REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

SMS SPAM FILTERING ON MOBILE COMMUNICATION

ISRAA HUSSAIN

MSc. THESIS
DEPARTMENT OF COMPUTER ENGINEERING
PROGRAM OF COMPUTER ENGINEERING

ADVISER
ASSOC. PROF. DR. SIRMA YAVUZ

İSTANBUL, 2017

REPUBLIC OF TURKEY

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

SMS SPAM FILTERING ON MOBILE COMMUNICATION

A thesis submitted byIsraaHUSSAINin partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee on 12.06.2017 in Department of Computer Engineering,Computer Engineering Program.

Thesis Adviser

Assoc.Prof.Dr. Sırma YAVUZ

Yıldız Technical University

Approved by the Examining Committee

Assoc. Prof. Dr. Sırma YAVUZ _____

Prof. Dr. Hasan HUSEYIN

Yıldız Technical University _____

Assoc. Prof. Dr. Metin ZONTUL

Istanbul Aydin University _____

ACKNOWLEDGEMENTS

Foremost, I thank God for making everything possible to me.

I would like to express my heartfelt gratitude to my supervisor, Assoc. Dr. SirmaYavuz for her guidance and assistance throughout the research. Furthermore, my appreciation goes to my beloved family (parents, sisters, and brothers) for their love, sacrifices, encouragement, and moral support especially my father.

Additionally, I would like to express my deepest appreciation and thanks to the academic staff of Computer Engineering Department, Electrical and Electronics Engineering Faculty in Yildiz Technical University. Finally, yet importantly, I would like to thank all my friends, my classmates, and all of those who have helped me directly or indirectly during the course of this work.

Jul, 2017

IsraaHUSSAIN

1.1 Literature Review

Nowadays and all over the world, the use of Short Message Service (SMS) has an indispensable roll in people's life's communications in many fields. With the increase use of the Internet services, e-mails and mobile phones, the use of SMS is also growing. SMS were reported to be part of a massive commercial industry that worth billions of dollars globally [1]. However, unsolicited messages sent over the internet known as electronic spam were also produced. These spam messages were sent to a large number of users for different purposes such as commercial advertising, phishing, spreading, etc. Spam started with e-mails as it was the primary communication technique since the early 1990s. On the other hand, SMS spam is junk messages delivered as text messaging via SMS to cell phones. SMS spam has become one of the most superb forms of spam ever since its purchase. Numerous efforts by researchers and free-mailing/message service areas such as; yahoo mail, Gmail, mobile phone, etc. have been adopted in order to build efficient spam filters against e-mail or SMS spam. However, spam filters are still failing to arrest SMS spam. Understanding the techniques used by spammers and processing a set of features that can help in identifying of SMS spam from normal messages are still under focus by researchers [2].

In the last decade, different attempts have been reported concerning the use of different approaches to classify SMS spam by different machine learning techniques. In 2010, Yoon et al. proposed an anti-spam approach based on a hybrid filtering framework. They have used a combination of content-based filtering and a challenge-response. First, a message that is classified by content-based filter as to be UNCERTAIN was further checked via a challenge-response by sending the message back to the sender. Hence, when the spam generator is unable to send back a correct response, the message would be classified as spam. Their results suggested that this hybrid technique is capable of utilizing high accuracy regardless of the algorithm used by the content-based filter [3]. In 2014, Ahmed et al. proposed another hybrid SMS message classification system to differentiate between spam and ham, using Naïve Bayes classifier and Apriori algorithm. They performed SMS collection, feature selection, vector creation, filtering process and system updating. Their results were reported to produce significant improvement of 98.7% effective accuracy from the traditional Naïve Bayes [4]. In 2011,

Nuruzzaman et al. proposed SMS spam filter on an independent mobile phone using Naïve Bayes and Word Occurrences table without the need for additional compute system support. Obtaining reasonable accuracy, low storage consumption and acceptable processing time were the main advantages reported by the authors. They have claimed that such an approach may ensure security and privacy since spammers have no chance to enter the filtering system and users have no SMS storage. Besides, they proposed that the performance based approach can significantly reduce the number of words in the word occurrences table without significantly compromising their accuracy [2].Anchal& Sharma in 2014 proposed different hybrid approach in text classification using Principal Component Analysis (PCA) algorithm, and combination of ICA/Custom Neural Network algorithms. They adopted their work in MATLAB environment. They reported that this technique could be used to reduce the number of variables in regression and clustering using trained dataset. The results of their proposed technique were recorded to be promising in achieving an average rate of Accuracy, Recall, Precision, and Specificity [5].

In 2012, Uysal et al. introduced a novel framework to arrest unsolicited SMS messages. They first used two well-known ranking measures: information gain (IG) and chi-square metrics (CHI2) to rank all terms in the collection. Later, the classification of messages to be spam or not was performed using two distinct feature selection approaches based on different Bayesian based classification algorithms, namely the binary and probabilistic models. They have reported that this approach was highly accurate in differentiation between both spam and legitimate SMS messages [6].Mahmoud& Mahfouz (2012)proposed an anti-spam technique using Artificial Immune System (AIS) for filtering spam messages. The system used AIS features to produce detectors, by initial training phases. In the training process, messages were classified using the trained dataset which contains several features such as; Phone Numbers, Spam Words, and Detectors. A set of stages were stated in order to help in building dataset such as tokenizer, stop word filter, and training process. The proposed technique when compared with Naïve Bayesian algorithm showed high accuracy in classification of SMS spam or ham [7]. In 2015, Al-Hasan & El-Alfy explored a number of content-based feature sets to enhance the mobile phone text messaging services in filtering unwanted messages (spam). They used a combination of the most relevant features and by fusing decisions of the two machine learning algorithms used (Naive Bayes and

Support Vector Machines) with the Dendritic Cell Algorithm (DCA) model. The performance was evaluated empirically using two SMS spam datasets. The proposed DCA-based model resulted in a significant improvement in the overall accuracy, recall, and precision of spam and ham messages[8].

Nagwani&Sharaff(2015)based their work in SMS spam detection and thread identification on art clustering-based algorithm. The work was setup in two stages. In the first stage, the binary categorization of SMS messages as spam and non-spam SMS was applied. In the second stage of their work, SMS clusters were performed for non-spam messages using non-negative matrix factorization with K-means clustering techniques. A threading-based similarity feature, that is, time between consecutive communications, was described for SMS threads identification [9]. In the same year, Kim et al, proposed a simple method of filtering SMS messages within a mobile device using a Frequency Ratio (FR) measure. Firstly, the spam and ham were divided and the appearance frequencies of words on the messages were evaluated. Then the appearance frequencies of each word were aggregated and divided by the number of messages in order to calculate the average. They performance of this method was compared with Naive Bayes, J-48 Decision Trees, and Logistic, suggesting that the FR measure has a simple formula that could create indexes quickly [10]. Mujtaba and Yasinin 2014 used four features of the SMS messages; the size of the message and existence of frequently occurring monograms in the message, existence of frequently occurring diagrams in the message and message class to be used in trained machine learning algorithms to identify and remove spam messages as soon as it is received at the mobile phone. The reported results concluded that Naïve Bayes algorithm has better performance than Artificial Neural Networks and Decision Tree classifier [11].

1.2Objective of the Thesis

Every year, and all over the world, electronic industries marketing different types of mobile phones. The increase use of mobile phone attracted by their modern and multiple applications encourages not only users to use SMS messages but also SMS spammers. Hence, spam has increasingly become part of mobile users' life. SMS spam messages besides being distracting and annoying to mobile users, they quickly fill up message inboxes. And as a consequence, users waste their time to open up, read, and delete those messages. The main difficulty in dealing with the spam is their

differentiation from other legitimate messages. SMS messages can be classified by number of techniques such as; Naïve Bayes, neural networks, rule induction, K-nearest neighbors, and Support Vector Machine. Many algorithms have been proposed by the researchers in order to reach themost demanded accuracy in message classification. However, as part of electronic industry, spammer can come up with new ways to send their spam. Hence, the researches in this field are an ongoing without ending processes. Therefore, the aim of this study was tobuild up a set of feature selection methods that deal with three machine learning algorithms; Artificial Neural Network, K-Nearest Neighbors and Naïve Bayes to achieve better SMS messages classifications.

1.3Hypothesis

Depending on the features that were allocated for this purpose (Figure 1.1) and using specific dataset of SMS messages to solve spam problem and try to improve the classification results. Comparative analysis of the experimental results was performed among them and with other researches.

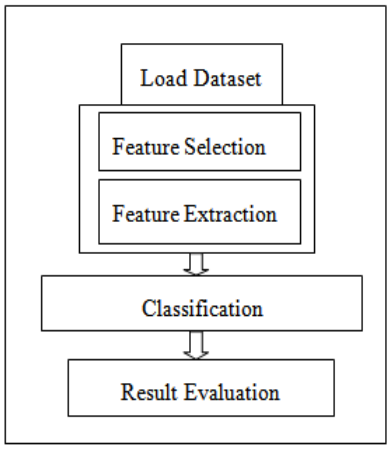


Figure 1.1 Architecture of SMS Filtering System

1.3.1 Objectives:

Theobjectives of the current study included the followings:

- To extract a new set of features selection methods for the SMS classification.
- To compare between the three classifiers used in the research: Artificial Neural Network, K-Nearest Neighbors and Naïve Bayes.
- To enhance the Accuracy values extracted by the three clasifiers using featureselection methods (Information Gain & Information GainRatio).

- Tocompare the results of each classifier with each other and with more or less similar recent studies.
- To discuss the reasons of the extracted results of each method.

1.3.2 Thesis Organization

The current thesis was organized as follows: Chapter 2; includes a general introduction concerning SMS spam on mobile network and the filtering techniques in particular the three techniques that have been under focus in this research study. InChapter 3, detailed explanation was provided about feature selection algorithms including information gain and information gain ratiotechniqueused in this workwiththe three classification methods used which were; Artificial Neural Network, Naïve Bayes and K-Nearest Neighbors algorithm.In Chapter 4, the algorithm results were discussed in detailsand compared with other researchers. In Chapter 5, the extracted conclusions of the current research were identified and some ideas for future works were suggested.

CHAPTER 2

GENERAL INFORMATION

2.1 Overview

Internet service has become one of the most discussed technologies in the recent years. But what is a Web service? In a simple term, a Web service is an application that can provide a specific set of functions to achieve specific endings. It can provide communications to application without any human assistance or intervention. Web service gives companies an unprecedented opportunity to e-business. Shopping on the internet may help patients who have access to medical services on the internet from their mobile phones. Considering all these facilities, the internet services have become part of every day's life [12]. However, and as part of the modern modalities in communications, cell phones have become the most widely used mean of communication worldwide. Between the years 2005 and 2010, subscribers using cell phone reached up to more than 5 billion. Although this kind of communication could be influenced by the economy and the living standard of each country. It has been seen that at the end of 2010, subscription levels in developed countries were greater than 100%, while in some developing countries reached 73% of the population. Short messaging service (SMS) using cell phone is one of the simplest and most convenient modalities in communications that can be used by the users [13]. Hence, cell phones have become the biggest target of spammer using short text messages spam via short text messages (SMS). SMS spam has been used to damage smart phones and became widespread because of the high cost of SMS to spammer messaging charges dropped below US $0.001 in China and free of charge in others because of the popularity of SMS in the world [14].SMS is used to able customers to connect with each other through sending and receiving instant messages by mobile phones. New products, information and advertisements are widely marketed by sending messages to customers. This marketing

techniques are the newest media market industry. Cell phone messages are very useful for communication between customers; adults and kids by sending and receiving documents and data. It also shortens the distance between customers around the world [10].

2.2 Short Message Service (SMS)

SMS was known to be a key to deliver text information on signaling channels needed by Global System for Mobile (GSM). Throughout the years, SMS was developing to give better services for customers. Voice calls were first introduced later short message services were spread dramatically. The limitation of text message characters raised from 128 up to 140 then to 160 characters [2]. In addition, international encoding in SMS included different languages such as; Arabic and Chinese [2]. Then, pagers were introduced. More recently, SMS replaced the pager technology. SMS nowadays include pictures, sounds, animations and files. SMS has been very successful since its introduction. It has been reported that about 5 billion messages are being sent every day [15]. Although there are other ways of communications such as emails and chatting. The use of SMS as an active mean of personal communication has contributed to the expansion of the text messaging marked with 2.4 billion active users. About 74% of all mobile phone subscribers send and receive text messages on their phones. SMS not only provides a useful way for a range of innovative services over mobile networks but also serves as a point of entry for new data services such as wireless application Protocol (WAP) [16].

2.2.1 The Applications of SMS

The applications of short message are so wide and various including different aspect of outlives including education and health services. The educational information like the enrollment of information and Library services that can be improved through SMS-based administrative support. While, health services include communication technology interventions between the patients and their medical care employees to facilitate and provide better and faster health care.

General applications of SMS can be summarized in the followings:

1. Regularly utilized as an approach to ready representatives if there should be an occurrence of issues.

2. As an apparatus for Teaching and Learning Support.
3. Over-The-Air (OTA) programming: Provisioning data and phone settings.
4. Electronic installment: Premium-rate SMS services.
5. Notifications: e.g. voicemail and Fax messages.
6. Exchange with different conventions: e.g. email/Fax gateways.
7. Two-Factor Authentication: TANs in SMS, e.g., for online banking.
8. Monitoring [16].

2.2.2Spam Message

Generally, spam referring to the arbitrarily sending of unasked messages to a large group of addresses. Though e-mail spam is a wide known form of spam, it has become popular in many other electronic media such as; social networks, chatting, internet telephony, web spamming, etc.

Although, spam might be considered as a modern problem, in fact, it is related back to 1975[17]. The cost of transmission of these messages is borne by the users who receive it and the Internet Service Provider (ISP) who cannot help the spam traffic and are forced to increase bandwidth to accommodate the traffic [18]. The spammers only need to manage the lists of their mails that they target.

Some common examples of spam are:

- Advertisements that are in the form of pop-ups selling products or giving free downloads when clicking any link on a web page.
- Unsolicited e-mails with inappropriate context, purchasing offers, political views, etc.
- Redundant calls on IMs like Skype offering mortgages, loans with low interest rates.
- Links on social networks that leads to free downloads, and easy income pornography [19].

Spam can be stopped at two points in the network:

- From the service provider network.
- The system (users' mobile devices).

However, SMS spam filter is hard to control and spam are easy to spread in the network operator because cell phone network operator cannot control what customers agreed to receive and accept [13].

2.2.3 Ways to stop receiving spams:

There are several ways to avoid receiving spam like:

- Hiding email addresses: Hiding the email address will keep the spammer away.The email address can be set to be known to trusted parties only.
- Pattern matching; whitelist and blacklist. Incoming emails can be matched where friends' emails(whitelist)are matched against unknown emails (blacklist). However, this method needs frequent updating of the lists every now and then. Besides, it may stillsometimes give a false matching.
- Rule based filters: It developed by spam filtering software and applying a set of rules to every incoming email. The rules are built up by regular data and software. It must be updated periodically[20].
- Statistics filters: Statistics filtering requires training on both spam and no spam emails, and then filter the reported spam messages.
- Email verification: Email verification is hard to pass, it sends out one verification email to pass the system. The only way for it to pass the filter is if the sender responds to the challenge email.
- Distributed blacklists of spam sources; these filters use a blacklist to verify whether or notthe received email is spam. The blacklist keeps a record of known spam sources such as IP address.
- Distributed blacklist of spam signatures: This blacklist has the characteristic of distributed blacklist of spam sources but the difference is that these blacklists contain spam signature not the sources. One customer can warn many others. But the disadvantage of it is that the spam fighters are not able to recall all the spam.
- Money Email Stamp: The sender has to pay fees for the stamp and it's not a lot amount of money. There are two types of stamps:

a. Money Stamp

b. Proof of work stamps. But there are problems with this method because it raises money questions like who benefits from this money? Who's allowed to sell it?

