NB) and random forest (RF). According to the survey ensemble learning methods, such as bagging and random forest, outperform traditional single classifiers. The ensemble methods combine the predictions of several base machine learning algorithms in order to improve accuracy and robustness over single algorithms. In previous studies, ensemble methods employed traditional classifiers like decision trees to effectively filter spam messages.

1.2 IMPORTANCE OF SPAM DETECTION

The idea of spam is very simple: to send a message to millions of people and profit from the one person who replies. Recent studies have shown that on average 80 % of e-mails is spam, with significant differences in spam rates among countries. As a result, serious negative effects on the worldwide economy have been observed including lower productivity, the costs associated with delivering spam, and the cost with delivering spam and viruses/phishing attacks. Therefore, an effective spam filter may also improve user productivity and reduce the consumption of information technology resources such as the help desk.

For individuals, more accurate spam filters may increase their trust in e-mail communication. The availability of unlimited pre-pay SMS packages has enabled the same approach for SMS spam. Increasing the cost of sending spam and reducing the burden spam places on users require highly accurate spam filters. Statistics show that a large proportion of all messages in social networks are spam messages. For instance, the study by Nexgate, a major company specialized in cyber security, reported that during the first half of 2013 there has been a 355% growth of social spam. For every seven new social media accounts, five new spammers are detected. The growing opportunities of social networks and their popularity have attracted many users. These days the base of social network users is steadily growing, and considerable amount of communication is done through social networks. However, along with legitimate and useful information, inappropriate and unwanted content is also released on these networks. Indeed, spam senders target social network users as well. Moreover, business social networks like LinkedIn are also affected. This has serious economic and social consequences.

Spam messages decrease work productivity, increase IT support related resources (help desk) and may even result in security incidents. This is why a considerable attention is given to spam filtering in social networks. Fake reviews are unwanted and misleading reviews which can be submitted and listed on multiple online platforms, such as online shops and travel aggregators. In correlation with the number of internet users the number of users who shop online is growing as well. TripAdvisor is one of the most popular travel related website. User base of TripAdvisor is over 455 million average monthly unique visitors. Moreover, there are 600 million reviews about 7.5 million properties, restaurants, tours, etc. Many users take into consideration other users' reviews while choosing a property to stay. And fake review is becoming a problem due to the fact they may mislead potential buyers which will result in potential lawsuit against the seller and other adverse effects. Recent researches have shown that about every third review is fake on

TripAdvisor. In order to guarantee fair competition, it is crucial to detect and remove fake reviews, since they give competitive advantage or disadvantage.

1.3 E-MAIL SPAM DETECTION

Spammers (persons sending spam messages) gather e-mail addresses from a wide range of sources, such as websites and chatrooms, send unsolicited messages in bulk. This has serious adverse effect on the recipient, including waste of time and resources. Specifically, e-mail spam has negative effects on the memory of email server, CPU performance and user time. Moreover, the fraudulent practices of spammers may result in substantial financial losses of the recipients. Although the global spam volume (percentage of total e-mail traffic) decreased to about 55% in the last decade, the volume of e-mail messages with pernicious attachments is steadily increasing. The largest share of spam e-mail spam was produced in China with about 20% of e-mail spam volume. Spam senders are strongly motivated to send bypass spam filters in order to increase the revenue. Therefore, spam filtering represents a challenging task because spammers use different techniques, in order to decrease spam detection rate.

There are a number of methods such as using irrelevant, random or misspelled words, to evade commonly used spam filters. Spam filtering techniques can be categorized into non-machine learning and machine learning approaches. The former include legislative approaches, changes to protocols and models of operation, rule-, signature-, and hash-based filtering, whitelists (trusted senders) and blacklists, and traffic analysis. With machine learning approaches, spam filtering starts with text pre-processing, with tokenization performed first to extract the words (multi-words) in each message. Next, typically, the initial set of words is reduced by stemming, lemmatization, and stop-words removal. Bag-of-words (BoW), also known as the vector-space model, is a common approach to represent the weights of the pre-processed words. Term frequency–inverse document frequency (tf.idf) is a popular specific weighting scheme.

Feature selection algorithms, such as filters or wrappers, may then be applied to reduce the size of the feature space, which is useful mainly because not all classification methods can handle high-dimensional data. Finally, machine learning methods are applied to classify the preprocessed dataset. The first spam classifiers employed NB algorithms due primarily to their simplicity and computational efficiency. Concerning SVM, another popular spam-classification algorithm, it was shown that SVMs are robust to both different datasets and preprocessing techniques. Its superiority to NB, k-nearest-neighbor (k-NN), decision trees, and NN approaches has been demonstrated in comparative studies. Artificial immune systems (AISs) represent another promising method for spam filtering. Zitar and Hamdan used a genetic algorithm to train AISs to improve spam filter performance. Meta-learning algorithms have also recently attracted increasing attention. The combination of boosting and SVM outperformed single classifiers on several benchmark datasets in Trivedi and Dey.

1.4 SMS SPAM DETECTION

Short message service (SMS) is a popular mean of communication these days. The increasing number of mobile phones in use leads to increased number of SMS sent and received. The rapid smartphones penetration has contributed to the growth of online instant messaging and SMS usage. Due to constant decrease of SMS price along with introduction of unlimited mobile phone plans, spammers can send spam messages at a very low cost or for free. Various techniques were developed in order to address SMS classification. After evaluating results of the experiments, researches come to conclusion that that Bayesian filtering technique can be employed successfully to detect SMS Spam. The results of the experiments showed that SVM and NB demonstrated better classification performance than kNN. Some other researches used terms normalization to create new attributes and later used to expand original text sampling aiming to alleviate factors which may lead to lower algorithm classification performance. Another proposed method used distinctive features while eliminating uninformative ones considering certain requirements on term characteristics. Indeed, SVM represents the most popular machine learning method in recent comparative studies

1.5 SOCIAL NETWORK SPAM DETECTION

User base of social networks is growing over the number of years. For instance, Facebook, one of the biggest social networks in the world, grew from one billion to two billion users just in 5 years. Social network spam has become a major concern of industry and academia because it may include unwanted content, such as insults, hate speech, malicious links, etc. Such messages can be seen by the recipient's followers. Moreover, they may lead to confusions and misdirection in public discussions. Fighting social network spam with traditional legal methods has serious limitation because spam messages in social networks can be sent from different countries.

It is important to note that spammers may use anonymizers, making it difficult to trace them. In order to overcome this problem, several social network spam filters have recently been developed. Features related to tweet content and user behavior were identified and used for machine learning using SVM. A statistical analysis of language used in tweets represents an alternative approach, which identifies spam tweets in isolation (i.e., without user information) using their trending topics. URLs in social media have also been used in the behavior-based spam detection system proposed. More precisely, the behavioral signals were obtained from both the URL sender and receiver. In other words, a high accuracy was achieved without using other tweets' attributes such as those based on message content. In addition to spam messages detection, recent studies have also considered an alternative task of social spammer (profile) detection. These are reported as the major supporters of malicious users, and a graph-topology based classifier was used to detect such bridge linkages. These feature distributions were used in a social spammer detection framework that integrated this information with a social regularization term incorporate into a classification model.

A multilayer social network was defined, and the identification of spammers was based on the existence of overlapping community-based features of users represented in the form of hypergraphs, such as structural behavior and URL characteristics. Indeed, it was shown that combining social spammer filtering with spam message filtering improves the performance of both tasks. Although Twitter represents the most frequently used source of data, alternative social networks have also been examined. The most important features were then used in the SVM classifier for spam detection. This approach outperformed traditional supervised classifiers for the spammer detection task. In recent years, there has been an increasing interest in dimensionality reduction techniques with the aim of improving the prediction performance and stability of social network spam filters. Several researchers employed feature selection and extraction methodologies to identify the most important features for social network spam filtering.

1.6 REVIEW SPAM DETECTION

Review spam (fake review) has been increasingly recognized as a major concern for online shopping. To affect consumers' decisions and thus achieve competitive advantage, positive and negative review spam are intended to promote or demote target products. As consumers have limited capacity to identify review spam, machine learning methods have been employed for their early detection. To automatically classify reviews into spam or truthful class, an annotated corpus of reviews (with class labels) is typically used for training and testing. A considerable amount of literature has been published on the automatic detection of review spam in the last decade. More precisely, spammers' tendency to duplicate their product reviews was utilized.

To detect spammers who can adapt their behavior, proposed a heterogeneous review graph that captures the relationships among reviews, reviewers and reviewed shops. Thus, the trustiness of reviewers, the honesty of reviews and the reliability of shops could be calculated without considering review content. Inspired by this approach, proposed a probabilistic graph classifier, in which the multimodal embedded representation of nodes is obtained using a bidirectional NN with attention mechanism. Review metadata (content, timestamp and rating) were combined with relational data in a unified semi-supervised framework called SpEagle.

Spam attacks were reported to be correlated to review ratings and, therefore, abnormal temporal patterns in the ratings may indicate spam attacks. By elaborating this idea, a list of indicative signals of review spam over time was used for realtime detection of abnormal review events. Furthermore, temporal features were combined with users' spatial patterns to find that review spam exhibit geographical outsourcing and spammers are more active in weekdays. Most existing review spam detection systems extract informative features from the review content. Such features are typically represented by bag-of-words (n-grams) psycholinguistic word lists (e.g., positive/negative words or spatial words) or partof-speech tagging (e.g., first-person pronouns). Aspect sentiment was identified in Liu et al. to detect fraud users. Word embeddings have recently been used to obtain the semantic representation of reviews. The CBOW model was also used together with relational features to develop a semi-supervised framework. Word embeddings were also trained using sentence-based CNNs to produce document representations for review spam detection in several product domains.