- Proof of work email: Microsoft announced in 2004 that spam problem could be solved with two years by adding proof of work stamp to each other. A cheat proof mathematics puzzle is send instead of the fee to the sender when a solution is found it is sent back to the sender and the email is allowed to be send to the receiver.
- Legal measures: Spam law is widely used these days. Laws dose not completely stop spammers but at least slow them done. It needs international movements [20].

2.3 Machine Learning

Machine learning is programming computers to advance in execution foundation utilizing illustrated information or past experiences. Besides, Machine learning utilizes the hypothesis of measurements in building scientific models, in light of the fact that the center undertaking is making deduction from an example.

A machine designer when produces machines often do not work as supposed to or desired in the environments in which they are used. It has been found that not all the characteristics of the working environment can be completely recognized at the designed time. Usually, machine learning methods are used for improving on-the-job work of the already existing machine designs. The amount of knowledge periodically appearing about certain tasks is in fact too large for human to explicit encoding. Redesigning of AI systems constantly to conformtoany new knowledge is time consuming uneconomical and impractical.Machine learning methods have proven to be able to deal with many of these situations.Therefore, machines that are adopted to identify new knowledge might be able to handle it better and faster than humans. Similarly, the rapid changes in theenvironments over time can be better adjusted by changing environment adapted machines without the need for periodic redesign [21].

There are two methods to classify the texts; either supervised or unsupervised.

- Unsupervised Learning

In this way, the training consists only an input vector without displaying the output on the network and this method called learning without predefine output

- Supervised Learning

In this way, the training consists of two types of vectors; the input to network and output of network. Training is done by adjusting the weights. This method is termed learning with predefine output [22].

2.3.1Feature Extraction

A feature in computer engineering isdefined asany characteristic detail or aspect of something that is used within a computer program. These features may represent the words or the terms that are present in any written document.

Space is the original feature of changing to a new place more compact. All the original features converted to drop a new space without deleting and replacing the original features, but by a reduced representative group. This is when the number of feature in a very large data entry so that cannot be processed and will change the data input to the low representation of a grub of features.

2.3.2 Feature Selection Algorithm

As a generally utilized procedure in information preprocessing for machine learning, feature selection distinguishes imperative features and expels insignificant, excess or commotion features to diminish the dimensionality of feature space. It enhances productivity, exactness and fathomability of the models worked by learning calculations. Feature selection procedures have been generally utilized in an assortment of utilizations, for example, genomic examination, data recovery, and content order[23].

The decision for choosing the specific learning algorithm to be used is a critical one. A preliminary testing with satisfactory results is needed in order to identify the classifier (mapping from unlabeled instances to classes) that can be set for routine using.

A typical method of selecting feature consists of four basic steps:

- Subset generation.
- Subset evaluation.
- Stopping criterion and.
- Result validation [24].

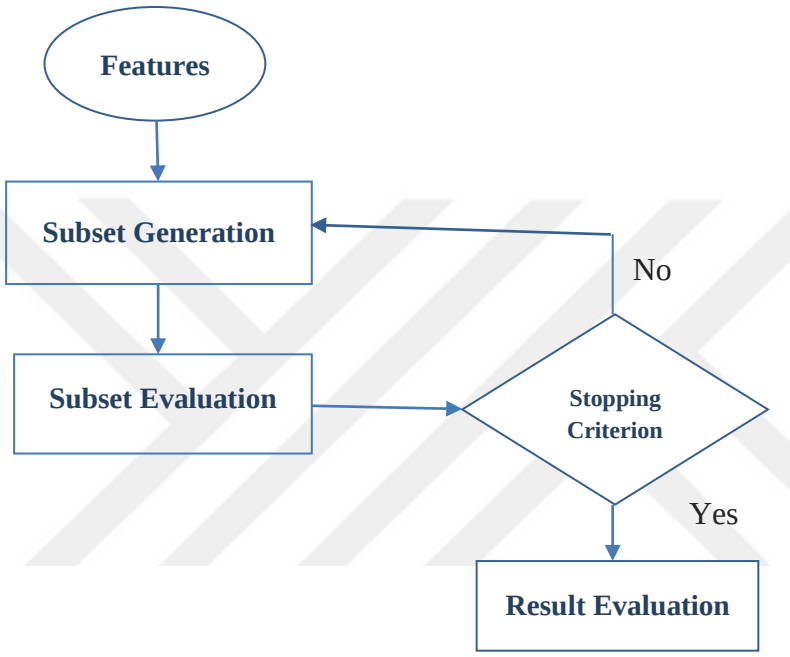


Figure2.1 Procedure to Select the Feature [24].

Figure 2.2. Represents some of the techniques that have been used by researchers for feature selection.

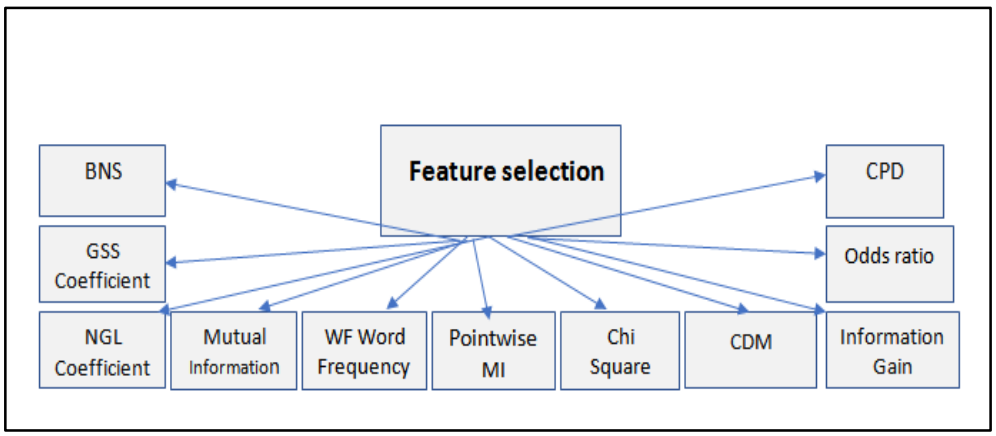


Figure 2.2 Supervised Feature Selection Techniques

After feature subsets are created, they are assessed by a specific model to gauge their integrity. For the most part, the integrity of feature subsets implies the segregating capacity of subsets to recognize among various classes. In view of whether they are subject to the inductive learning calculations, feature selection calculations can be comprehensively partitioned into three classes:

- Wrapper:
- Filter.
- Hybrid.

Generally, wrappers require broader calculations than filters. This is due to over and over calling of the enlistment calculations to assess every competitor included in a subset[21].

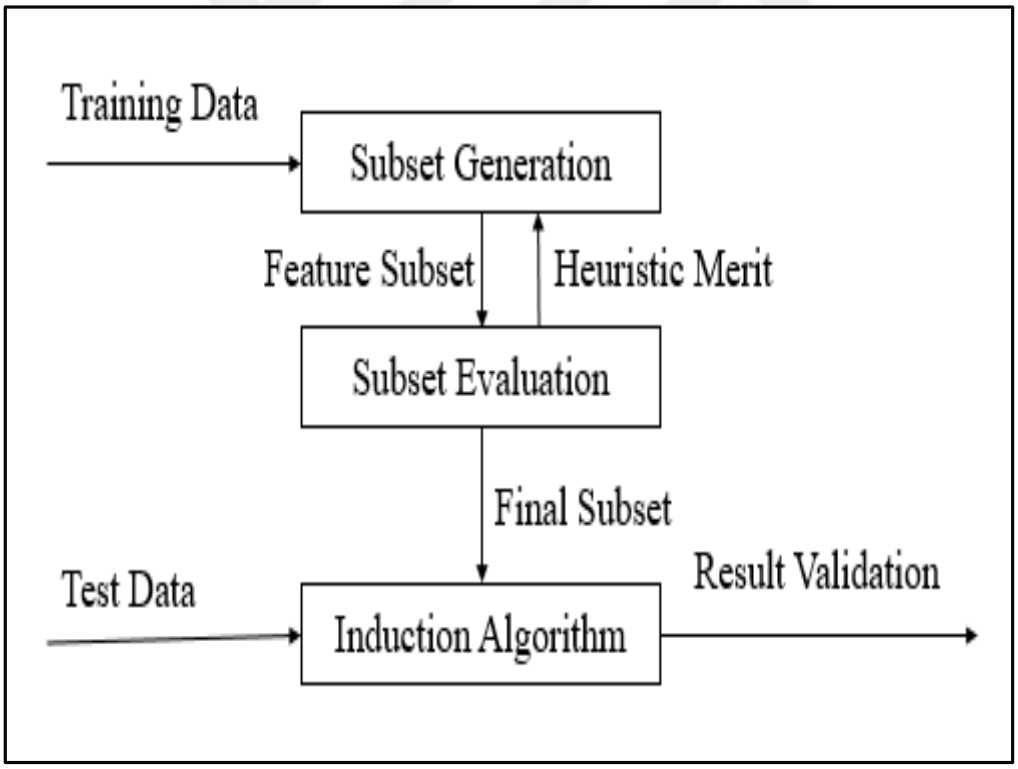


Figure 2.3 The Filter Approach for Feature Selection[24]

2.4Classification

2.4.1A Model Classification

Machine learning technique became widely popular during the nineties. In this technique, artificial intelligence is used to make classifiers that can mechanically assign category labels to documents. The classifiers are trained on manually reclassified documents called training documents. This technique is preferable because of faster re-training when the categories or other aspects change. Because of limited computing power however, it was not feasible before the late eighties.

Classifier is a mapping of examples to anticipated classes. Some grouping models create a constant yield to which distinctive limits might be connected to foresee class participation. Different models deliver a discrete class mark showing just the anticipated class of the occasion. To recognize the genuine class and the anticipated class, {Y, N}names were utilized for the class forecasts created by a model. Expression categorization is performed by classifiers to classify into one of the already defined [25].

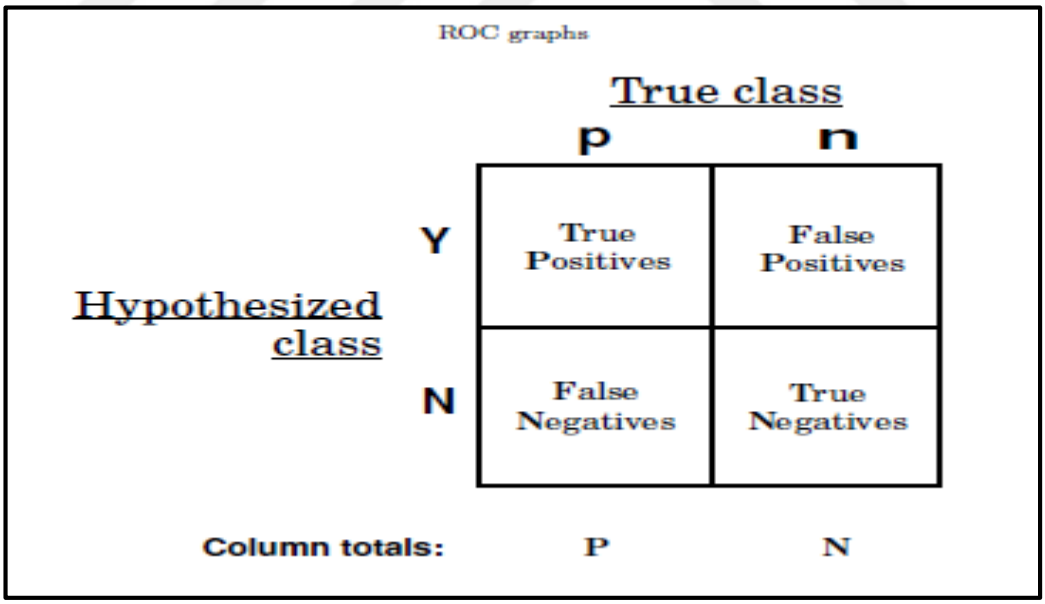


Figure 2.4 Parameter Matrix

2.4.2 The Approaches of Classifiers

There are three approaches to start a classifier:

1. Model a classification rule directly(discriminative classification) such as: K-NN, SVM,decision trees, and perceptron.

2. Model the probability of class memberships by giveninput data(discriminative and probabilistic classification) such as: perceptron with the cross-entropy cost.

3.Make a probabilistic model of data within these classes (generative and probabilisticclassification)such as: model based classifiers and Naive Bayes.

Text Classification techniques have five steps:

- Data Set and Running Environment,
- Feature Selection,
- Vector Construction,
- Filtering Method (Naïve Bayes or K-NN).
- Updating Filtering Structure.

2.5 Classifier Evaluation

The performance of a classification model can be evaluated by many metrics such as accuracy, recall, F-measure and precision.

In addition to that, time can be considered an important factor for filteringefficiency that has been performed by many researchers.

Each of these metrics was calculated in specific equations as follow:

$$\text{Accuracy} = \frac{\text{TPi} \pm \text{TNi}}{\text{TPi} + \text{TNi} + \text{FPi} + \text{FNi}} \qquad (2.1)$$

$$\text{Recall} = \frac{\text{TP}}{\text{P}} \tag{2.2}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{TF}} \tag{2.3}$$

$$\text{F} - \text{measure} = \frac{2}{1/\text{precision} + 1/\text{recall}} \tag{2.4}$$

$$\text{FP rate} = \frac{\text{FP}}{\text{N}} \tag{2.5}$$

Where (FP) all negatives incorrectly classified and (N) including the total negatives.

$$\text{TP rate} = \frac{\text{TP}}{\text{P}} \tag{2.6}$$

Where (TP) including all positives correctly classifiedand (P) including the total positives[24].

2.5.1 The Classifiers and the Instances

The instance could be either (+ve) or (-ve). However, when an instance is (+ve)andits classifiers (+ve)eventually it is considered as a True Positive (TP).While, when the instance is (+ve) and the classifiers (-ve), it is counted as a False Negative (FN).
If the instance is (-ve) and it is classified as (-ve), it is counted as a True Negative (TN), but if its classified as positive, it is counted as a False Positive (FP)[24].

2.6 Artificial Neural Networks

Neural networks are known as neural networks because these networks are formed from interconnected components [26]. Those components have the characteristics of a biological nervous system. That means neural networks are used to build machines equivalent to human's brain. The main role of neural system is producing an output pattern as a result of the input patterns. Pattern classification is one form of the neural network. It is the process of sorting a pattern into one group.

A new study in the field of Artificial Network system (ANN) is formed from a new neural network algorithm and analog VLSL implementation techniques [26]. ANN is

formed using digital or hybrid dedicated electronic or visual hardware [27]. The performance of Analogy optoelectronic hardware of ANN since it's been introduced in 1989 has been on the spotlight for many reasons. The main reason is that optoelectronics or the visual approach contains huge interconnectivity, parallelism and flexibility of the optics. Capability offered by electronics. It appears to be more effective to shape analog neural hardware by full optical characteristics, where switching signal forms optical to electronic carrier and the other way around. Because ANN is inspired by biological nervous system, it's necessary to study the brain and how it's processes.