1.7 OBJECTIVES

- To propose an Ensemble Classifier based web spam detection technique with Random Forest, K-Nearest Neighbor, Support Vector Machine and Decision Tree classifiers.
- To utilize Principal Component Analysis (PCA) based feature extraction approach
- To use Pearson Correlation Coefficient (PCC) based feature selection approach.

CHAPTER 2

LITERATURE REVIEW

2.1 A PERFORMANCE EVALUATION OF MACHINE LEARNINGBASED STREAMING SPAM TWEETS DETECTION

The popularity of Twitter attracts more and more spammers. Spammers send unwanted tweets to Twitter users to promote websites or services, which are harmful to normal users. In order to stop spammers, researchers have proposed a number of mechanisms. The focus of recent works is on the application of machine learning techniques into Twitter spam detection. However, tweets are retrieved in a streaming way, and Twitter provides the Streaming API for developers and researchers to access public tweets in real time. There lacks a performance evaluation of existing machine learning-based streaming spam detection methods. In this paper, we bridged the gap by carrying out a performance evaluation, which was from three different aspects of data, feature, and model. A big ground-truth of over 600 million public tweets was created by using a commercial URL-based security tool. For real-time spam detection, we further extracted 12 lightweight features for tweet representation. Spam detection was then transformed to a binary classification problem in the feature space and can be solved by conventional machine learning algorithms. We evaluated the impact of different factors to the spam detection performance, which included spam to nonspam ratio, feature discretization, training data size, data sampling, time-related data, and machine learning algorithms. The results show the streaming spam tweet detection is still a big challenge and a robust detection technique should take into account the three aspects of data, feature, and model.

A dataset with ground-truth (annotated instances with class labels for referencing) is needed to perform a number of challenging machine learning-based streaming spam tweets detection tasks. However, we found that no datasets are publicly available specially for our task. Although there are a few dataset published by some researchers , the labeled instances are spammers instead of spam tweets. As a result, we decided to collect streaming tweets and generate the ground-truth. We will also make this dataset available for others researchers to use. In this section, we will describe our large dataset with over 600 million tweets, including more than 6.5 million spam tweets.

2.2 DETECTING AND PREVENTING CYBER INSIDER THREATS: A SURVEY

Information Communications Technology systems are facing an increasing number of cyber security threats, the majority of which are originated by insiders. As insiders reside behind the enterprise-level security defence mechanisms and often have privileged access to the network, detecting and preventing insider threats is a complex and challenging problem. In fact, many schemes and systems have been proposed to address insider threats from different perspectives, such as intent, type of threat or available audit data source. This survey attempts to line up these works together with only three most common types of insider namely traitor, masquerader and unintentional perpetrator, while reviewing the countermeasures from a data analytics perspective. Uniquely, this survey takes into account the early-stage threats which may lead to a malicious insider rising up. When direct and indirect threats are put on the same page, all the relevant works can be categorised as host, network or contextual data-based according to audit data source and each work is reviewed for its capability against insider threats, how the information is extracted from the engaged data sources, and what the decisionmaking algorithm is. The works are also compared and contrasted. Finally, some issues are raised based on the observations from the reviewed works and new research gaps and challenges identified.

2.3 STATISTICAL TWITTER SPAM DETECTION DEMYSTIFIED: PERFORMANCE, STABILITY AND SCALABILITY

With the trend that the Internet becoming more accessible and our devices being more mobile, people are spending an increasing amount of time on social networks. However, due to the popularity of online social networks, cyber criminals are spamming on these platforms for potential victims. The spams lure users to external phishing sites or malware downloads, which has become a huge issue for online safety and undermined user experience. Nevertheless, the current solutions fail to detect Twitter spams precisely and effectively. In this paper, we compared the performance of a wide range of mainstream machine learning algorithms, aiming to identify the ones offering satisfactory detection performance and stability based on a large amount of ground truth data. With the goal of achieving real-time Twitter spam detection capability, we further evaluated the algorithms in terms of the scalability. The performance study evaluates the detection accuracy, the TPR/FPR and the F-measure; the stability examines how stable the algorithms perform using randomly selected training samples of different sizes. The scalability aims to better understand the impact of the parallel computing environment on the reduction of training/testing time of machine learning algorithms.

2.4 SPAM DETECTION USING ANN AND ABC ALGORITHM

Social network becomes an effective method to engage with our friends. It enables the users to extract a number of in-formation and its usage are increasing day by day. Amidst all the social networking sites, Twitter is one of the interactive social networking services. To change the authorized user accounts, many spammers are utilized a vast amount of spam. Machine Learning (ML) technique is utilized for spam detection system in social sites and for the detection of spammer. Data collection is usually done from H-Spam 14 site with the help of preprocessing mechanism, the data is transforming into lowercase. After the first step, pre-processed data comes under feature extraction phase, which utilizes tokenization process to breakdown each sentence into word group in order to extract the best feature from the raw data. The optimization algorithm referred as Artificial Bee Colony (ABC) is used to pick the optimized value from extracted set of features. It is also utilized to get the optimal sets of features from spam and nonspam data. At the end, performance measure criterion and comparing the existing and proposed work in order to look over the progress of the proposed work. In this work, spam detection system is having higher accuracy, precision, recall, and Fmeasure as compares to classifiers used previously such as, Naïve Bayes and Support Vector Machine (SVM).

2.5 A NEURAL NETWORK-BASED ENSEMBLE APPROACH FOR SPAM DETECTION IN TWITTER

As the social networking sites get more popular, spammers target these sites to spread spam posts. Twitter is one of the most popular online social networking sites where users communicate and interact on various topics. Most of the current spam filtering methods in Twitter focuses on detecting the spammers and blocking them. However, spammers can create a new account and start posting new spam tweets again. So there is a need for robust spam detection techniques to detect the spam at tweet level. These types of techniques can prevent the spam in real time. To detect the spam at tweet level, often features are defined, and appropriate machine learning algorithms are applied in the literature. Recently, deep learning methods are showing fruitful results on several natural language processing tasks. We want to use the potential benefits of these two types of methods for our problem. Toward this, we propose an ensemble approach for spam detection at tweet level. We develop various deep learning models based on convolutional neural networks (CNNs). Five CNNs and one feature-based model are used in the ensemble. Each CNN uses different word embeddings (Glove, Word2vec) to train the model. The feature-based model uses contentbased, user-based, and n-gram features. Our approach combines both deep learning and traditional feature-based models using a multilayer neural network which acts as a meta-classifier. We evaluate our method on two data sets, one data set is balanced, and another one is imbalanced. The experimental results show that our proposed method outperforms the existing methods.

2.6 EVALUATING THE EFFECTIVENESS OF MACHINE LEARNING METHODS FOR SPAM DETECTION

Technological advances are accelerating the dissemination of information. Today, millions of devices and their users are connected to the Internet, allowing businesses to interact with consumers regardless of geography. People all over the world send and receive emails every day. Email is an effective, simple, fast, and cheap way to communicate. It can be divided into two types of emails: spam and ham. More than half of the letters received by the user spam. To use Email efficiently without the threat of losing personal information, you should develop a spam filtering system. The aim of this work is to reduce the amount of spam using a classifier to detect it. The most accurate spam classification can be achieved using machine learning methods. A natural language processing approach was chosen to analyze the text of an email in order to detect spam. For comparison, the following machine learning algorithms were selected: Naive Bayes, K-Nearest Neighbors, SVM, Logistic regression, Decision tree, Random forest. Training took place on a ready-made dataset. Logistic regression and NB give the highest level of accuracy – up to 99%. The results can be used to create a more intelligent spam detection classifier by combining algorithms or filtering methods.

2.7 SPAM REVIEW DETECTION USING DEEP LEARNING

A robust and reliable system of detecting spam reviews is a crying need in todays world in order to purchase products without being cheated from online sites. In many online sites, there are options for posting reviews, and thus creating scopes for fake paid reviews or untruthful reviews. These concocted reviews can mislead the general public and put them in a perplexity whether to believe the review or not. Prominent machine learning techniques have been introduced to solve the problem of spam review detection. The majority of current research has concentrated on supervised learning methods, which require labeled data - an inadequacy when it comes to online review. Our focus in this article is to detect any deceptive text reviews. In order to achieve that we have worked with both labeled and unlabeled data and proposed deep learning methods for spam review detection which includes Multi-Layer Perceptron (MLP), Convolutional Neural Network (CNN) and a variant of Recurrent Neural Network (RNN) that is Long Short-Term Memory (LSTM). We have also applied some traditional machine learning classifiers such as Nave Bayes (NB), K Nearest Neighbor (KNN) and Support Vector Machine (SVM) to detect spam reviews and finally, we have shown the performance comparison for both traditional and deep learning classifiers.

2.8 AN ENHANCED MECHANISM OF SPAM AND CATEGORY DETECTION USING NEUROSVM

Internet society suffers from a lot of problems including the spam in the content which an user get online. It is not necessary that the each user likes each type of categorical data. Hence, before identifying the spam, it is also required to identify the community of the data. The proposed architecture utilizes the concept of binary classification of SVM (Support Vector Machine) and the concept of multiclass classifier of Neural Network to identify the SPAM and category of the data received online. This paper describes a hybrid clustering mechanism of NEUROSVM to justify the classification mechanism. The evaluation has been done using evaluation parameters of tp, fp, tn, fn.

The problem of this research work is to identify the categorical spam out of given sequence of text. Identifying both the things requires a lot of sophistication in the cluster architecture and the cluster training mechanism. It is not necessary that the classified result provide the correct group always. If there are more than two categories in the classification due to a multiclass classifier, it is required to join it with a binary classifier. The research problem includes creation of a hybrid classifier in order to train and classify category and spam.