2.6.1 Brain's Analogy

The main part of neural network is designed after the structure of the brain, however, some of them are not similar to the brain and some do not even have a biological characteristic of the brain [28]. Human nervous system contains billions of neurons and synapses;about10^{11}neurons and 10^{14} synapses [29]. Therefore, the human's brain can notice objects, colors and details but computers with powerful software and network cannot compete with it. Computers are designed of one processor that take orders and commands written by programmers [30]. Neural networks are networks of simple computing devices which are designed after the architecture of biological nervous systems. The biological nervous system is briefly described in order to understand the artificial neural networks [27]. Neuron is the key element in the nervous system especially to the brain. It is a unit that consist cell body and its processes. Neuron receives signals from many neurons by dendrites, the dendrites then activate the neurons reaction if the signals are too strong, then neuron create an output signals to the axon. Some of the main characteristics of human neural networks are described in Table 2.1 [27].

Table 2.1 The Characteristics of Human Neural Network

The Number of neurons	10^{11} –10^{13}
The Number of connections	10-10000 Single

The Number of Afferent (Input)	10 percent
The Number of Efferent (Output)	90 percent
The Storage Capacity	10^{13}-10^{15} Bits
The Average Brain Weight	1.5 kilo gram
The Average Neuron Weight	$1.2*10^{-12}$ kilo gram
The Refractor Period	10^{-2} second
The Firing Frequency	50-100 Spikes/s
The Synapses	Excitatory and Inhibitory
The Membrane Potential	The Triggers Firing
The Operation Mode	The an Asynchronous

Each Axon contains two types of cells with smooth surface and great length, whereas a dendrite has irregular surface and more branches [32]. The structure of a biological neuron and the reaction on an axon is represented in Figures 2.5 and 2.6.

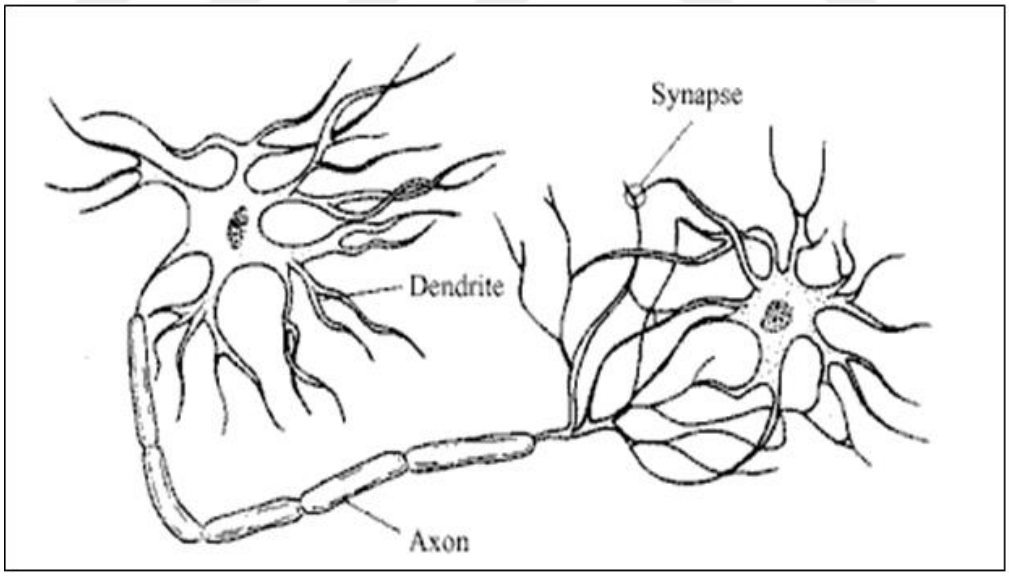


Figure 2.5 The Structure of a Biological Neuron

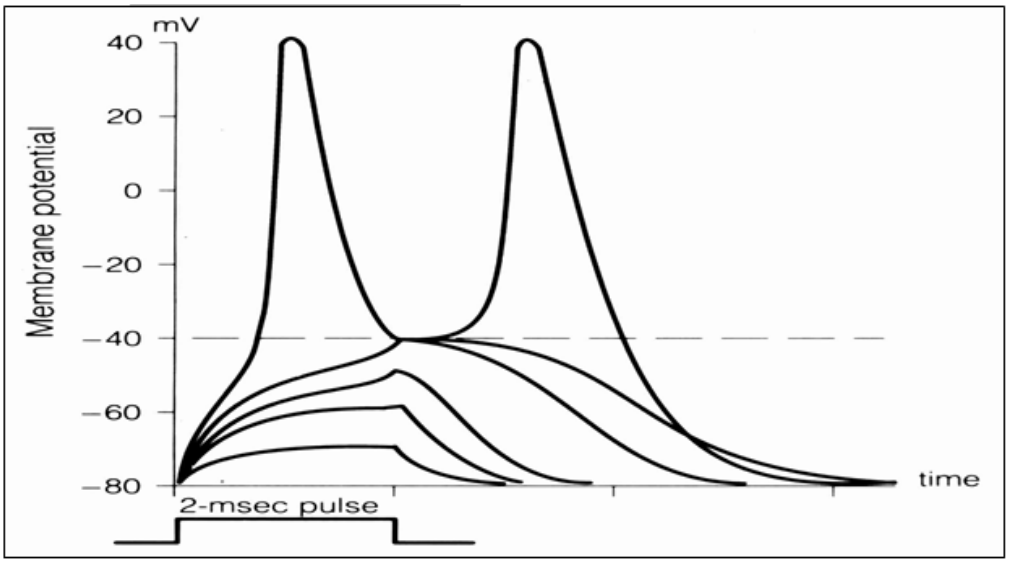


Figure 2.6 The Response of the Axon [32]

2.6.2 Neuron Model

The essential unit of neural systems is the artificial neuron. The artificial neurons are defently much simpler than the biological ones. The artificial neurons were performed to simulate the four basic functions of any natural neurons [29].

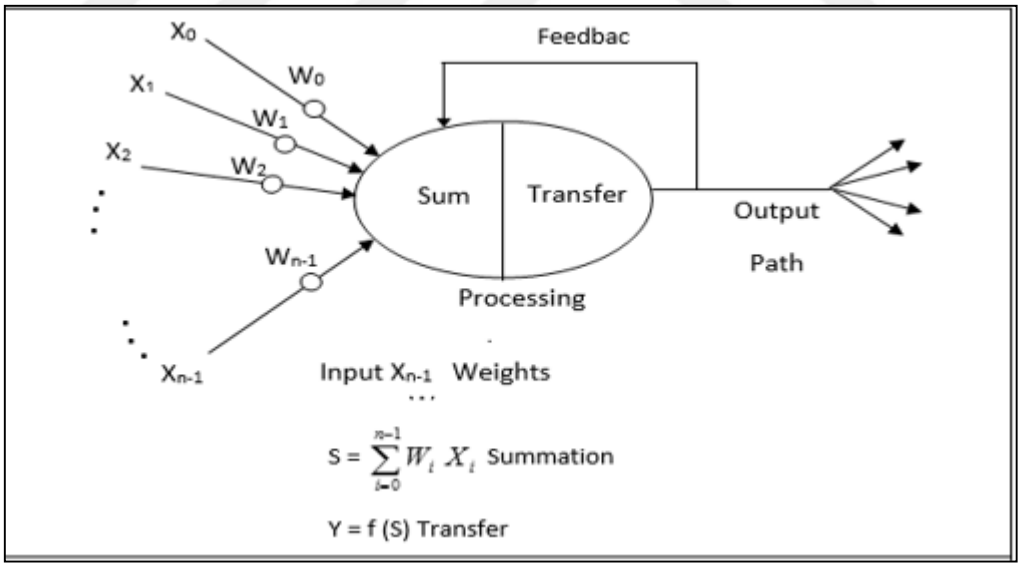


Figure 2.7 Nonlinear Model of a Neuron [27]

Figure 2.7 shows the block diagram of an artificial neuron model. This model illustrates the basic design of an artificial neural network.

The artificial neural model consists of the following basic elements:

- A set of connecting links or synapses whichare characterized by a weight or strength of their own.

A signal (xj) at the input of a synapse (j) connected to a neuron (k) is multiplied by the synaptic weight (wij).

The subscripts of the synaptic weight (wij) should be written in a particular manner. The first subscript refers to the neuron in question,while the second subscript refers to the input end of the synapse to which the weight is referring.

The synaptic weight (wij) of an artificial neuron may lie in a wild range that includesnegative as well as positive values.

- An adder for summing the input signals, weighted by the respective synapses of the neuron; the processesshow a linear combiner.
- An activation function for limiting the amplitude of the output of each neuron. Ideally, the normalized amplitude range of the output of a neuron is written as the closed unit interval [0, 1] or alternatively [-1, 1] [28].

The basic types of activation function are: [27, 32& 33].

a. Threshold function; For this type of activation function, defined in Figure2.8 (a), having:

$$\emptyset(\text{v}) = \begin{cases} 1 & \text{ifv} > 0 \\ 0 & \text{ifv} < 0 \end{cases} \quad (2.7)$$

In engineering literature, this form of threshold function is commonly referred to as aliteral side function.

b. Piecewise-Linear Function. For the piecewise-linear function defined in Figure.2.8 (b) which have:

$$\emptyset(\boldsymbol{v}) = \begin{cases} 1 & if v \geq +0.5 \\ v & if\ 0.5 > v > -0.5 \\ 0 & if v \leq -0.5 \end{cases} \quad (2.8)$$

Where the amplification factors inside the linear region of operation is assumed to be unity.

c. Sigmoid Function. The sigmoid function as shown in Figure2.8 (c) showing an s-shaped graph, represents far most the common form of activation function used in the construction of artificial neural networks. An example of the sigmoid function is the logistic function [27] defined by:

$$\emptyset(v) = \frac{1}{1+exp(-av)} \quad (2.9)$$

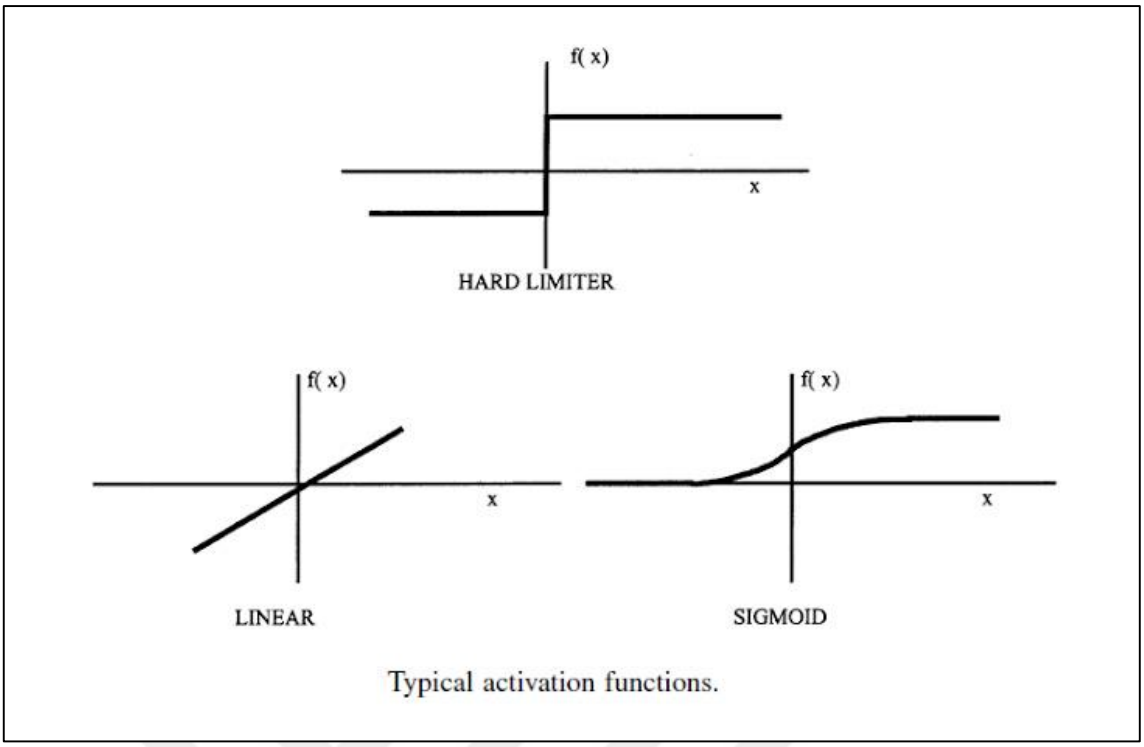


Figure 2. 8 (a) Threshold Function, (b) Piecewise Linear Function, (c) SigmoidFunction for Diverse Slope Parameters[32].

2.6.3Network Model

Network model: The directions of signals flow distinguish between networks, there are two types of networks; feed forward and feedback [31].

2.6.3.1Feed Forward Networks

Feed forward networks: An artificial neuron network where the information moves only forward from the input nodes to the output nodes, in a form of layers [32]. It is divided into three types:

- Single layer perception:contains one input layer and another output layer.
- Multilayer perception:containsa set of inputs connected by one or more hidden layers and an output layer.
- Radial basic function net:it uses radial basic function as activation function.

Figure 2.9 showed the feed forward neural network with the two hidden layers.

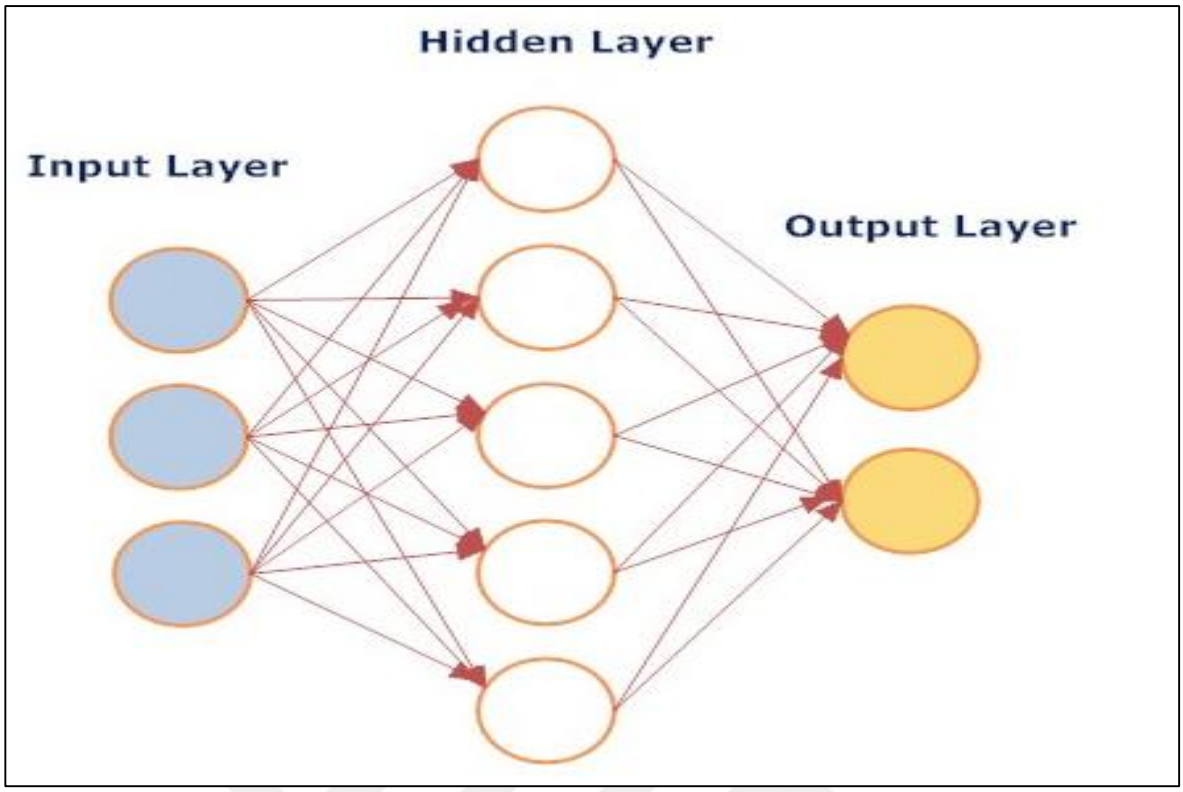


Figure 2.9 A Feed Forward Neural Network Type [34].