2.9 DETECTING REVIEW MANIPULATION ON ONLINE PLATFORMS WITH HIERARCHICAL SUPERVISED LEARNING

Opinion spammers exploit consumer trust by posting false or deceptive reviews that may have a negative impact on both consumers and businesses. These dishonest posts are difficult to detect because of complex interactions between several user characteristics, such as review velocity, volume, and variety. We propose a novel hierarchical supervised-learning approach to increase the likelihood of detecting anomalies by analyzing several user features and then characterizing their collective behavior in a unified manner. Specifically, we model user characteristics and interactions among them as univariate and multivariate distributions. We then stack these distributions using several supervised-learning techniques, such as logistic regression, support vector machine, and k-nearest neighbors yielding robust meta-classifiers. We perform a detailed evaluation of methods and then develop empirical insights. This approach is of interest to online business platforms because it can help reduce false reviews and increase consumer confidence in the credibility of their online information. Our study contributes to the literature by incorporating distributional aspects of features in machinelearning techniques, which can improve the performance of fake reviewer detection on digital platforms

2.10 OPINION SPAM DETECTION BY INCORPORATING MULTIMODAL EMBEDDED REPRESENTATION INTO A PROBABILISTIC REVIEW GRAPH

Spam reviews typically appear perfectly normal until examined in a large context. The standard approach to classifying reviews independently ignores these relations. In this study, we propose a complex probabilistic graph classification approach to address the problem of opinion spam detection. To obtain an initial effective spamicity estimation for the nodes (reviews, authors, and products) in the graph, we first train a neural network with attention mechanism to learn the multimodal embedded representation of nodes by leveraging both textual and rich features. Then based on the node prior computation, a heterogeneous graph is constructed to capture the relationships among different kinds of nodes, and the beliefs are further updated through iterative message propagation. To support this work, we collect two kinds of real-life datasets, which are separately composed of 97,839 restaurant reviews and 31,317 hotel reviews. The evaluation of the two datasets demonstrates the effectiveness of the proposed approach. We further analyze several salient rich features and the intermediate component of our model, thereby revealing that their states capture certain statistical characteristics of the datasets.

CHAPTER 3

PROPOSED APPROACH

3.1 EXISTING APPROACHES

- Support Vector Machine (SVM)
- Long short-term memory (LSTM)
- Fuzzy classifier
- Naive Bayes (NB)
- Logistic regression (LR)
- Random Forest (RF)
- Convolution neural network (CNN)
- Recurrent Neural Network (RNN)

3.1.1 Dataset Collection Techniques

The first phase in spam identification is the collecting of textual data, comprising spam and non-spam (ham) material, from social media sites such as Twitter, Facebook, online reviews, hotel evaluations, and e-mails. They are extracted with the help of an appropriate API, such as the Facebook API or the Twitter API, which are both free and allow users to search and collect data from several accounts. They also enable the capture of data using a “hashtag” or “keyword,” as well as the collecting of data posted over time. Based on the text content, we can identify data as spam or ham, and official social networking sites may flag some accounts or postings as spam.

3.1.2 Pre-Processing Techniques

Text-preprocessing is a significant technique for cleaning the raw data in a dataset, and it is the first and most important stage in removing extraneous text

Before extracting features from text, it is necessary to eliminate any undesired data from the dataset. Unwanted data in the text dataset include punctuation, http links, special characters, and stop words.

Tokenization. It entails breaking down words into little components known as tokens. HTML tags, punctuation marks, and other undesirable symbols, for example, are removed from the text. The most widely used tokenization method is whitespace tokenization. The entire text is broken down into words during this procedure by removing whitespaces. To split the text into tokens, a well-known

Python module known as "regular expressions" can be used, and it is frequently used to do Natural Language Processing (NLP) tasks.

Stemming. It is concerned with the process of reducing words to their fundamental meanings; for instance, the terms drunk, drink, and drank are reduced to their root, drink. Stemming can produce non-meaningful terms that aren't in the dictionary, and it can be accomplished using the Natural Language Tool Kit library in conjunction with PorterStemmer. Overstemming occurs when a significantly more chunk of a word is cut off than is required, resulting in words being incorrectly reduced to the same root word. Due to understemming, some words may be mistakenly reduced to more than one root word.

Lemmatization. It employs lexical and morphological analysis, as well as a proper lexicon or dictionary, to link a term to its origin. The underlying word is known as a 'Lemma,' and words such as plays, playing, and played are all distinct variants of the word 'play.' So 'play' is the root word or 'Lemma' of all these words. The WordNet Lemmatizer is a Python Natural Language Tool Kit (NLTK) module that searches the WordNet Database for Lemmas. While lemmatizing, you must describe the context in which you want to lemmatize. Normalization It is the process of reducing the number of distinct tokens in a text by reducing a term to its simplest version. It aids in text cleaning by removing extraneous information.

Stopwords removal. They are a category of frequently used terms in a language that have little significance. By removing these terms, we will be able to focus more on the vital facts. Stop words like “a,” “the,” “an,” and “so” are frequently used, and by deleting them, we may drastically reduce the dataset size. They can be successfully erased with the NLTK python library.

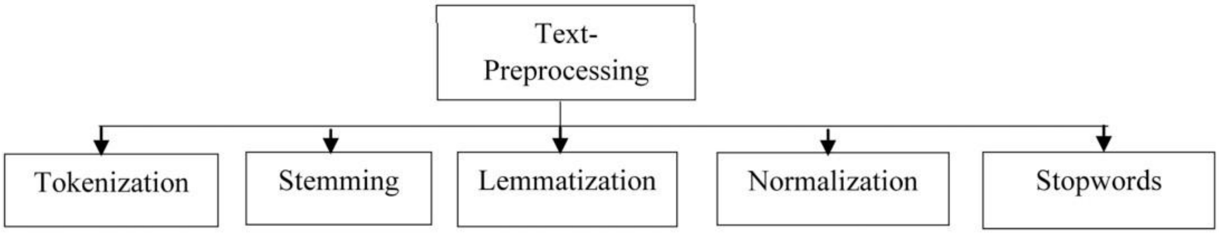


Figure 3.1 Text preprocessing techniques

3.1.3 Feature-Extraction Techniques

Because many machine learning algorithms rely on numerical data rather than text, it is required to convert the text input into numerical vectors. This method’s goal is to extract meaningful information from a text that describes essential aspects of it.

Bag of words (BoW). The bag of words strategy is the most common and straightforward of all feature extraction procedures; it generates a word presence feature set from all of an instance's words. Each document is viewed as a collection or bag that contains all of the words. We may obtain a vector form that tells us the frequency of each word in a document, as well as repeated words in our document. Barushka & Hajek (2019) developed a spam review detection model that uses ngrams and the skip-gram word embedding method. They employed deep learning models to detect spam in 400 positive and negative hotel reviews from the TripAdvisor website.

Term frequency-inverse document frequency (TF-IDF). When employing bag of words, the terms with the highest frequency become dominant in the data. Domain-specific terms with lower scores may be eliminated or ignored as a result of this issue. This technique is performed by multiplying the number of times a word appears in a document (Term-Frequency-TF) by the term's inverse document frequency (Inverse-Document Frequency-IDF) across a collection of documents. These scores can be used to highlight unique terms in a document or words that indicate crucial information. The computed TF-IDF score can then be fed into machine learning algorithms such as Support Vector Machines, which substantially improve the results of simpler methods such as Bag-of-Words.

One hot encoding. Every word or phrase in the given text data is stored as a vector with only the values 1 and 0. Every word is represented by a separate hot vector, with no two vectors being identical. The sentence's list of words can be defined as a matrix and implemented using the NLTK python package because each word is represented as a vector. Word embedding One-hot encoding is ideal when we just have a little amount of data. Because the complexity develops substantially, we can use this method to encode a vast vocabulary. Comparable words have similar vector representations in word embedding, which is a form of word representation technique. Because each word is mapped to a different vector and the technique resembles a neural network, it is usually referred to as deep learning.

Word2Vec. To process text made up of words, this approach transforms words into vectors and works in the same way as a two-layer network. Each word in the corpus is allocated a matching vector in the space. Word2vec employs either a continuous skipgram or a continuous bag of words architecture (CBOW). In the continuous skipgram, the current word is utilized to predict the neighboring words, whereas in the CBOW model, a middle word is predicted based on the surrounding or neighbouring words. The skip-gram model can accurately represent even rare words or phrases with a small quantity of training data, but the CBOW model is several times faster to train and has slightly better accuracy for common keywords. The word2vec approach has the advantage of allowing high-quality word embedding to be learned in less time and space. It makes it possible to learn larger embeddings (with greater dimensions) from a much larger corpus of text. Glove word embedding It's an unsupervised model for generating a vector for word/text representation. The distance between the terms is determined by their semantic similarity.

3.1.4 Spam Text Classification Techniques

Text classifiers can organize and categorize practically any sort of material, including documents and internet text. Text classification is an important stage in natural language processing, with applications ranging from sentiment analysis to subject labelling and spam detection. Text classification can be done manually or automatically, however in the manual approach, a human annotator assesses the text's content and categorizes it correctly. Machine learning techniques and other Artificial Intelligence (AI) technologies are used to automatically classify text in a faster and more accurate manner utilizing automatic text classification models.

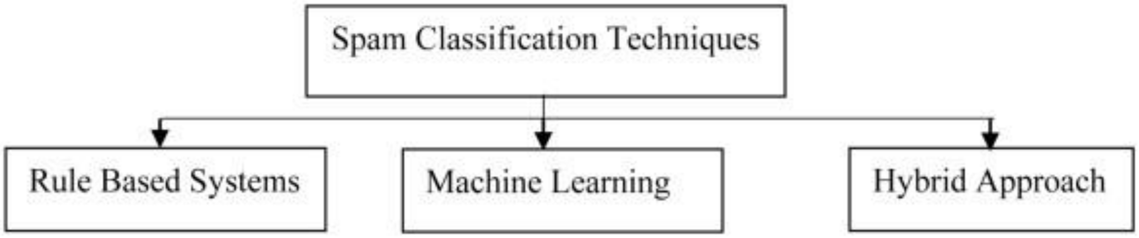


Figure 3.2 Spam Text Classification Techniques

- **Rule based systems.** They work by sorting the text into distinct groups using handcrafted linguistic rules. The entering text is classified using semantic factors based on its content. Certain terms can help you evaluate whether or not a text message is spam. The spam text has a few distinctive phrases that help differentiate it from non-spam language. The document is classified as spam when the number of spam words in it exceeds the number of non-spam (ham) terms. They operate by employing a set of framed rules, each of which is given a weight. The spam text corpus is scanned for spam content, and if any rules are found in the text, their weight is added to the overall score. SpamAssassin is open source software that aids in the creation of rules for various categories and is preferred by spam detection researchers. Some rule-based systems rely on static rules that can't be changed, so they can't deal with constantly changing spam content. To improve the method's ability to detect spam, the established rules must be updated on a regular basis. To deal with the varying nature of spam, the automatic rule generation concept can be used. For complex systems, rule-based systems have significant drawbacks in terms of time consumption, analysis complexity, and rule structuring.

They also require more contextual features for effective spam detection, as well as a large training corpus.

- **Machine Learning (ML).** To detect spam reviews, a variety of machine learning techniques have been deployed. There are two types of machine learning: supervised learning and unsupervised learning, both of which are extensively utilized in NLP applications. By combining two algorithms, this hybrid system was able to detect spam effectively while saving time and improving accuracy. They tested their algorithm on three publicly available e-mail spam datasets and discovered that it outperformed the others in spam filtering. They were able to obtain an accuracy of 88.12% using their hybrid approach. To protect social media accounts from spam, Sharma et al. (2021) used Decision Tree (DT) and K-Nearest Neighbor (K-NN) classifiers. They tested their method using the UCI machine learning e-mail spam dataset. With a classification accuracy of 90% and an F1score of 91.5%, the Decision Tree classifier produced better results. In their research, found that multi-algorithm systems outperform single-algorithm systems when it comes to spam classification. For e-mail spam detection, they compared the performance of supervised and unsupervised machine learning algorithms.

For better spam detection, the supervised approach outperformed the unsupervised approach. A two-step methodology used to ensure that the mail people received was not spam. They utilized URL analysis and filtering to see if any of the links in the email were malicious or not. A total of five machine learning algorithms were investigated. On the e-mail spam dataset, Naive Bayes and Support Vector Machine achieved the highest accuracy of over 90%. Machine learning has the ability to adapt to changing conditions, and it can help overcome the limitations of rule-based spam filtering techniques. Support Vector Machines (SVM), a supervised learning model that analyses data and identifies patterns for classification, is among the most significant machine learning techniques. SVMs are straightforward to train, and some researchers assert that they outperform many popular social media spam classification methods. However, due to the computational complexities of the data input, the resilience and usefulness of SVM for high dimension data shrinks over time. Another machine learning algorithm that has been successfully used to detect spam in social media text is the decision tree.

When it comes to training datasets, decision trees (DT) require very little effort from users. They suffer from certain disadvantages, such as the complexity of controlling tree growth without proper pruning and their sensitivity to over fitting of training data. As a consequence, they are rather poor classifiers and their classification accuracy is restricted. A Naive Bayes (NB) classifier simply applies

Bayes' theorem to the perspective classification of each textual data, assuming that the words in the text are unrelated to one another. Because of its simplicity and ease of use, it is ideal for spam classification and it could be used to detect spam messages in a variety of datasets with various features and attributes. An ensemble strategy, which combines various machine learning classifiers, can also be utilized to improve spam categorization jobs. We can deduce from various studies on Machine Learning for spam classification that ML techniques occasionally suffer from computational complexity and domain dependence. The researchers recommend Deep Learning (DL) techniques to avoid such limitations in ML techniques for spam classification because some algorithms take much longer to train and use large resources based on dataset.

- **Hybrid approach.** To increase spam classification performance, hybrid spam detection systems combine a machine learning-based classifier with a rule-based approach. To detect spam in emails, a hybrid technique is utilized that comprised "Rule Based Subject Analysis" (RBSA) and machine learning algorithms. Their rule-based solution involves assigning suitable weights to spam material and generating a matrix that is then submitted to a classifier. They tested their method on the Enron dataset (email corpus), and their proposed work with the SVM classifier achieved a very low positive rate of 0.03 with a 99% accuracy. The proposed "Conceptual Similarity Approach" computes the relationship between concepts based on their co-occurrence in the corpus. They tested their hybrid spam classification strategy using the Spambase and Enron corpus datasets. They have a near-perfect 98% accuracy rate. A novel technique used to spam detection in their work, merging Neural Networks (NN) with rule-based algorithms. They classified spam content using Neural Networks, rule-based pre-processing, and behavior identification modules with an encoding approach. They tested their approach on an email corpus containing lakhs of emails and scored a 99.60% spam detection accuracy score

- **Deep Learning (DL) Approaches.** Deep learning models are gaining popularity among NLP researchers due to their ability to solve challenging problems. Deep learning is based on the idea of building a very large neural network inspired by brain activities and training it using a massive amount of data. They can cope with the scalability issue and extract the features from the data automatically. The most popular deep learning models among NLP researchers are Convolutional Neural Networks (CNN) and Long Short Tern Memory (LSTM) networks. Convolutional

Neural Networks (CNN), one of the most important and extensively used Deep Learning approaches, has received a lot of attention in recent times for performing NLP tasks. It has been used successfully for sentiment analysis, and text categorization, pattern recognition, and other tasks.

Their technique was able to capture semantic information from text and outperformed CNN in classifying text texts. The LSTM model outperformed the GRU model in spam detection, achieving an accuracy of 98.39%. Alauthman

(2020) employed the Gated Recurrent Unit-Recurrent Neural Network (GRURNN) to recognize Botnet spam E-mails. On the SPAMBASE dataset, which included 4,601 spam and 2,788 non-spam e-mails, they achieved an accuracy of 98.7%. They evaluated the performance of GRU with several machine learning algorithms, but the GRU-based strategy produced the best results for spam detection. On spam text information obtained from the UCI machine learning repository, they achieved a 99% accuracy. A deep learning model based on LSTM and BERT used to overcome issues such as unfair representation, inadequate detection effect, and poor practicality in Chinese spam detection. They created this model to capture complex text features using a long-short attention mechanism. They could be able to outperform current methods in terms of classification performance by achieving an F1-Score of around 92.8.

3.1.5 Disadvantages

- Machine Learning classifier is not suitable for large data sets.
- It does not perform very well when the data set has more noise i.e. target classes are overlapping
- Large training data needed
- It fails to achieve better results

3.2 PROPOSED APPROACH

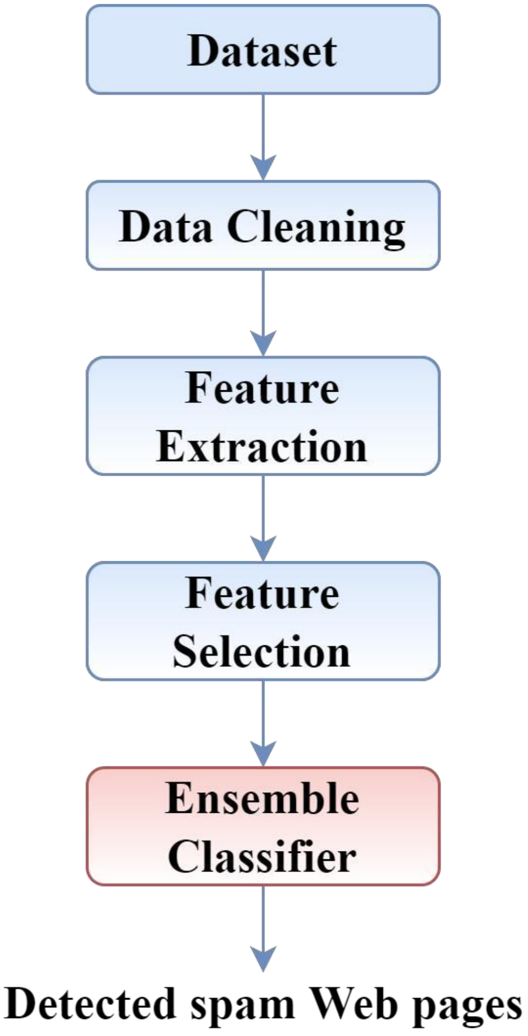


Figure 3.3 Block diagram of proposed approach

The steps involved in the proposed approaches are

- **Dataset cleaning:** Incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset is removed
- **Feature extraction:** Raw data transformed into numerical features
- **Feature selection:** The number of input variables reduced
- **Classification:** The Ensemble classifier is trained with dataset and the spam website is detected

3.2.1 Data Collection

- The proposed framework is validated by performing the experiments on the publicly available dataset, i.e., WEBSPAMUK2007.
- This data collection is launched by the Universita degli Studi di Milano, the Laboratory of Web Algorithms.
- It is the collection of 114529 hosts with 41 features.

3.2.2 Dataset Cleaning

The aim of data cleaning process is to improve the quality of dataset. This step performs removing duplicates, handling missing values and encoding. Machines can read only numeric data but the dataset consists of both numeric and nominal data. Thus, encoding is utilized to convert the characters in the dataset to numeric values. The last step in preprocessing is data scaling performed to speed up the process. The features in the dataset highly vary in range with magnitude and units. It is necessary to keep all the data in one format and hence scaling can normalize the data within the range between 0 and 1.

3.2.3 Feature Extraction using Principal Component Analysis (PCA)

Feature extraction reduces the dimension of the dataset. It is the procedure of transforming the correlated variables into uncorrelated variables. It is done by extracting the variance among the variables. It is also known as principal axis method or data compression technique. The standard value of each row is multiplied with the standard value of each column, which results in formation of Principal component (PC).