2.6.3.2 Feedback Networks

Neural model with feedback network capability has been adopted and employed by programmers as it possesses many advantages. These networks can be run without supervision learning, can be self-organized, capable of retrieving and restoring memory patterns,and provide computing solution to a variety of optimization problems that involve interpreting the state of the network after it is stabilized. Hence, it is necessary to state criteria for each design [27]. This network model was not under focus in the present study.

2.7Naïve Bayes

Naive Bayes is anindependent modelwhich is based on estimation (Nilsson, 1965). The methodwas first stated by Thomas Bayes in the 18th century, who developed its foundational mathematical principles. The method is well known now as Bayesian methods. This method is applied for describing the probability of events, and their revision in light of additional information[35]. Also for text classification, such asspam filtering, author identification, or topic categorization [36].

Naive Bayes is a conditional probability model that can specify a problem instance to be classified. It is represented by a vector that categorizessome (n) features (independent variables).It assigns to this instance probabilities for each of (K) possible outcome or class [37].

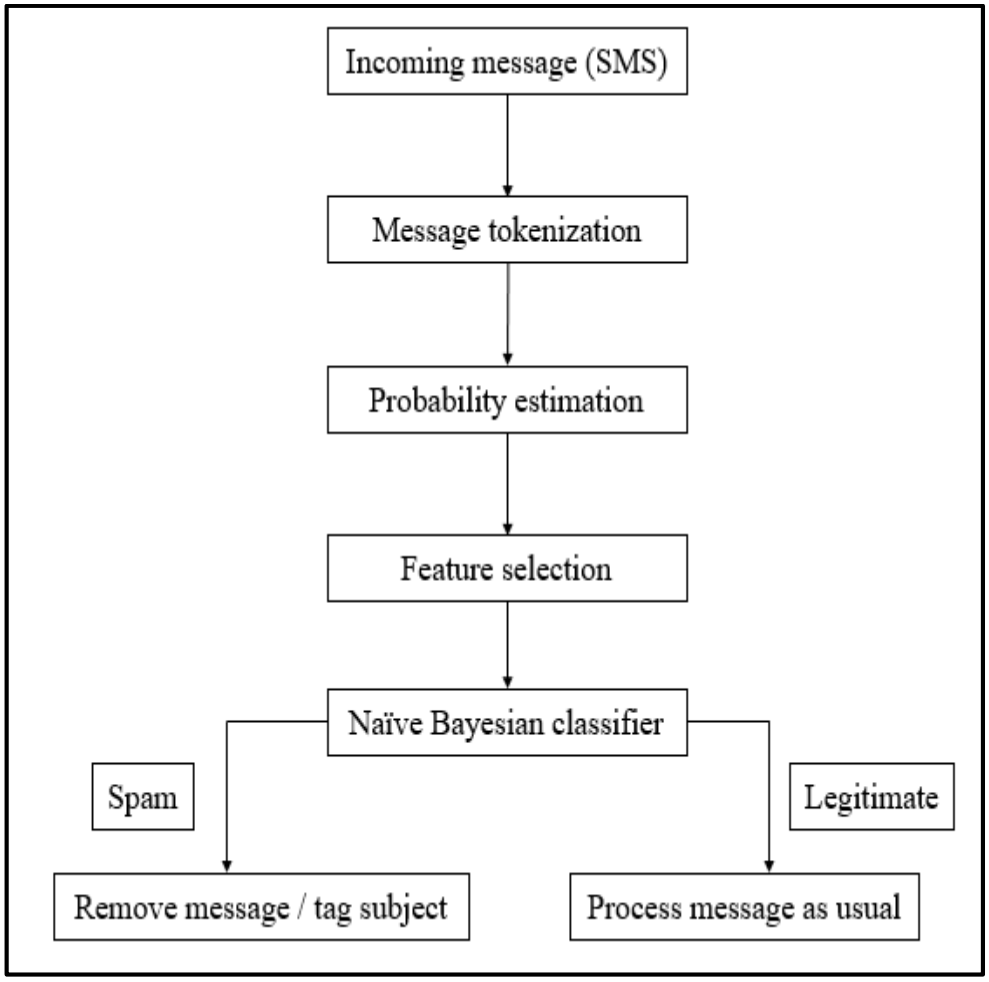


Figure 2.11Naive Bayesian Spam Filtering Model

A simple formula can be performed in many applications due to its enormous practical importance. It is often easier to calculate the probabilities, P(X | C), P(C), P(X) when the probability P(C|X) isrequired. This theorem is central to Bayesian statistics, which calculates the probability of a newevent on the basis of earlier probability estimates derived from empirical data.

$$Posterior = \frac{Likelihood \times Prior}{Evidence} \quad (2.10)$$

2.7.1Naive Bayesian Classifier

The conditional probability can be decomposed as:

$$P\left(\frac{C}{X}\right) = P(C) * \frac{P\left(\frac{X}{C}\right)}{P(X)} (2.11)$$

Where (X) is our feature vector, and(C) is the probable class.

Since the denominator will be constant, we are just keen on the numerator.

$$P\ (c/x) = \left(x_1, x_2 \dots \dots \dots x_{n\backslash C_j}\right) * P(x_2) \quad (2.12)$$

In this case,(x_n) are features of the feature vector where (n)is the feature number from one up to the feature vector's dimension size and (Cj)is the class type (for example spam or legitimate).

By calculating the posterior probability for (Cj) given (X) and knowing that each feature of (X)is assumed to be conditionally independent for Naive Bayes, theformula can be rewritten as: $P\left(\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3} \dots \mathbf{x_{n\backslash C_j}}\right)$into:

$$H(X) = \prod_{k=1} nP\left(\mathbf{x_{k\backslash C_j}}\right) (2.13)$$[37]

2.8 K-Nearest Neighbors (K-NN)

In 1951, Fix and Hodges [38], presented a non-parametric technique called the category of training.A large group of examples that affect the speed of network learn, using the k-nearest neighbor base method. which is differ from the K-NN calculation technique that classify objects in light of nearest preparing cases in the element space, it realizing where the capacity is just approximated locally and all calculation is conceded until classification [39].

Basically, K-NN classifier is dictated by both the decision of K and the separation of metric connected. When K is small, the nearby gauge is to be very poor attributable,and the loud, vague or mislabeled focuses. If Kof inquiry is expanded, this will make the gauge over smoothing and the characterization execution corrupts with the presentation of the anomalies from different classes.

To enhance the grouping execution of K-NN, the most straightforward greater part voting of consolidating the class marks for K-NN can be an issue if the closest neighbors shift broadly over their separations and the nearer ones all the more dependably demonstrate the class of the question protest.

A key issue to obtain a prediction we choose an appropriate neighborhood estimate K to a great extent influences the order execution of K-NN for over four decades this approach has been available proved to be the top-performer on Reuters corpus.
When the Query instance ($x_{q)}$ to aand has given query instance (x) to be classified,

Let x_1,x_2….x_k denote the (k) instances from training examples that are nearest to $_(x_q)$.

These (n) attributes are considered to be the independent variables.

Return the class that represents the maximum of the k instances.

If k= 5, then in this case query instance (x_q) will be classified as (-ve) since three of its nearest neighbors are classified as (-ve).

Distance usually relates to all the attributes and assumes all of them have the same effects on distance. The similarity metrics do not consider the relation of attributes which result in inaccurate distance and then can produce an impact on the classification precision. The presence of irrelevant attributes could produce Wrong classification and is usually termed as the curse of dimensionality.

For example: each instance is described by 20 attributes that only 2 of them are relevant for determining the classification of the target function. Hence, instances that have identical values for the two relevant attributes may be distant from one another in the 20-dimensional instance space.There are four approaches as below:

The first Approach deals with:

- Associate weights with the attributes.
- Assign weights according to the relevance of attributes.

a. Assign random weights.

b. Calculate the classification error.

c. Adjust the weights according to the error.

d. Repeat till an acceptable level of accuracy is reached.

The second Approach (Attribute Weighted)

a. Gradient Descent

b. Assign random weights to all the attributes

c. Train the weights using Cross Validation.

The third Approach (Instance Weighted)

a. Gradient Descent

b. Assign random weights to all the training instances

c. Train the weights using Cross Validation

The fourth Approach

Backward Elimination by the following steps:

a. For all attributes do Delete the attribute.

b. For each training example (xi) in the training data set, finding the K-nearest neighbors in the training data set based on the Euclidean distance.

c. Predict the class value by finding the maximum class represented in the K-nearest neighbors then calculate the Accuracy as:

Accuracy=(number of correctly classified examples/number of training examples) * 100

- If the accuracy has decreased,
- Restore the deleted attribute.
- Read the training data from a file.
- Set K to some value.
- Normalize the attribute values in the range 0 to 1.
- Value = Value / (1+Value);
- Apply Backward Elimination for each testing example in the testing data set.
- Find the K-nearest neighbors in the training data set based on the Euclidean distance.
- Predict the class value by finding the maximum class represented in the K-nearest neighbors.

Calculate the Accuracy as:

Accuracy=(number of correctly classified examples/number of testing examples)*100

Starting with the full set of features and greedily removing the one that most improves the performance, or degrades the performance slightly.

Equation to find the Accuracy:

Accuracy = (number of correctly classified examples/number of examples) * 100

Equation to find Standard Euclidean Distance:

$$D(Xi, Xj) = \sqrt{\text{for all attributes a} \sum (Xi, a - Xj, a)^2} \quad (2.14)$$

2.8.1 Advantages and Disadvantages of K-Nearest (K-NN)

Studying K-Nearest Neighbors indicate many advantages as well as some disadvantages.

The most important advantages of K-NN are: simplicity, effectiveness, intuitiveness and competitive classification execution in numerous areas. However, K-NN has disadvantages as it is robust to noisy training data when training data is big. Besides, it is limited with poor run time performance and sensitivity to irrelevant or redundant features because all components add to the similitude and consequently to the classification[39].

2.8.2 Applications of K-Nearest Neighbors

K-Nearest Neighbors is widely applied as a data mining technique in regression and classification like text mining (categorization different numbers rather than a fixed number across all classes. K value is the only parameter that is needed to be chosen because of less sensibility. Data mining techniques in agriculture related fields can be of particular importance in evaluation of forest inventories and for estimating forest variables when satellite imagery is used. Other uses of K-NN as in medicine to calculate and determine the amount of glucose in the blood of a diabetic person or prostate cancer risk factor using demographic variables. Besides, data mining can be used as a process of finding useful patterns and correlations in financial modeling like linear and non-linear models multi-layer neural networks [40].

CHAPTER 3

MATERIAL AND METHOD

3.1 Overview

Efficient and accurate recognition of SMS messages as either spam or ham from a set of messages is one of the foremost challenges for cell phone users. The current study was proposed to extract and analyze the messages that have been received from SMS dataset. Three different classifications algorithems have been used individually (Naive Bayes, Artificial Neural Network and K-Nearest Neighbors) dealing with hug number of dataset. Different feature selections were adopted by each one of these classifiers. The results extracted by these classifiers were compared between each other as well as with other studies. The sequence of this research was represented in Figure 3.1.

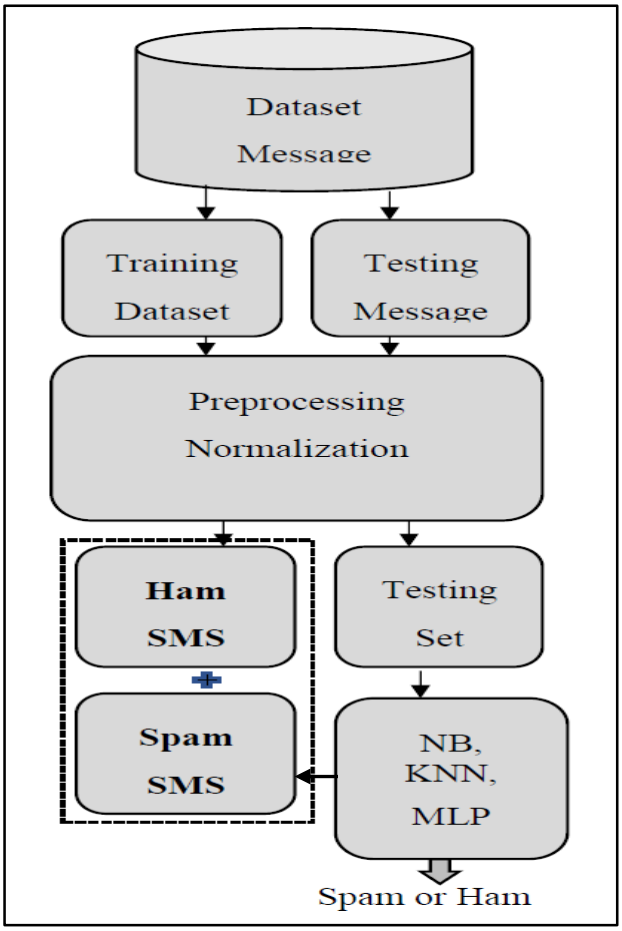


Figure 3.1 Block Diagram for SMSFiltering System

3.2 SMS Dataset

The data set were collected from the net source provided for researchers. The data is available for cell phone researches. About 5,574 messages were adopted for this research. These messages were selected to be as follow: English, non-encoded messages, and classified as legitimate (ham) or spamhttp://archive.ics.uci.edu[41] as shown in Table 3.1. This dataset Contains noise that needs processing prior to direct classification and will be described in detail in next stepswas used by other studies [10] and their results were used for comparison with the results of the present study.

Table 3.1 The Dataset Details

Application	**Number of Spam SMS**	**Number of Ham SMS**	**Total SMS**

General	747	4827	5574

The data (The Dataset Access file) were performed in a two-column table:

- The first column represents the type of messages (spam or ham).
- The second column represents the message.
- This table it is contain 5574 record.

Table 3.2 The Original Dataset Access file

1	ham	Go until jurong point, crazy. Available only in bugis n great world la e buffet... Cine there got amore wat...
2	ham	Ok lar... Joking wif u oni...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives around here though
:	:	…… … …… … ……..
5574	spam	FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun? Tb ok! XxXstdchgs to send, 1.50£ to rcv

3.3 Preprocessing of Dataset

The dataset was loaded and read with put it in table named “dt” using C sharp program can see in figure 3.2.

```
publicstaticDataTable DBConnect()
{
OleDbConnection con = newOleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=DataSet.accdb");
con.Open();
string sql = null;
            sql = "SELECT * from Dataset" ;
DataTable dt = newDataTable();
DataSet ds = newDataSet();
System.Data.OleDb.OleDbDataAdapter da = new System.Data.OleDb.OleDbDataAdapter();
ds.Tables.Add(dt);
            da = newSystem.Data.OleDb.OleDbDataAdapter(sql, con);
da.Fill(dt);
con.Close();
return dt;
}
```

Figure 3.2 loadina and reading dataset by C# Program

The preprocessing functions were performed according to the following steps:

- All the (uppercase letters) were changed to (lowercase letters) within the message words as shown in the message below example shows the shape of the message after character conversion:

Hi. Hope you had a good day. Have a better night

hi. hope you had a good day. have a better night

Figure 3.3 Message Example

Code used C # for character conversion within Datasetshown in figure below

```
 privatestaticint CountWord(string T, string W)
{
T = T.ToLower();
int x = 0;
T = T.ToLower();

if (T.Contains(W))
{
```

Figure 3.4 cod program use changeduppercase letters to lowercase letters by C#

- Tokenization: which is a process of cutting text into individual words to be recognized by the program, this process is done by separating the message into a specific set of words which depends on the separation method that is determined by the token program.
- Identification of Stop words and their elimination.

Stop words are all the words that can be seen so frequently in any message or on web pages. Identification of these Stop words in order that search engines would ignore them when indexing the words in any message. Stop words can be words as: a, and, is, on, of,

or, the, was, with, etc. [42], in this study we use database contain all the words that can pass during in message and they are saved in string of memory this memory we use to identification of stop words.This is shown within the code in Appendix-A

- Encoding of the dataset:

During the process of transforming the dataset from the net to the PC, the data may become unrecognized by the program due to encoding, hence, some coding correction should be performed to make it usable by the program.

- Term Frequency:

The dataset was transformed to numerical dataset.

We can see all program code to the steps from 2-5 in appindex-A

3.4 Feature Extraction

Two types of feature extraction for about 27 features were selected for this study that used for training and testing data, throw using bag of words and structure.

Were the bag of words: is a model used [for text and document classifications using the most frequency used words of a text to be as bag of words for the same text [43]. In the current study, about 22 words (features) have been used as bag of words. In this stage, the results were culected as shown in appendex B figure B.2.

The words are
{Money, price, prize, win, winner, won, call, congratulation, http, www, sms, code, urgent, cash, xxx, customer, service, offer, guarantee, subscribe, cash, txt}[44][45]
The structure features: the set of features used included five structure features as below:

- Words: this feature represents the number of the words. For example, the words count in each message equals to the number of spaces between them.
- Length: the number of letters, numbers, symbols, digits, etc. that are included within each message.
- Letters: the total number of characters in each message.
- Digit: the percentage of digits to each message.
- Symbols: the percentage of the symbols in relation to the characters in the message.

The figure below represents the length of the message relative to the number of characters. The number of letters the number of digits, letters.

```
publicstaticvoid Extraction(){
            FeaturesV = newdouble[5574, FName.Length];
int f = 0;
foreach(string Tx in TEXTS)
{
                FeaturesV[f, 0] = Tx.Split(' ').Length ;
                FeaturesV[f, 1] = Tx.Length;   //  length of message
                FeaturesV[f, 2] = CountLetters(Tx);
                FeaturesV[f, 3] = CountDigits(Tx);
                FeaturesV[f, 4] = CountSymbols(Tx);
//FeaturesV[f, 0] = CountWord(Tx, SpamWords[0]);
for (int i = 0; i <SpamWords.Length; i++)
                    FeaturesV[f, i + 5] = CountWord(Tx, SpamWords[i]);
f++; }}
privatestaticdouble CountDigits(string T)//  The number of digits relative to the length of the text
{
return (float)T.Count(char.IsDigit) / (float)T.Length;
}
privatestaticdouble CountLetters(string T)
{
return (float)T.Count(char.IsLetter) / (float)T.Length;
}
privatestaticdouble CountSymbols(string T)
{
returnRegex.Matches(T, "[~!@#$%^&*()_+{}:\"<>?.',-]").Count / (float)T.Length;
}
```

Figure 3.5Code Program Usedto Extract the Featuresby C#

Example: if we have a message such as (“22 where are you?”) inthis amessage the number of letter is (11), length of message is (14) without space Then the percentage of letters to message is (11/14) =78.5%

In order to train SMS data, the data set should be normalized. Normalization means implies that all values from the dataset should take given values ranging from 0 to 1.

For this purpose, the following formula was used:

$$X\,n = \frac{X - X\,min}{X\,max - X\,min} \quad (3.1)$$

If (X) is the value that should be normalized, (Xn) is the normalized value.

(Xmin) is the minimum value of (Data), (Xmax) is the maximum value of (Data)

Hence, if a feature has a (6) value where4, 12, 13, 24, 9 then:

$$X\,n = \frac{6 - 4}{24 - 4} = 0.1$$

3.6 Feature Selection

There are two types for feature selection

1. Information Gain (IG)

The IG is a feature selection method that is used to identify which attribute in a given set of training feature vectors is most useful between the used classes. In order to calculate IG, entropy should be first extracted.

Both IG and entropy can be calculated as in the following example:

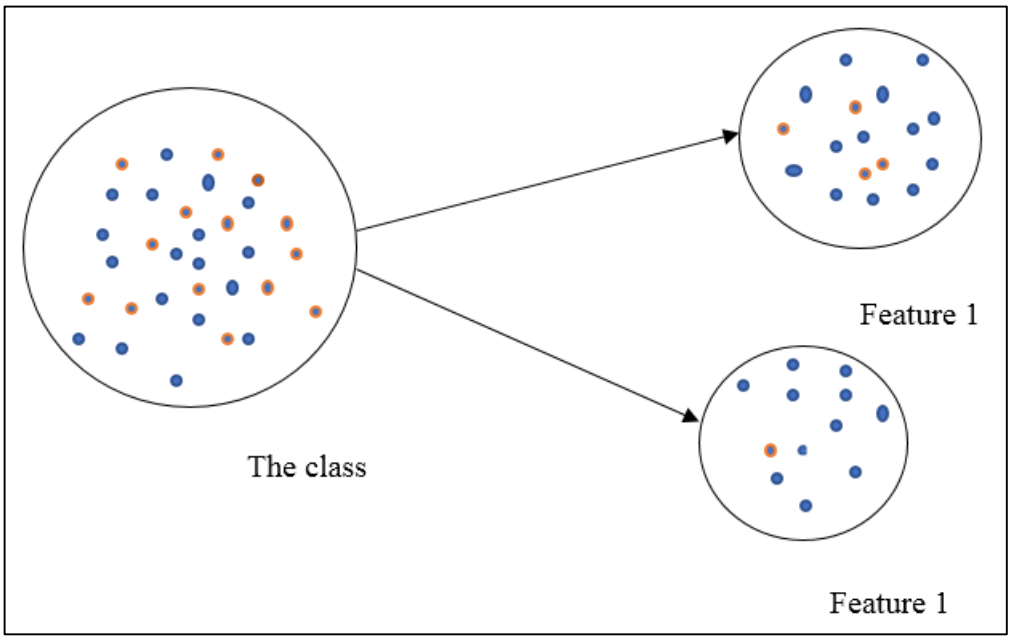


Figure 3.6Example for Information Gain Extraction

To calculate the probability (pi) of any class (i), follow the following steps:

For example, if we have one class with two features, IG can be calculated as:

IG=Entropy (class) - [weighted average] Entropy (features) (3.2)

Where Entropy (E1) for feature 1 =-13/17 log 13/17 - 4/17 log 4/17

E1=0.787

Entropy (E2) for feature 2 = -1/13 log 1/13 - 12/13 log 12/13

E2=0.391

Class entropy= - 14/30 log 14/30 -16/30 log 16/30

=0.996

(Weighted) Average Entropy of features=17/30* 0.787 +13/30**0.391

=0.615

IG=Entropy (class)- [weighted average] Entropy (features).

Information Gain= 0.996-0.615=0.38 [46].

2. Information Gain Ratio

The information gain ratio (IGR) is a ratio ofinformation gain (IG) to the intrinsic information or also named Intrinsic Value (IV).

IGR=IG/IV (3.3)

It is used to reduce a bias towards features with multi-values by taking into account all the number and size of branches when choosing feature [47].

You can find the gain through the following equation:

$$\text{Gain} = \text{IG} - \text{E} \quad (3.4)$$

3.6.1 Methods for Selecting the Features

Using the standard method of classification (non-feature selection), we calculated the Accuracy, Recall, F-measure, Precision and time values for the 26features that have been set for the study. These values are going to be compared with the classification values that are expected to results using the suggested methods of the current study.

Feature selection, using IG and IGR, was performed to eliminate the five features with the least 5 values that were recorded for the accuracy, recall, F-measure, precision and

time. This step was repeated until ending with only four features. The results for each step were recorded and used for comparison with each other.

3.7 Classification

The classification algorithm is the last step in this work which is used to evaluate message, three types of methods were used in this study; Naïve Bayes, Artificial Neural Network and K-NN.

The dataset were divided into: 80% training dataset and 20% testing dataset. The training dataset were used for classification in all the three-used classifier. The code used Calls set of labraryin C# program seen in figure 3.7