PCA is a dimensionality reduction that identifies important relationships in our data, transforms the existing data based on these relationships, and then quantifies the importance of these relationships so we can keep the most important relationships and drop the others. To remember this definition, we can break it down into four steps: We identify the relationship among features through a Covariance Matrix. Through the linear transformation or eigen decomposition of the Covariance Matrix, we get eigenvectors and eigen values. Then we transform our data using Eigenvectors into principal components. Lastly, we quantify the importance of these relationships using Eigen values and keep the important principal components.

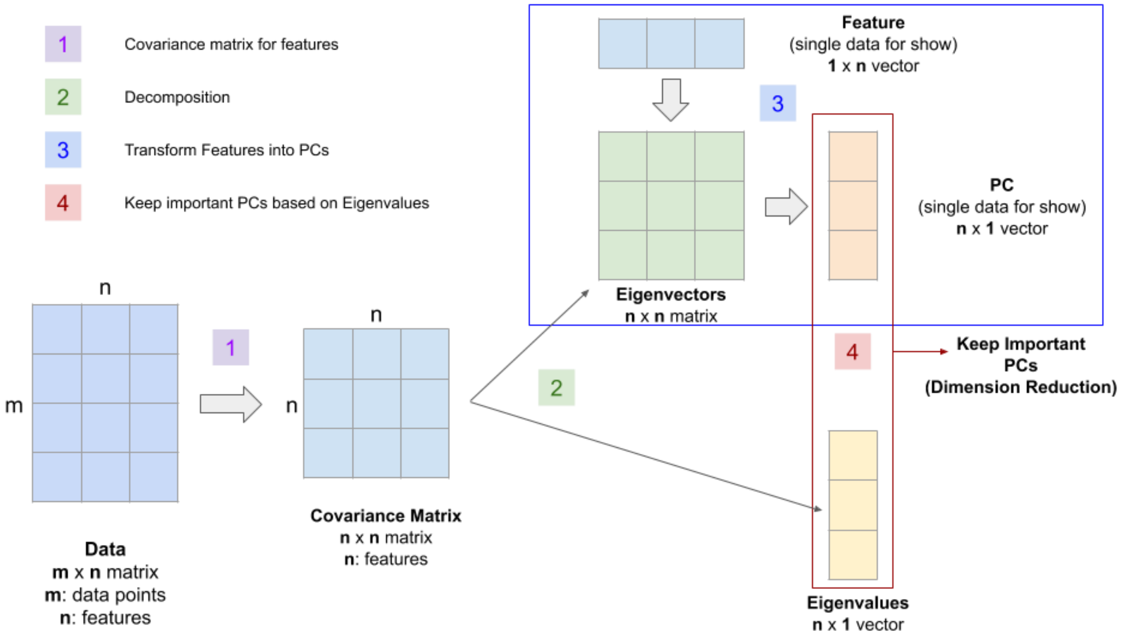


Figure 3.4 PCA based feature extraction

3.2.4 Feature Selection Using Pearson Correlation Coefficient (PCC)

Feature selection is the process of reducing the number of input variables when developing a predictive model. It is desirable to reduce the number of input variables to both reduce the computational cost of modeling and, in some cases, to improve the performance of the model.

The Pearson correlation coefficient (PCC) also known as Pearson's r, the Pearson product-moment correlation coefficient (PPMCC), the bivariate correlation, or colloquially simply as the correlation coefficient is a measure of linear correlation between two sets of data. It is the ratio between the covariance of two variables and the product of their standard deviations; thus it is essentially a normalized measurement of the covariance, such that the result always has a value between −1 and 1. As with covariance itself, the measure can only reflect a linear correlation of variables, and ignores many other types of relationship or correlation. As a simple example, one would expect the age and height of a sample of teenagers from a high school to have a Pearson correlation coefficient significantly greater than 0, but less than 1.

3.2.5 Ensemble classifier

A voting classifier is a machine learning estimator that trains various base models or estimators and predicts on the basis of aggregating the findings of each base estimator. The aggregating criteria can be combined decision of voting for each estimator output. The voting criteria can be of two types: Hard Voting: Voting is calculated on the predicted output class. Soft Voting: Voting is calculated on the predicted probability of the output class. How Voting Classifier can improve performance? The voting classifier aggregates the predicted class or predicted probability on basis of hard voting or soft voting. So if we feed a variety of base models to the voting classifier it makes sure to resolve the error by any model. Ensemble is the technique which is used to improve the performance of classifiers. The ensemble approach has been used, which is built in such a manner that it improves the performance of each classifiers. In the proposed framework, the three ML models, i.e., Random Forest, K-Nearest Neighbor, Support Vector Machine and Decision Tree classifiers are ensembled.

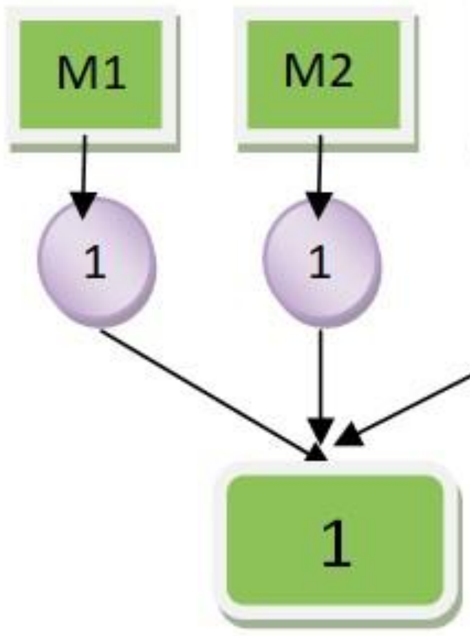


Figure 3.5 Ensemble voting classifier

□ Random Forest

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

K-Nearest Neighbor

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

Support Vector Machine

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine

Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

□ Decision Tree

Decision tree algorithm falls under the category of supervised learning. They can be used to solve both regression and classification problems. Decision tree uses the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree. We can represent any boolean function on discrete attributes using the decision tree.

CHAPTER 4

EXPERIMENTAL RESULTS AND DISCUSSION

4.1 EVALUATION METRICS FOR CLASSIFICATION

The performance results of the proposed work have been measured with four basic classification metrics i.e., TN (true negatives), FN **(**false negatives), TP (true positives) and FP (false positives). The performance measures utilized in this paper include accuracy, precision, detection rate, specificity, F-measure, FPR, FNR and FAR.

- **TP:** It is the count of correctly detected attack instances.
- **TN:** It is the count of correctly detected normal instances.
- **FP:** It is the count of incorrectly detected attack instances.
- **FN:** It is the count of incorrectly detected normal instances.
- **Accuracy:** It measures the capability of the model to predict all the instances correctly as denoted in Equation (4.1). It is the count of correctly detected instances over the total detected in the test data.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{4.1}$$

- **Precision:** It is the count of correctly detected attack instances over the total attack instances in test data and it is computed as denoted by Equation (4.2).

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{4.2}$$

- **Recall:** It is also called as sensitivity or recall. It calculates the capability of the attack detection as denoted in Equation (4.3). It is the number of correctly detected attack instances over classified attack instances.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{4.3}$$

- **Specificity**: It is called as specificity or selectivity. It is the number of correctly detected normal instances over classified normal instances. It calculates the capability of the normal instance detection as denoted in Equation (4.4).

$$\text{Specificity } = \frac{\text{TN}}{\text{TN+FP}} \tag{4.4}$$

- **F-measure:** It is defined as the harmonic mean of ADR and Precision. It is also known as F-score and it is calculated as shown in Equation (4.5).

$$\text{F} - \text{measure } = 2 \times \frac{\text{Precision}\times\text{ADR}}{\text{Precision+ADR}} \tag{4.5}$$

- **FAR:** It is the average of false negative rate (FNR) and false positive rate (FPR) computed as denoted in Equation (4.6).

$$\text{FAR } = \frac{\text{FPR+FNR}}{2} \tag{4.6}$$

Here, FPR and FNR are calculated using Equations (4.7) and (4.8) respectively.

$$\text{FPR} = \frac{\text{FP}}{\text{TN+FP}} \tag{4.7}$$

$$\text{FNR} = \frac{\text{FN}}{\text{FN+TP}} \tag{4.8}$$

- **MCC:** Mathew correlation coefficient is calculated as given in Equation (4.9).

$$\text{MCC } = \frac{\text{TP.TN}-\text{FP.FN}}{\sqrt{(\text{TP+FP})(\text{TP+FN})(\text{TN+FP})(\text{TN+FN})}} \tag{4.9}$$

- **Kappa:** It can be measured as given in Equation (4.10).

$$\text{Kappa } = \frac{\text{Accuracy}-\text{RA}}{1-\text{RA}} \tag{4.10}$$

Here, RA is random accuracy can be calculated as denoted in Equation (4.11).

$$\mathrm{RA} = \frac{(\mathrm{TN}+\mathrm{FP})\times(\mathrm{TN}+\mathrm{FN})+(\mathrm{FN}+\mathrm{TP})\times(\mathrm{FP}+\mathrm{TP})}{\mathrm{Total}\times\mathrm{Total}} \tag{4.11}$$

4.2 IMPLEMENTATION DETAILS

The software used is MATLAB. MATLAB2022a is used to implement this approach. MATLAB is an interactive programming environment for scientific computing. MATLAB is heavily used in many technical fields for data analysis, problem solving, and for experimentation and algorithm development. Disciplinespecific software written in MATLAB, organized into libraries of functions called toolboxes, is widely used as well. MATLAB has found extensive use as the basis for computational laboratory work in technical education; more than 1000 textbooks use MATLAB as a teaching vehicle.