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using weka.classifiers;
using weka.core;
```

figure 3.7Weka Library Including in C#

After that use this libraray in C# pragram open sourse to select the classifications operating to the three algorithms and spesefect the paraamerter to each algorithm .We can show that in appendix-A2

3.7.1Naïve Bayes

For SMS classification using Naive Bayes classifiers, the general formula was used to calculate the highest probability of SMS classification in the classifier models between texting and training dataset.

For dataset (d), and a class(c) finding the probability as below:

$$p\frac{c}{d} = \frac{p(d/c)p(c)}{p(d)} \tag{3.5}$$

To calculate the probability of the messages to be ham for all the 26 features, the following equation was used:

P(h) =p(x1/h)* p(x2/h)* p(x3/h)* …..p(x26/h) (3.6)

Where P(h) refer to probabilityof ham, and (x) is the feature.

To calculate the probability of the messages to be spam for all the 26 features, the following equation was used:

$$P(s) = p(x1/s)* p(x2/s)* p(x3/s)* \ldots\ldots p(x26/s) \quad (3.7)$$

Where P(s) refers to probability of spam, (x) is the feature.
Finally, the higher value for the probabilities is to be conceded, if Ps > Ph then the message is spam.

3.7.2Artificial Neural Network (ANN)

In the current study,Multi-Layer Perceptron(MLP)network was adopted.

The feed-forward Multi-Layer Perceptron networks can be learned using gradient valuescalculated by an algorithm namedError Backpropagation.In multilayer network learning, the usual objective or error function can be minimized. These errors have the form of squared errors.

Figure 3.8 represents MLP structure used in this study. The structure consists of input layer (26 features),seven hidden layers and an output layer with size equal to two, which is the number of classes (spam or ham).

The parameters that have been used with their respective numbers were:

Learning Rate (0.1)
Momentum (0.2)
Training Time (1000)
Hidden Layers(7), and the neurons numbers are randomly chosen by program

3.7.2.1 Finding the Best Number of The Hidden Layers

Table 3.3 Selecting the Number of the hidden Layers

No.of layers	**1 Layer**	**2 Layer**	**3 Layer**	**4 layer**	**5 Layer**	**6 layer**	**7 Layer**	**8 Layer**
Accuracy	97.66	97.84	97.48	97.84	97.93	97.30	98.02	98.02
Time	12.65	12.61	18.55	23.49	33.98	39.55	31.70	42.70

In Multi-Layer Perceptron, the number of hidden layers used may affect the Accuracy and the time. Therefore, and to enhance our results extracted for these two parameters, the best number of hidden layers must be selected. Several trials were adopted to reach the best number of hidden layers that resulted in higher accuracy and lower time. Depending on the results shown in Table 3.3, the number of hidden layers was selected to be 7 layers.

3.7.2.2 Determine the Most Appropriate Learning Rate

Table 3.4 Effect of Learning Rate on Accuracy

Learning rate	0.1	0.2	**0.3**	0.4	0.5	0.6
Accuracy	97.75	97.66	98.02	97.84	97.93	97.75

In Multi-Layer Perceptron, the training rate has high impact on Accuracy in the classification algorithm. In the current study, the best training rate value that is needed to achieve the highest Accuracy was selected to be 0.3. This value was selected after performing several trial implications to achieve the highest Accuracy as shown in Table 3.4.

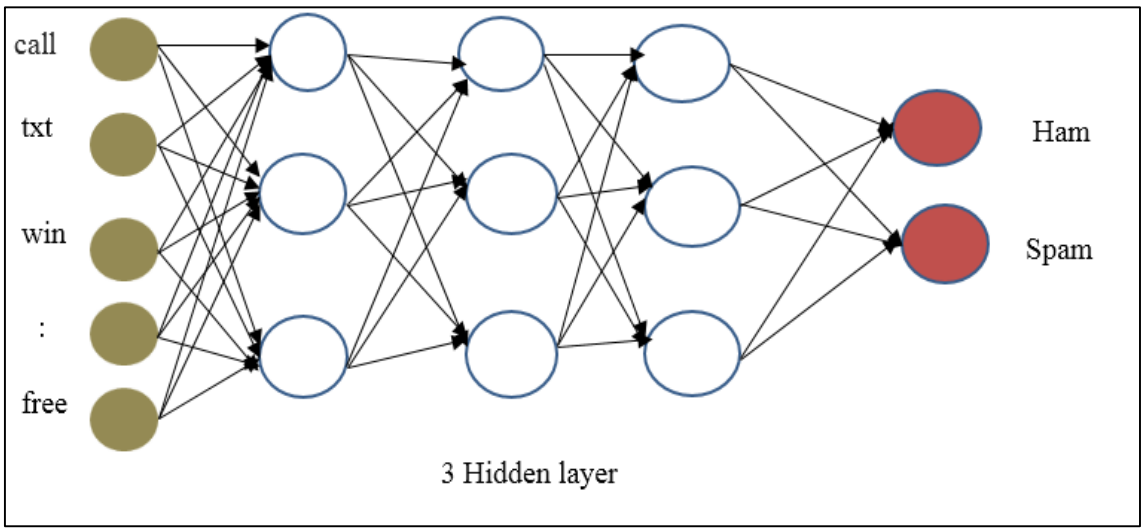


Figure 3.8 Example of Multi-Layer Perceptron Classifier

3.7.3 K-Nearest Neighbors

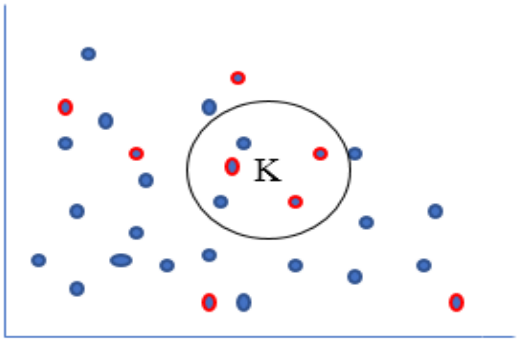


Figure 3.9Example of K-NN Classification

In K-NN classification or also named the lazy classifier, the messagesare classified by a majority vote of their neighbors, with the messages being assigned to the class most common among its k nearest neighbors (where K is typically a positive small and odd integer). The output of the classifier is representing a class membership.

The classifier will test the distance between testing message and the training message.

This distance is calculated by the distance between two points. It can be calculated in this equation:

$$D(\mathrm{Xi}, \mathrm{X\,j}) = \sqrt{\sum (X\mathrm{i}, \mathrm{a} - Xj, a)^2} \qquad (3.8)$$

Where (K) value is hypothetically equalsto odd numbers such as; 3, 5,7, …….

The testing messages would be input into the classifier. Depending onthe nearest neighbor messages; each test message will be selected to its own classification.

3.7.1 Finding the best Value of (K)

Several tiers were performed in order to reach the best value of (K) to be used in this study. According to the Table 3.5, the best values for K were: 7, 15, 17, and 19, in terms

of precision selection with the lowest time. According to Imandoustan and Bolandraftar (2013), the value of (K) =7 was used in order to avoid the possibility of overlap and similarity between classes as well as decreasing noise [38].

Table 3.5 Effects of K value on Accuracy and Time

Number	**K=1**	**K=3**	**K=5**	**K=7**	**K=9**
Accuracy	95.571	95.410	95.571	95.911	95.225
Time	1.202	1.127	1.202	1.447	1.491
Number	**K=11**	**K=13**	**K=15**	**K=17**	**K=19**
Accuracy	95.242	94.894	96.283	96.622	96.453
Time	1.558	0.924	1.580	1.687	1.807

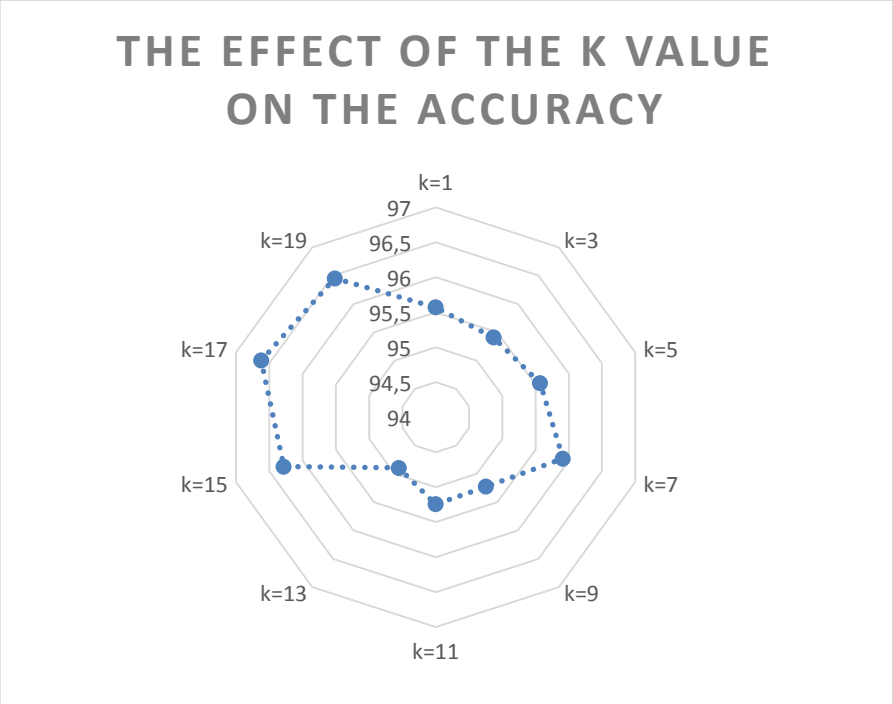


Figure 3.10The Effect of changing K value on Accuracy

The best approximation of the closest neighbor classification algorithm is at 17, 19, 15 and 7 respectively (Figure 3.10).

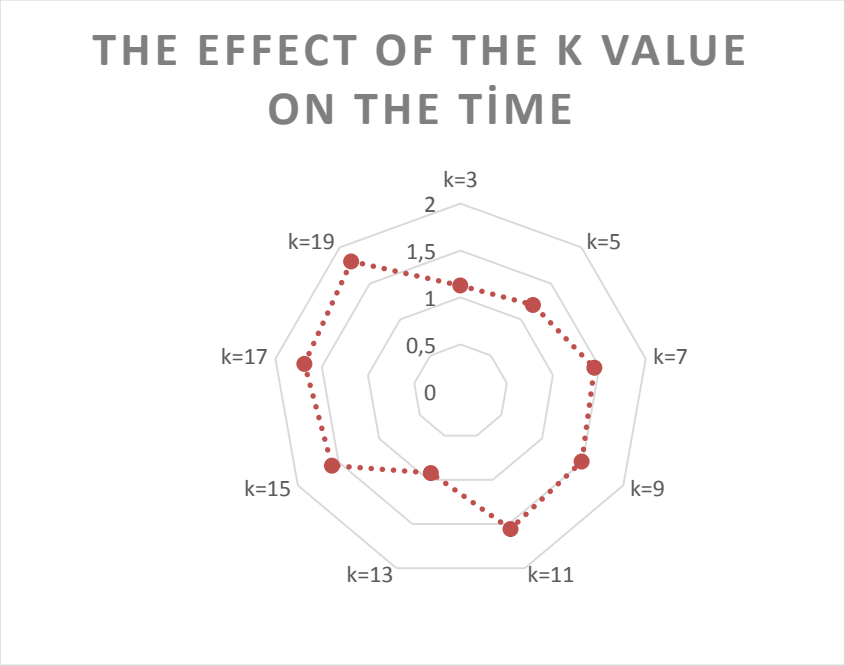


Figure 3.11The Effect of changing K value on Time

The minimum time required to classify the k-nearest neighbor classification algorithm was calculated at K = 13 and then 3 (Figure 3.11).

In this study (K) value was chosen to be 7.

3.8 Implementation of the study

This study was implemented using the following personal computer specifications;
CPU i7, Q840, Memory ram 4.00 GB, the Access program version 2016.
For faster access to the data set, the original data were converted from a Comma Separated Values (CSV) to Access file database. The programming of the study was adopted using C sharp version 6.0 and the environment of visual studio community (2017) [48].
The pre-processing of the dataset, the two feature selection methods andthree classifiers algorithmswere performedby C sharp program depending on WEKA laybrary that we incloding in it by open sorce program[49]. The calculated parameters were represented as in Figure 3.12.

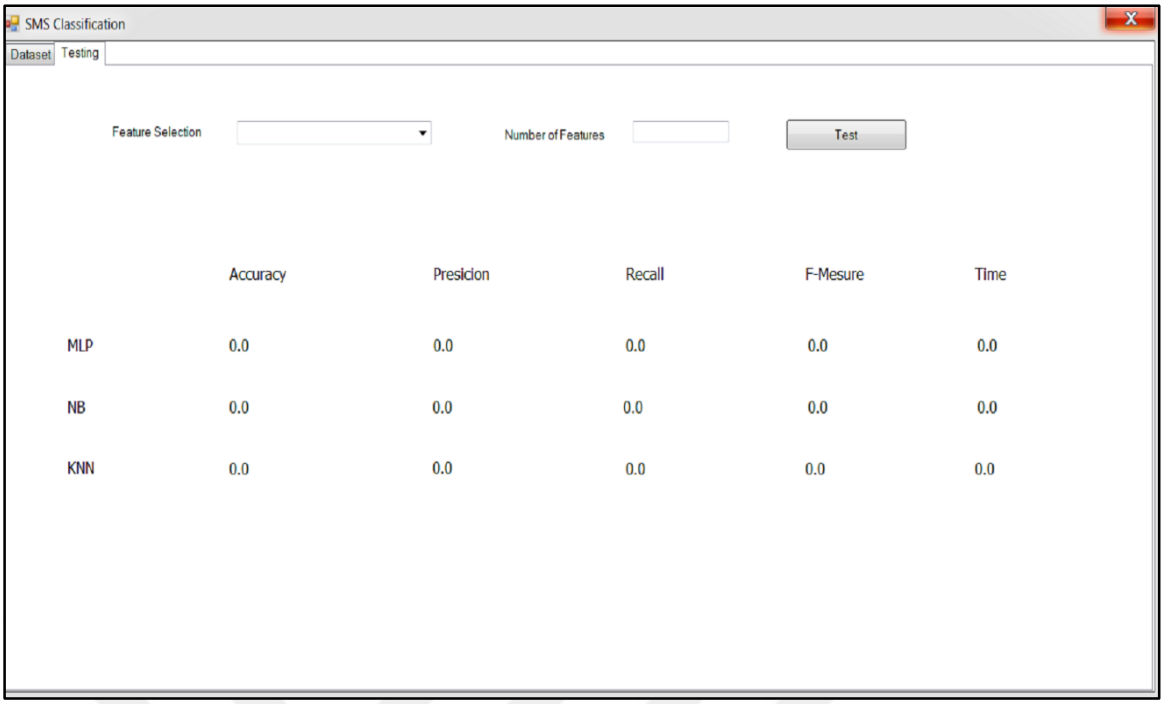


Figure 3.12Program Execution Interface

The basic class of the WEKA is expressed in Java (jar file) to be used with C Sharp.

The jar file was then converted to Dynamic Link Libraries (DLL) for easy of handling by dot net languages using IKVM (IKVM is a software that converts jar to dot net).

WEKA has been used by programmers for several reasons:

1. Free open source environment used by most of the machine learning that can be used for different projects.

2. It is possible to access the main class and programmed in Java language to be used by dot net.

3. Contains rich information documents.

4. The results are provided by reliable scientific researches.

3.9 Evaluation of the Parameters

Accuracy, Recall, F-measure, Precision and time were calculated by their correlated equations. These parameters were calculated with C sharp as shown in Figure 3.13.

```
Public static double[,] Confusion Matrix(inttp, inttn, intfp,
intfn)
        {
Double [,] EV = new double [2, 2];
            EV [0, 0] = 100 * ((double) (tp + fp) / (tp + tn +
fp + fn)); //accuracy
            EV [0, 1] = ((double)(tp ) / (tp + fp));
// Precision
            EV [1, 0] = ((double)(tp) / (tp + fn));
// Recall
            EV [1, 1] = (2 * EV [0, 1] * EV[1, 0]) / (EV[0, 1] +
EV[1,0]);
// F-M
return EV;
        }
```

Figure 3.13Code of Calculation of Classification Criteria

To evaluate the classifications of the current study, the confusion matrix was used.Confusion Matrix is a table describes the performance of the classification model. Table 3.6 shows the parameters used with a representative example for a total of 100 datasets (60 for the Condition Positive and 40 for the Condition Negative) for calculation of the matrices used in the study.

Table 3.6Representative Example with theParameters used for Classification Evaluation

Total Prediction	Prediction Positive	Prediction Negative	Total No. of Dataset
Condition Positive	True Positive (TP) If (TP)= 55	False Negative (FN) If (FN) = 5	60/100
Condition Negative	False Positive (FP) If (FP) = 3	True Negative(TN) If (TN) = 37	40/100

In order to calculate the Accuracy which is defined as all incorrect prediction data divided by the total number of dataset, the following equation was adopted:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (3.9)$$

Therefore:

Accuracy $= \frac{55+37}{55+37+3+5} = \frac{92}{100} = 92\%$

Recall, or what is called sensitivity or true positive rate, is the number of positive predictions data divided by the total number of positive. It was calculating as follow:

$$\text{Recall} = \frac{\text{TP}}{\text{P}} \quad (3.10)$$

$$\text{Recall} = \frac{55}{55+37} = 0.597$$

Precision takes all retrieved documents into account, but it can also be evaluated at a given cut-off rank, considering only the topmost results returned by the system. For a text search on a set of documents, precision can be calculated as the number of correct results divided by the number of all returned results.

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{TF}} \quad (3.11)$$

Thus, $\text{Precision} = \frac{55}{55+5} = 0.916$

F-measure or what is also called the balanced F-score is a measure that combines Precision and Recall, it is the harmonic meanof Precision and Recall. It is calculated as:

$$\text{F} - \text{measur} = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} \quad (3.12)$$

Therefore, $\text{F} - \text{measure} = \frac{2}{\frac{1}{0.916} + \frac{1}{0.597}}$

The time values for the three-machine learning algorithms used were extracted from the program depending on the criteria of the used PC.

All these five parameters were calculated by the three machine learning classifiers and then were used for comparison among them and with other studies. The result of this step represed in appendex B (Figure B.3).

CHAPTER 4

RESULTS AND DISCUSSION

This study focused on comparing and discussing the results recorded for only Accuracy and the time using the three-machine learning algorithms; Naive Bayes, Multi-Layer Perceptron, K-Nearest Neighbors. The three classifiers were adopted using two feature selection methods; Information Gain and Information Gain Ratio.

4.1 Results Without Feature Selection Methods

Table 4.1, shows the results for Accuracy, Precision, Recall, F-measure and Time obtained by filtering using the three types of classifiers that were calculated without using any feature selection method.

Table 4.1 The Classification Without Using Feature Selection

Type	Accuracy	Precision	Recall	F-measure	Time (second)
MLP	98.02	0.8763	0.9979	0.9332	31.7
NB	96.05	0.8683	0.9697	0.9162	0.07

K-NN	96.63	0.8744	0.9804	0.9244	0.91

Figure 4.1, shows the Accuracy values recorded by the three classifiers without feature selection method. The Accuracy value recorded by Multi-Layer Perceptron classifier was the highest (98.02) compared to the Accuracy values recorded by NB and K-Nearest Neighbors (96.05 and 96.63, respectively). This could be due to that Multi-Layer Perceptronclassifier can be modified if new training weights are set [51].However, the Accuracy values that were recorded by all the three classifiers are consider highly acceptable with the use of the 26 features that were adopted for this study.

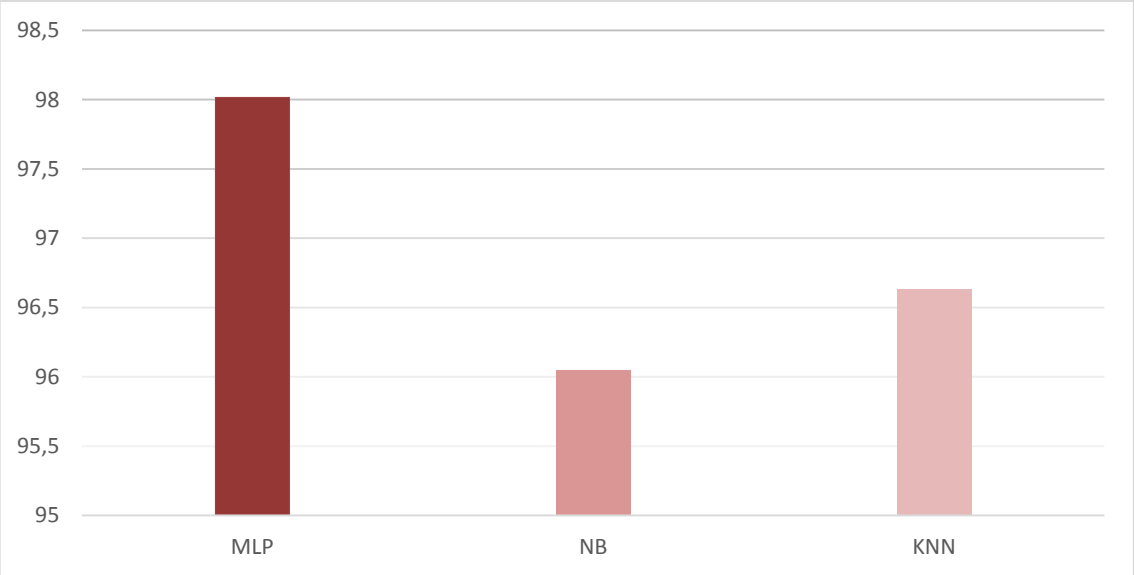


Figure 4.1 Accuracy Values Between the Three Filters without Feature Selection Method.

As shown in Figure 4.2and in respect to Time, NB algorithm completed the filtering process within few seconds (0.07), whileK-NN and Multi-Layer Perceptron recorded (0.91 and 31.) seconds, respectively. According to these resultsMulti-Layer Perceptron classifier recorded the highest accuracy value but on the expense of time, while, NB recorded the shortest time but with the least accuracy value compared to the other two classifiers.

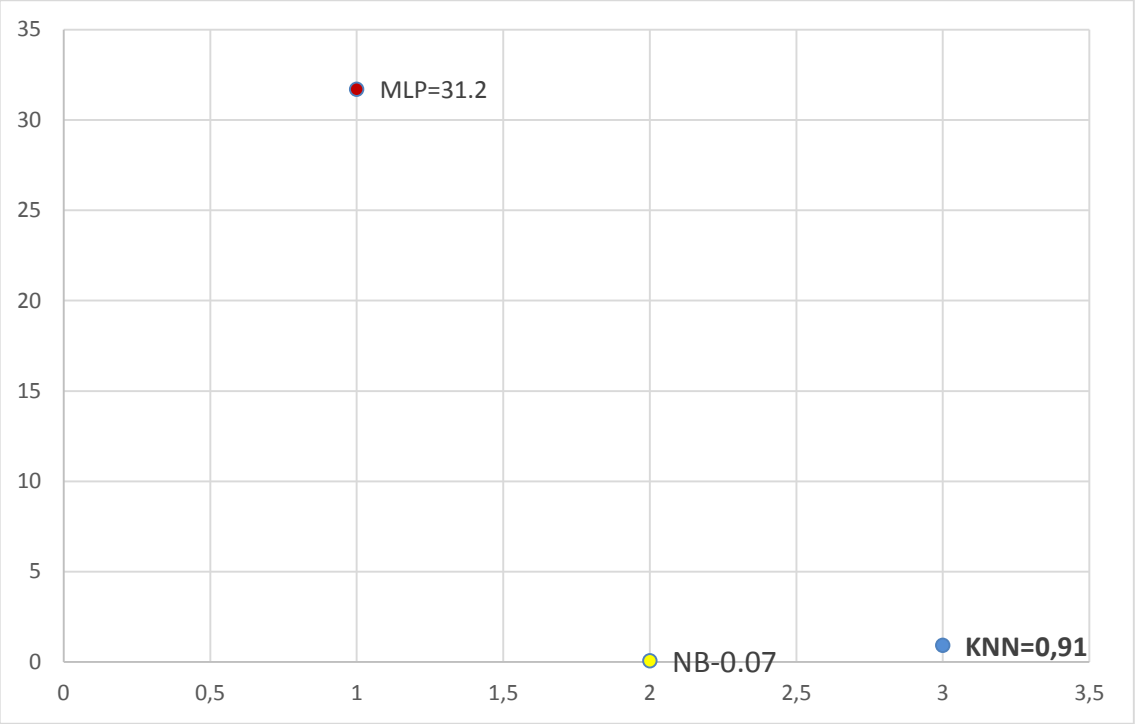


Figure 4.2 Time recorded by theThreeclassifierswithout feature selection method

4.2 Results with Feature Selection Methods

4.2.1 Feature Selection with Information Gain method

The results of the Accuracy recorded bythe three classifiers using Information Gain method isshown in Figure 4.3.The results showed that the feature selection may affect the Accuracy of each classifier in a different way.

For Multi-Layer Perceptron, the Accuracy values were higher than the other two classifiers in all the levels of information gain regardless of the number of feature deleted. The Accuracy ranged from (97.66) for feature selection (18) to (98.86) when using feature selection of (5). While compared with Accuracy recorded without feature selection which was (98.020).

Table 4.2, shows the effects of changing the number of features on the Time needed for spam filter process for each of the three filters used in this study.

Table 4.2 The Time Recorded by Three Classifiers (IG)

Classifier	NO F.S.	5 Feature	9 Feature	12 Feature	14 Feature	18 Feature
MLP	31.2	14.84	8.98	15.6	14.02	12.10
NB	0.07	0.05	0.02	0.06	0.05	0.07
KNN	0.91	0.87	0.80	0.86	0.80	2.46

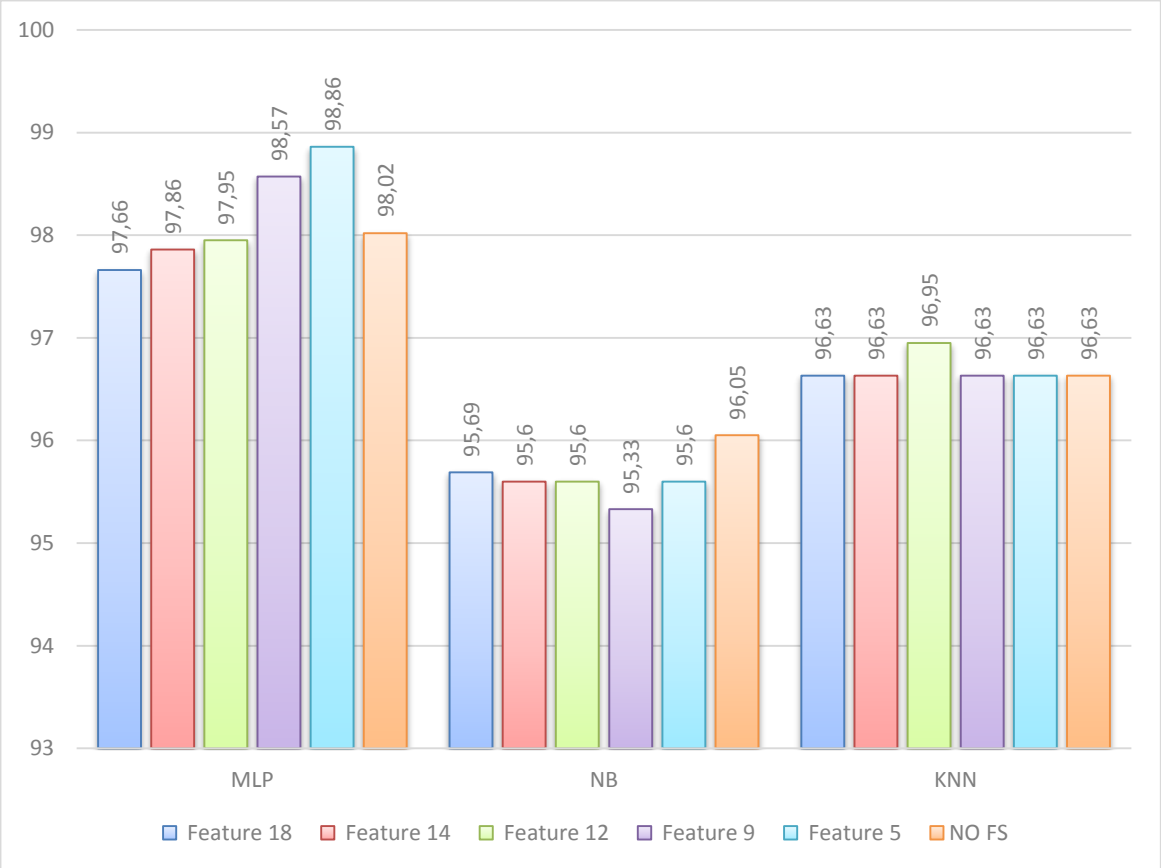