MATLAB is a product of The Mathworks of Natick, Massachusetts, USA. There is general agreement in the technical computing community that the main reasons for MATLAB's success are its intuitive, concise syntax, the use of complex matrices as the default numeric data object, the power of the built-in operators, easily used graphics, and its simple and friendly programming environment, allowing easy extension of the language. To this one can add the reliability of the numerical methods on which the operators are based.

4.3 RESULTS

Accuracy	0.8880

Error	0.1120
Sensitivity	0.9282
Specificity	0.8264
Precision	0.8914
False Positive Rate	0.1736
F1_score	0.9094
Matthews Correlation Coefficient	0.7641
Kappa	0.7631

Table 4.1 Performance of classifier

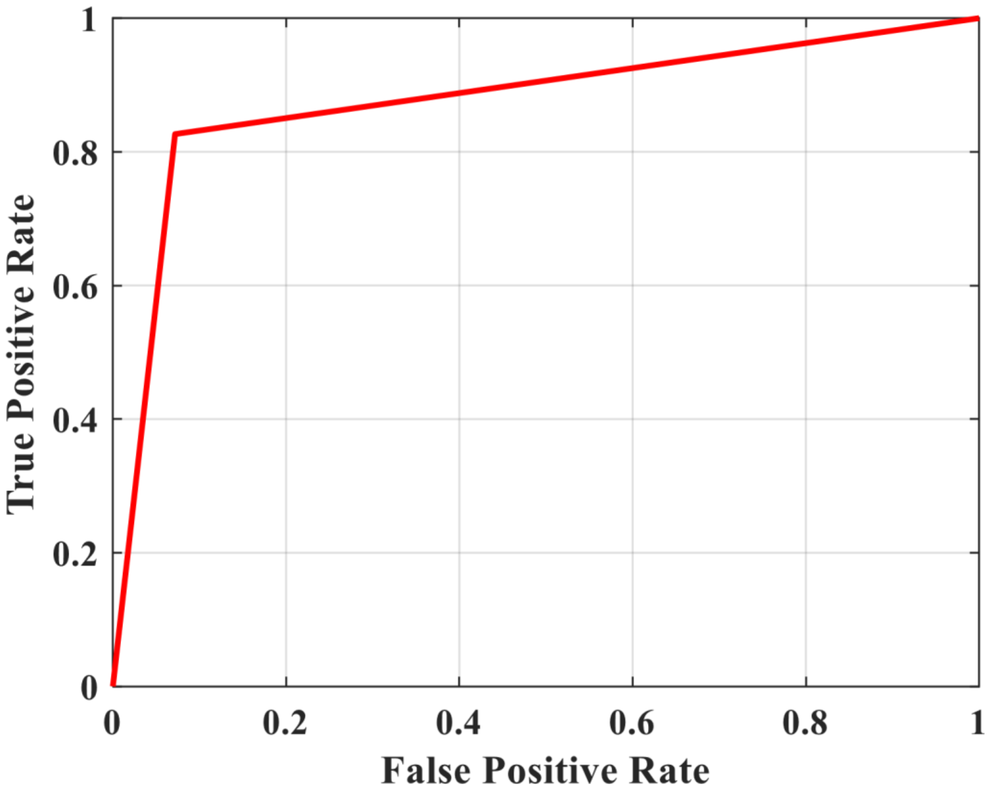


Figure 4.1 ROC Curve

The Table 1 shows the values for the performance evaluation parameters. The measured parameters are accuracy, error, sensitivity, specificity, precision, false positive rate, f1 score, Matthews correlation coefficient and Kappa.

An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate False Positive Rate. The Figure 4.1 shows the obtained ROC curve for the proposed approach. The proposed approach obtained better ROC curve.

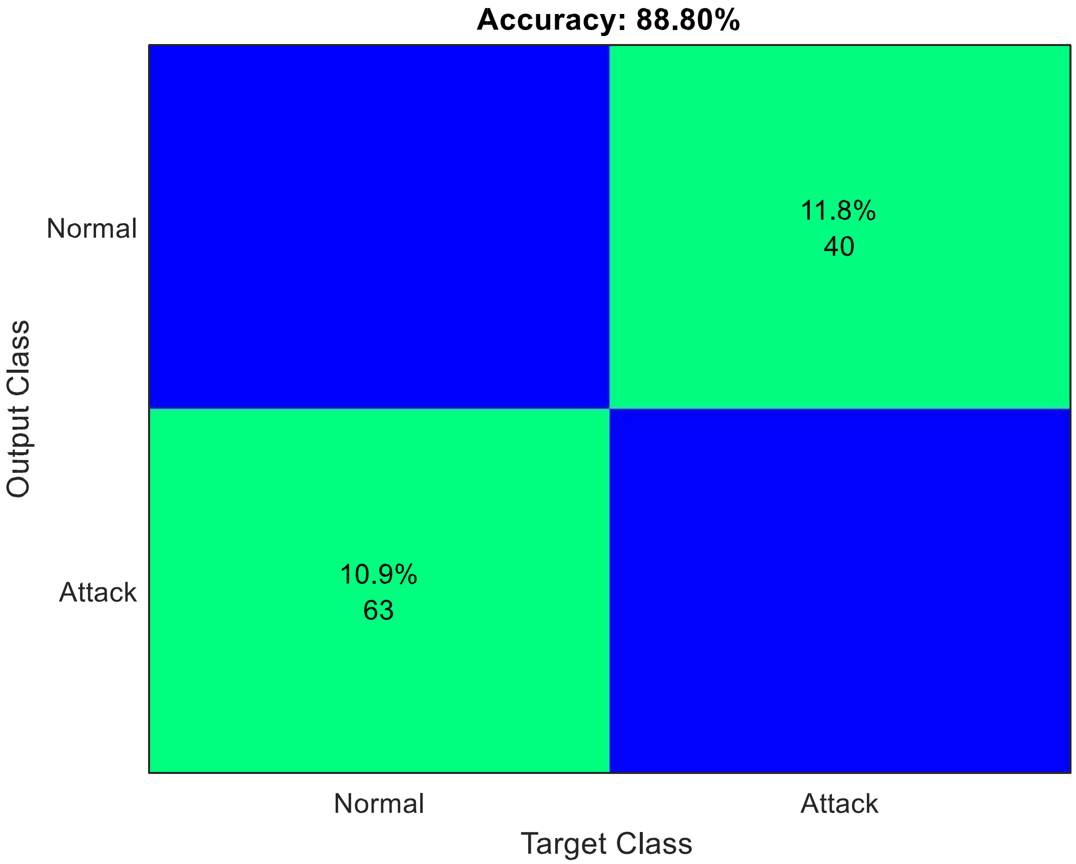


Figure 4.2 Accuracy

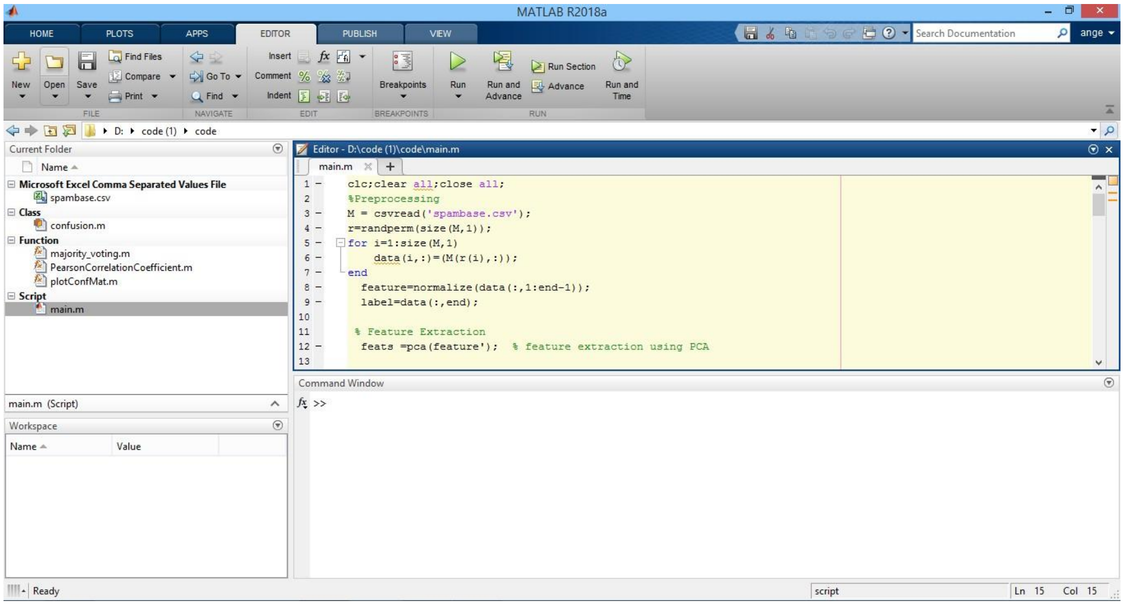
A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing.

The four indicators are presented together in a table, showing a confusion matrix, as shown in Figure 4.2. The confusion matrix counts the number. Sometimes it is difficult to measure the pros and cons of the model by simply counting the numbers. Therefore, the confusion matrix extends the secondary index accuracy (Accuracy) in the basic statistical results. Through the secondary index, the result of the quantity in the confusion matrix can be converted into a ratio between 0–1.

- Every row of the matrix links to a predicted class.
- Every column of the matrix corresponds with an actual class.
- The total counts of correct and incorrect classification are entered into the table.
- The sum of correct predictions for a class goes into the predicted column and expected row for that class value.
- The sum of incorrect predictions for a class goes into the expected row for that class value and the predicted column for that specific class value.
- Confusion matrix not only gives insight into the errors being made by your classifier but also types of errors that are being made.

4.4 SCREENSHOTS

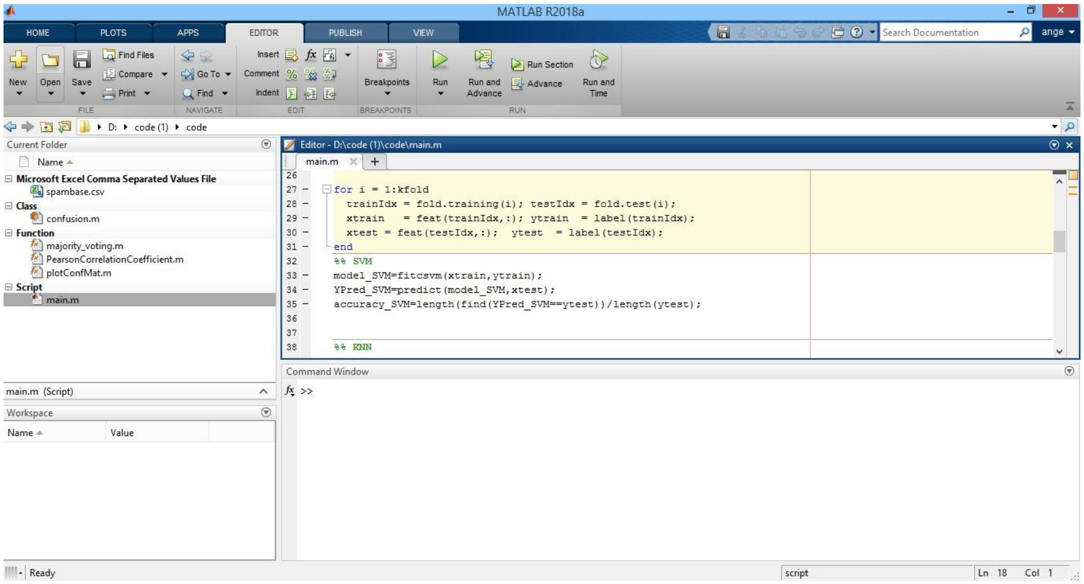
□ Pre-processing



```
  clc;clear all;close all;
  %Preprocessing
  M = csvread('spambase.csv');
  r=randperm(size(M,1));
for i=1:size(M,1)
    data(i,:)=(M(r(i),:));
end
  feature=normalize(data(:,1:end-1));
  label=data(:,end);

 % Feature Extraction
  feats =pca(feature');  % feature extraction using PCA

```

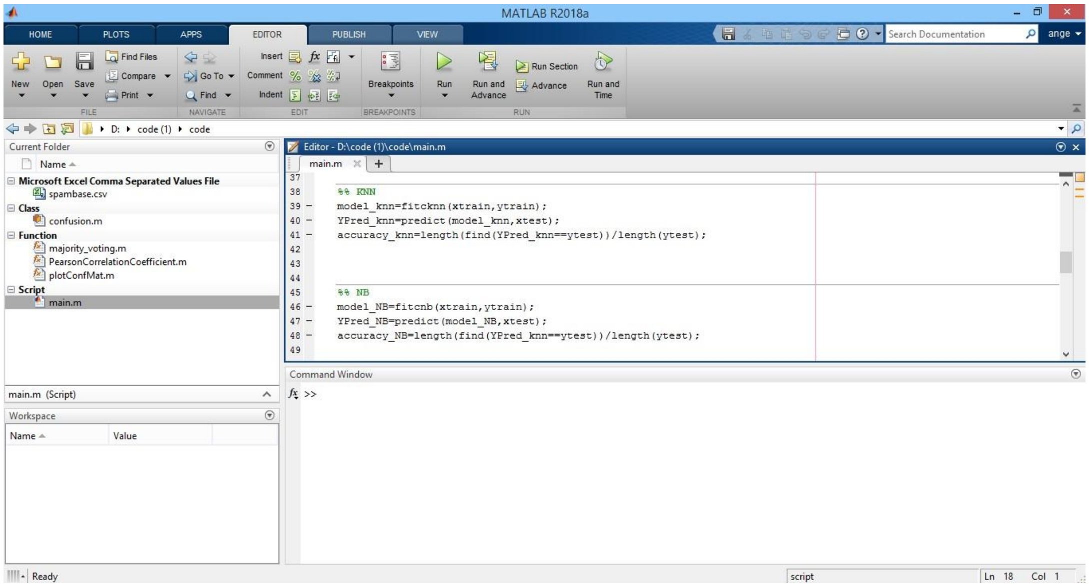
```
>>
```

□ Ensemble classifier



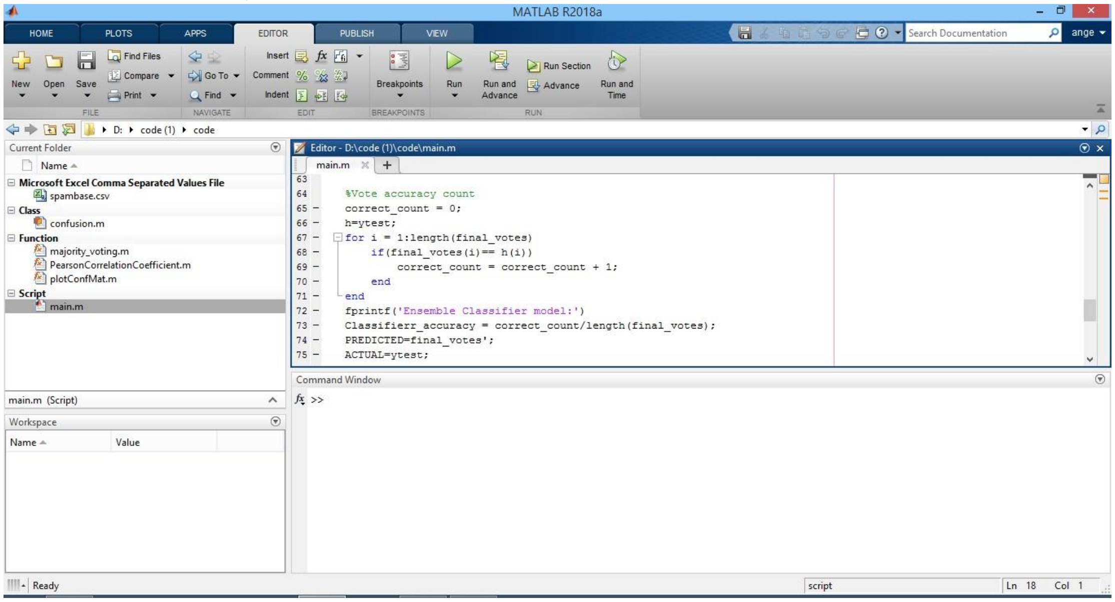
```

for i = 1:kfold
  trainIdx = fold.training(i); testIdx = fold.test(i);
  xtrain   = feat(trainIdx,:); ytrain  = label(trainIdx);
  xtest = feat(testIdx,:);  ytest  = label(testIdx);
end
%% SVM
model_SVM=fitcsvm(xtrain,ytrain);
YPred_SVM=predict(model_SVM,xtest);
accuracy_SVM=length(find(YPred_SVM==ytest))/length(ytest);


%% KNN
```

```
>>
```



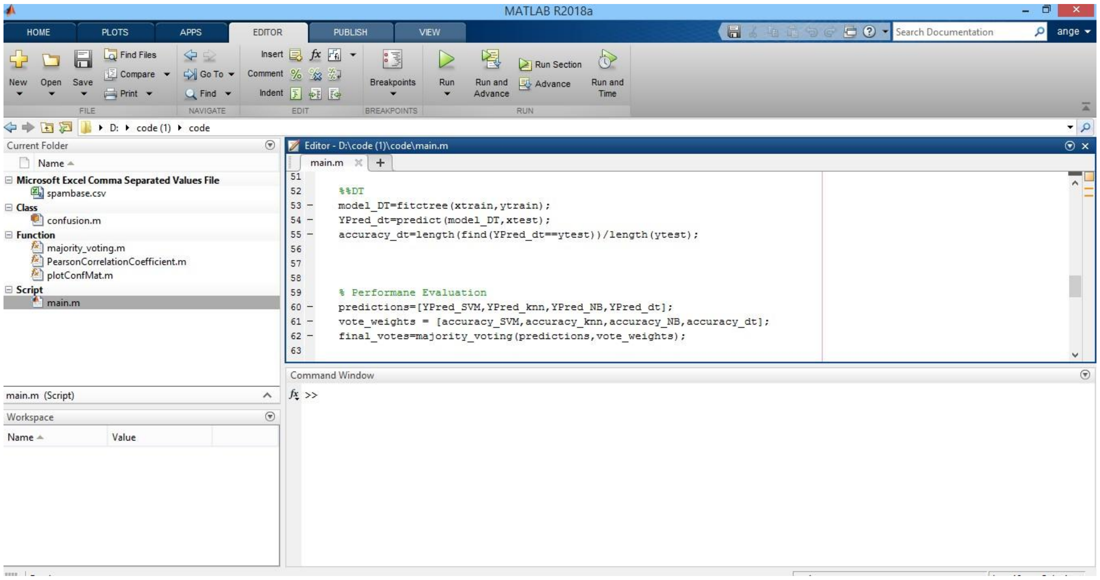
```
%% KNN
model_knn=fitcknn(xtrain,ytrain);
YPred_knn=predict(model_knn,xtest);
accuracy_knn=length(find(YPred_knn==ytest))/length(ytest);



%% NB
model_NB=fitcnb(xtrain,ytrain);
YPred_NB=predict(model_NB,xtest);
accuracy_NB=length(find(YPred_knn==ytest))/length(ytest);
```

- **Classifier accuracy**



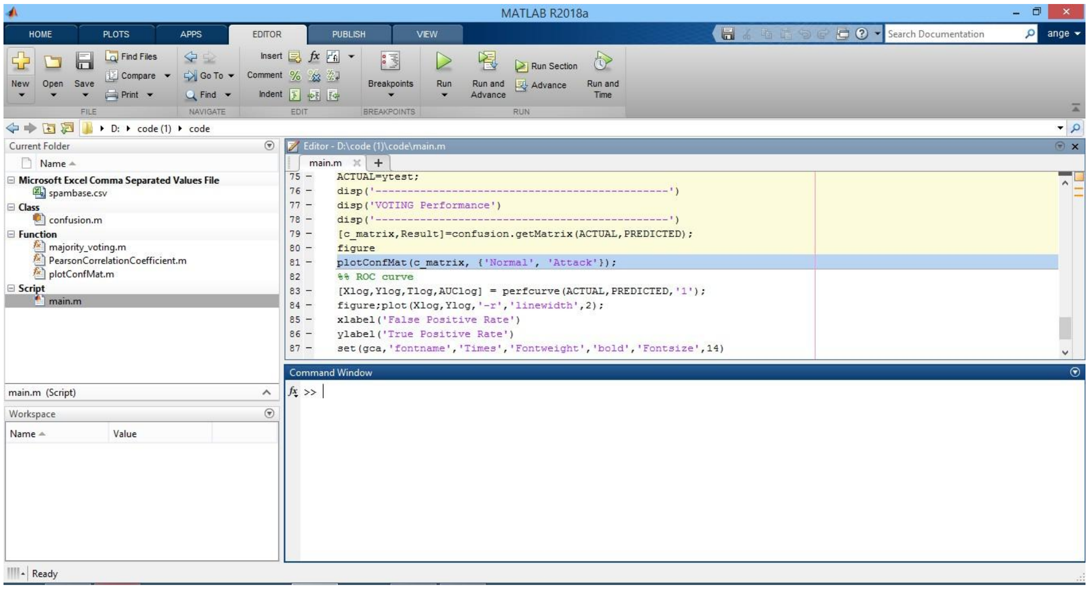
```
%Vote accuracy count
correct_count = 0;
h=ytest;
for i = 1:length(final_votes)
    if(final_votes(i)== h(i))
        correct_count = correct_count + 1;
    end
end
fprintf('Ensemble Classifier model:')
Classifierr_accuracy = correct_count/length(final_votes);
PREDICTED=final_votes';
ACTUAL=ytest;
```

- **Performance Evaluation**



```
%%DT
model_DT=fitctree(xtrain,ytrain);
YPred_dt=predict(model_DT,xtest);
accuracy_dt=length(find(YPred_dt==ytest))/length(ytest);



% Performane Evaluation
predictions=[YPred_SVM,YPred_knn,YPred_NB,YPred_dt];
vote_weights = [accuracy_SVM,accuracy_knn,accuracy_NB,accuracy_dt];
final_votes=majority_voting(predictions,vote_weights);
```

- **ROC curve**



```
ACTUAL=ytest;
disp('----------------------------------------------')
disp('VOTING Performance')
disp('----------------------------------------------')
[c_matrix,Result]=confusion.getMatrix(ACTUAL,PREDICTED);
figure
plotConfMat(c_matrix, {'Normal', 'Attack'});
%% ROC curve
[Xlog,Ylog,Tlog,AUClog] = perfcurve(ACTUAL,PREDICTED,'1');
figure;plot(Xlog,Ylog,'-r','linewidth',2);
xlabel('False Positive Rate')
ylabel('True Positive Rate')
set(gca,'fontname','Times','Fontweight','bold','Fontsize',14)
```

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

The major cause for the failure of IoT devices is due to the attacks, in which web spam is more prominent. There seems a requirement of a technique which can detect the web spam before it enters into a device. Motivated from these issues, a Ensemble Classifier based web spam detection technique is proposed. Random Forest, K-Nearest Neighbor, Support Vector Machine and Decision Tree classifiers are used in the Ensemble classifier. Each classifier produces the quality score of the webpage. These quality scores are then ensemble to generate a single score, which predicts the spam of the web page. The proposed approach is implemented using MATLAB2022a software. The proposed approach showed better results than existing approaches.

5.2 FUTURE WORK

There are a limited number of labelled datasets available for spam text, as well as a limited number of attributes available in these text datasets, which is a problem. For efficient research, a dataset with correct labelling is required, as is large computational power in the case of a large dataset. Only a few studies have used deep learning techniques and semantic approaches to detect spam. Exploring the use of multimodal content (text and images) from social media for social media would be a significant future challenge.

CHAPTER 6

APPENDIX

Sample Code

```
clc;clear all;close all;
%Preprocessing M = csvread('spambase.csv');
r=randperm(size(M,1)); for i=1:size(M,1)
data(i,:)=(M(r(i),:)); end
feature=normalize(data(:,1:end-1));
label=data(:,end);  % Feature Extraction   feats
=pca(feature');  % feature extraction using PCA
  % Feature Selection   % Parameters opts.K
= 3;     % number of nearest neighbors
opts.Nf = 10;    % select 10 features
% Perform feature selection
FS     = PearsonCorrelationCoefficient(feats,label,opts);
% Define index of selected features
sf_idx = FS.sf; feat=feats(:,sf_idx);
kfold  = 5; fold   =
cvpartition(label,'KFold',kfold); for i =
1:kfold   trainIdx = fold.training(i);
```

```
testIdx = fold.test(i);   xtrain   =
feat(trainIdx,:); ytrain  =
label(trainIdx);   xtest = feat(testIdx,:);
ytest  = label(testIdx);  end
%% SVM
model_SVM=fitcsvm(xtrain,ytrain);
YPred_SVM=predict(model_SVM,xtest);
accuracy_SVM=length(find(YPred_SVM==ytest))/length(ytest);
%% KNN
model_knn=fitcknn(xtrain,ytrain);
YPred_knn=predict(model_knn,xtest);
accuracy_knn=length(find(YPred_knn==ytest))/length(ytest);
%% NB
model_NB=fitcnb(xtrain,ytrain);
YPred_NB=predict(model_NB,xtest);
accuracy_NB=length(find(YPred_knn==ytest))/length(ytest);
%%DT
model_DT=fitctree(xtrain,ytrain);
YPred_dt=predict(model_DT,xtest);
accuracy_dt=length(find(YPred_dt==ytest))/length(ytest);
% Performane Evaluation
predictions=[YPred_SVM,YPred_knn,YPred_NB,YPred_dt];
vote_weights = [accuracy_SVM,accuracy_knn,accuracy_NB,accuracy_dt];
final_votes=majority_voting(predictions,vote_weights);
%Vote accuracy count correct_count
= 0; h=ytest; for i =
1:length(final_votes)
```

```
if(final_votes(i)== h(i))
correct_count = correct_count + 1;
end end
fprintf('Ensemble Classifier model:')
Classifierr_accuracy = correct_count/length(final_votes);
PREDICTED=final_votes'; ACTUAL=ytest;
disp('--------------------------------------------') disp('VOTING
Performance')
disp('--------------------------------------------')
[c_matrix,Result]=confusion.getMatrix(ACTUAL,PREDICTED);
figure plotConfMat(c_matrix, {'Normal', 'Attack'});
%% ROC curve
[Xlog,Ylog,Tlog,AUClog] = perfcurve(ACTUAL,PREDICTED,'1');
figure;plot(Xlog,Ylog,'-r','linewidth',2); xlabel('False Positive Rate')
ylabel('True Positive Rate')
set(gca,'fontname','Times','Fontweight','bold','Fontsize',14) grid on
```

REFERENCES

1. C. Chen, J. Zhang, Y. Xie, Y. Xiang, W. Zhou, M. M. Hassan, A. AlElaiwi, and M. Alrubaian, “A performance evaluation of machine learning-based streaming spam tweets detection,” IEEE Trans. Comput. Social Syst., vol. 2, no. 3, pp. 65–76, Sep. 2015.

2. L. Liu, O. De Vel, Q.-L. Han, J. Zhang, and Y. Xiang, “Detecting and preventing cyber insider threats: A survey,” IEEE Commun. Surveys Tuts., vol. 20, no. 2, pp. 1397–1417, 2nd Quart., 2018.

3. G. Lin, N. Sun, S. Nepal, J. Zhang, Y. Xiang, and H. Hassan, “Statistical Twitter spam detection demystified: Performance, stability and scalability,” IEEE Access, vol. 5, pp. 11142–11154, 2017.

4. A. Singh, N. Chahal, S. Singh and S. K. Gupta, "Spam Detection using ANN and ABC Algorithm," 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2021, pp. 164-168, doi: 10.1109/Confluence51648.2021.9377061.

5. Madisetty, Sreekanth; Desarkar, Maunendra Sankar (2018). A Neural Network-Based Ensemble Approach for Spam Detection in Twitter. IEEE Transactions on Computational Social Systems, (), 1–12. doi:10.1109/TCSS.2018.2878852.

6. Kontsewaya, Y., Antonov, E., & Artamonov, A. (2021). Evaluating the effectiveness of machine learning methods for spam detection. Procedia Computer Science, 190, 479-486.

7. Shahariar, G. M., Biswas, S., Omar, F., Shah, F. M., & Hassan, S. B. (2019, October). Spam review detection using deep learning. In 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) (pp. 0027-0033). IEEE.

8. Dhawan, S. (2018). An enhanced mechanism of spam and category detection using Neuro-SVM. Procedia computer science, 132, 429-436.

9. Kumar, N., Venugopal, D., Qiu, L., & Kumar, S. (2018). Detecting review manipulation on online platforms with hierarchical supervised learning. Journal of Management Information Systems, 35(1), 350-380.

10. Liu, Y., Pang, B., & Wang, X. (2019). Opinion spam detection by incorporating multimodal embedded representation into a probabilistic review graph. Neurocomputing, 366, 276-283.