Figure 4.3 Accuracy Values for The Three Classifiers with Information Gain Method

Multi-Layer Perceptron recorded the best time in all the levels of information gain method compared to the other two classifiers as shown in Figure 4.4.

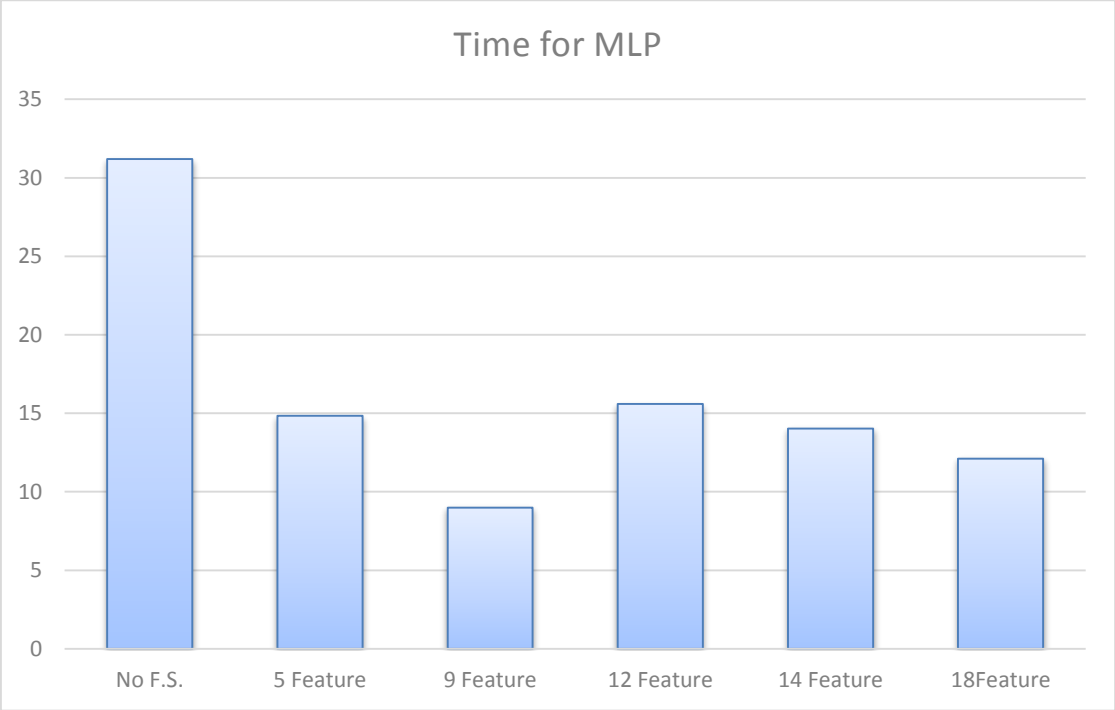


Figure 4.4 Time (in seconds) for MLP Classifiers with Information Gain Method

For Naive Bayes classifier, the recorded accuracy values ranged from (95.33 to 95.69) when the number of weight attributes was reduced from (5, 9, 12, 14) and finally (18). These values were close to the value recorded without feature selection method (96.05).it is evidence that NB did not show any improvement in the Accuracy values with all the levels used in the information gain method.

In respect to Time for Naive Bayes, the recorded time was enhanced with different levels of information gain(Figure 4.5). The best time recorded was with reduced (9) level where the time was (0.02) compared with the time recorded without feature selection(0.07).

For K-NN, the best recordedvalue for Accuracy was when using a reduced 12 attributes in IG feature selection method. This may give an indication that this classifier can be improved if more manipulation in the feature selection was adopted.

The time recorded by K-NN, was case dependent (Figure 4.6).The recorded valuesfor time were reduced but not in a proportional ratio with the levels used. The least time value was recorded for both (9)and (12) attributes (0.80 seconds) compared with time

values recorded without feature selection (0.91). The highest time value was for (18) attributes which was (2.36) seconds.

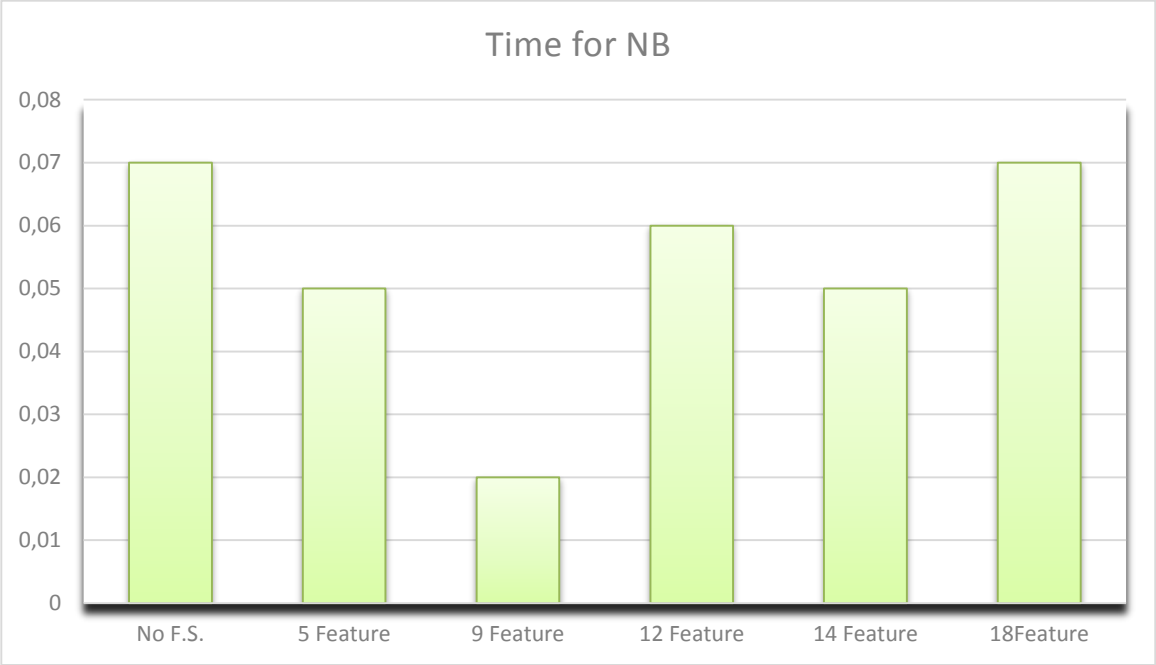


Figure 4.5 The Time Recorded by Naive Bayes Classifiers usingIG method

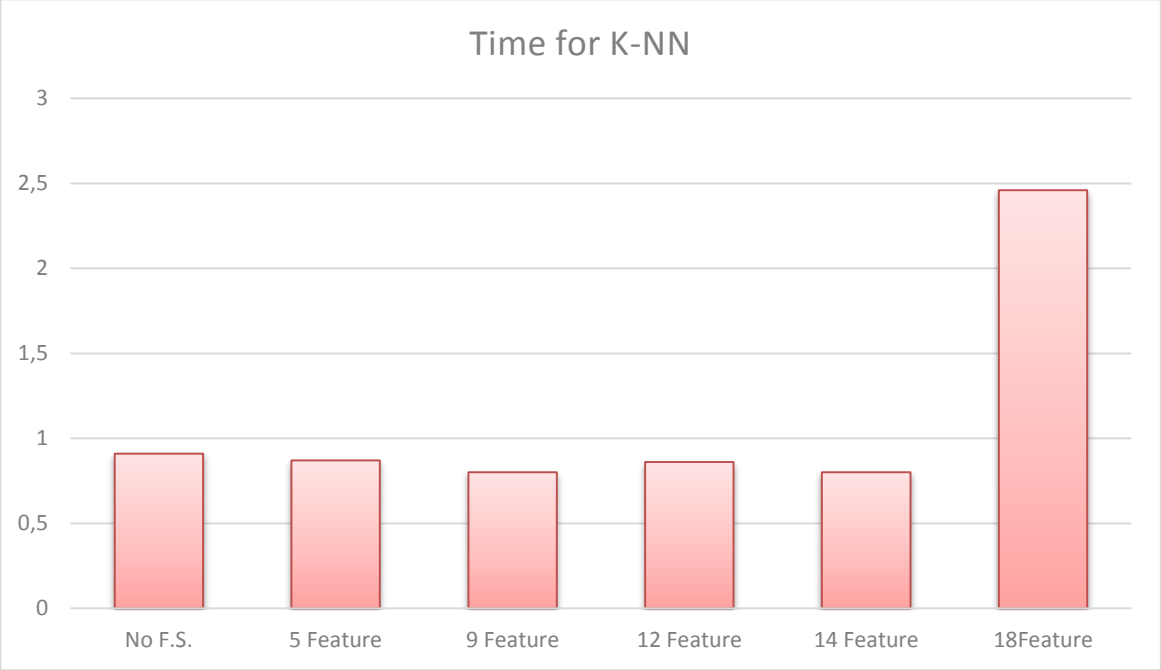


Figure 4.6The Time Recorded by K-NN Classifiers using IG method

4.2.2 Feature Selection with Information Gain Ratio Method

For Multi-Layer Perceptron, the Accuracy values using Information Gain Ratiowere raised up by reaching up to (98.93)as shown in Figure 4.7. Besides, the Time values recorded were enhanced dramatically in all the levels used as it was raised from (31.2)seconds without feature selection method reaching(8.05) seconds with Information Gain Ratio method as shown in Figure4.8.

For Naive Bayes using Information Gain Ratio both Accuracy and Time were enhanced. The Accuracy value recorded was (97.41) and time (0.01) secondsas shown in Figures 4.7& 4.9.

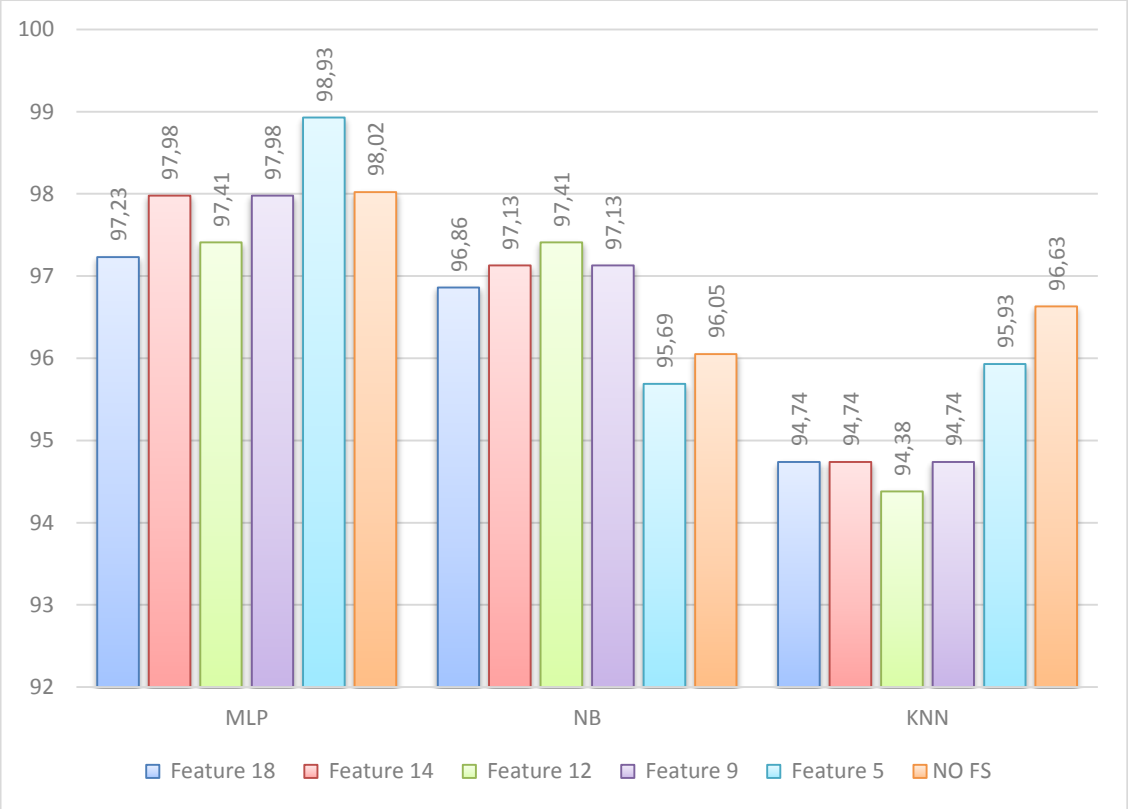


Figure 4.7 Accuracy Values for the Three Classifiers with Information Gain Ratio Method

Table 4.3The Time Recorded by Three Classifiers (IGR)

Classifier	NO F.S.	5 Feature	9 Feature	12 Feature	14 Feature	18 Feature
MLP	31.2	23.51	17.38	8.05	13.19	11.22
NB	0.07	0.11	0.05	0.01	0.07	0.04
KNN	0.91	0.99	3.64	1. 62	2.92	0.77

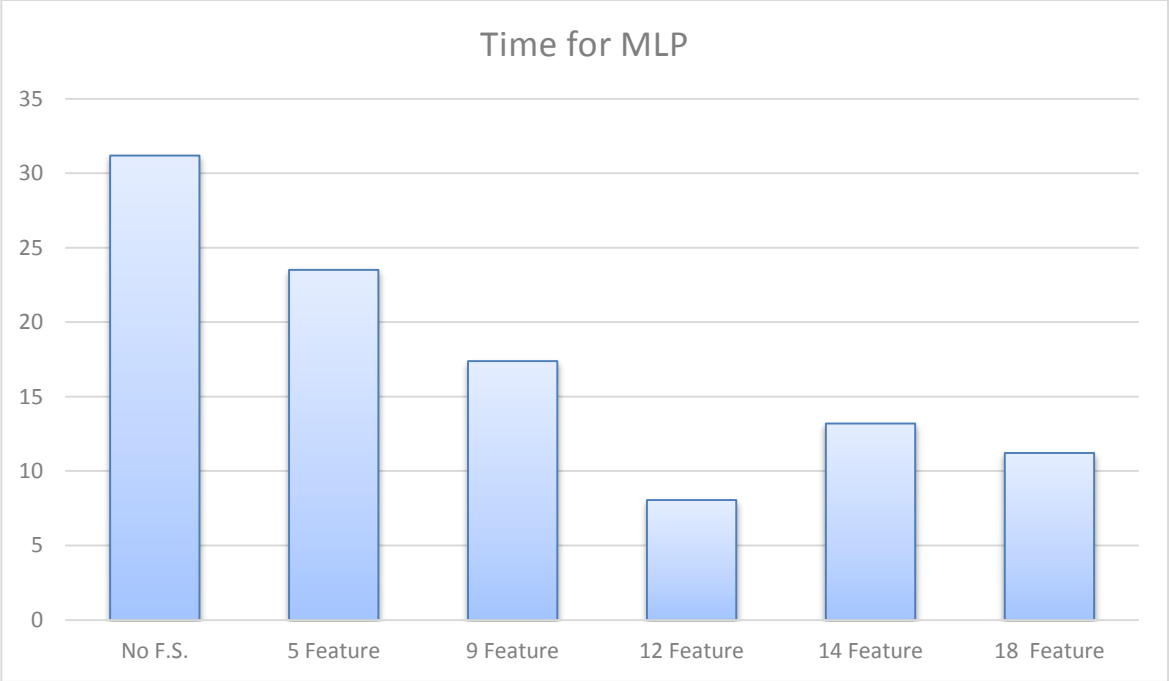


Figure 4.8Time for MLP Classifiers with Information Gain Ratio Method

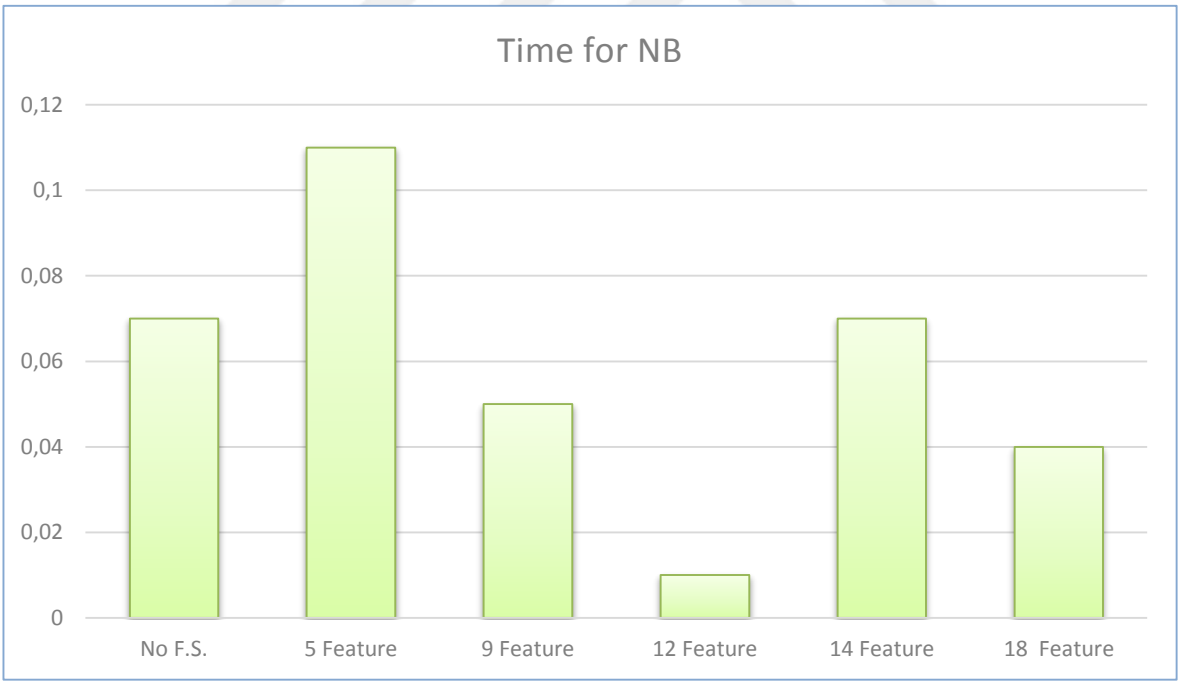


Figure 4.9Time for NB Classifiers with Information Gain Ratio Method

K-NN, on the other hand, did not provide any improvement for both the Accuracy and time when using the IGR method. In fact, the time values were higher in some of the levels of the feature selection used (Figure 4.7 and 4.10).

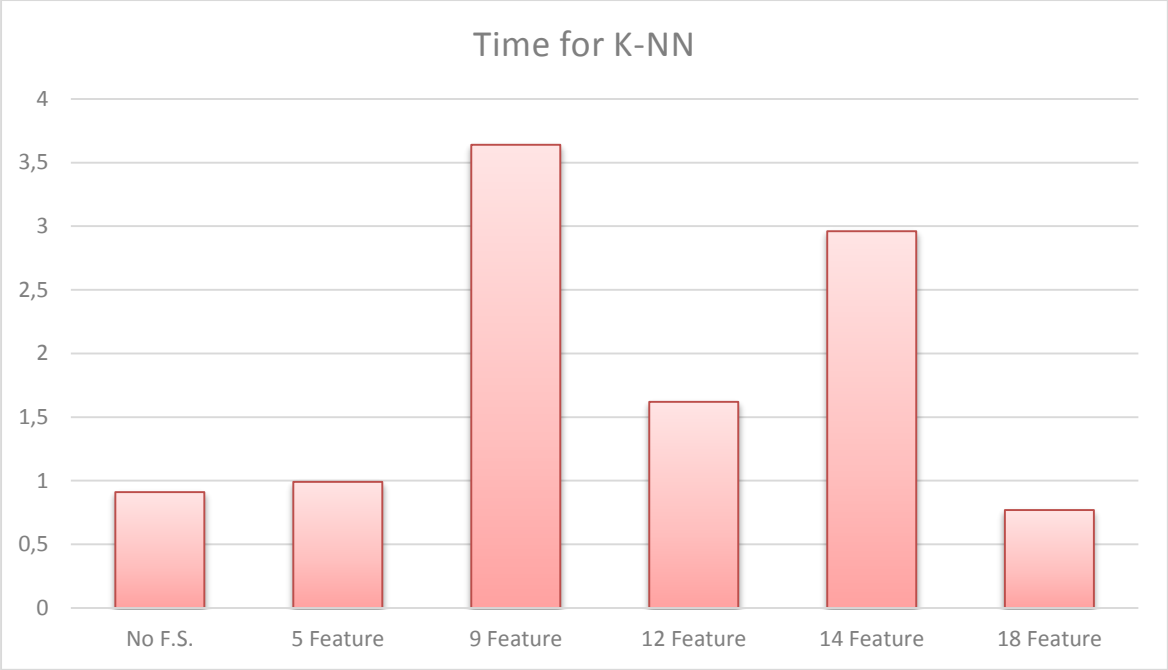


Figure 4.10Time for KNN Classifiers with Information Gain Ratio Method

4.3 Results Comparisons with other Studies

The results of this study were used to be compared with other similar studies. Although it is hard to simulate the criteria used in this study with other studies, comparisons are essential in order to clarify the causes in respect to results differences besides theses comparisons may give some indications about the limitations in the programming processes as well as enhance our prospect for future works.

For Multi-Layer Perceptron, using (5574) messages the Accuracy values recorded in this study with IG and IGR feature selection methods were (98.86 and 98.93). respectively). Anchal and Rimt-Iet (2014) reported a range of (80.8-84.84) Accuracy values by using only (5) messages [50]. They have suggested that in the MLP structure the classifier can be minimized/maximized via any gradient based feature selection procedure.

For Naive Bayes, the recoded values in this study for Accuracy and timewith IG were; (96.05) and (0.02) seconds respectively, while for IGR the accuracy was (97.41) and the time was (0.01) seconds.

Kim et al. 2015 with Naive Bayes used the same dataset that have been used in this study and reported an Accuracy of (96.92) which is somewhat close the values recorded in the current study. However, the time reported by them was (1.23 seconds) [10].

Nuruzzaman and Choi (2014) with Naive Bayes used 20 SMS as training data and 885 SMS as filtering and updating data. They have achieved Accuracy and time values; (90.17%) of (0.04 seconds) respectively, when misclassified SMS were used to update the filtering system. However, they reported higher accuracy values reached up to (95.32%) when they updated all the incoming SMS[2].

CHAPTER 5

CONCLUSIONANDFUTURE WORK

5.1 Conclusion

The aim of this study was to analyze SMS classifications with three of the most famous classification algorithms.The classification was based on text data to solve the problems of SMS spam. Filters were applied and their effects on the selection of features was determined. According to the results of the current study the folowing conclusion can be extracted:

- The three classifiers used; Artificial Neural Network (Multi-Layer Perceptron), Naïve Bayes and K-Nearest Neighbor, recorded different results for the Accuracy and Time irrespective to the feature selection methods used.
- The best Accuracy and Time values were recorded by Multi-Layer Perceptron with both feature selection methods; Information Gain & Information Gain Ratio.
- The best Time value was recorded withNaïve Bayes although the improvement of the Accuracy was limited.

- K-Nearest Neighbors recorded no difference in Accuracy with and without the feature selection methods with a very little improvement in Time.

5.2 Future Works

The field of mobile communication is a growing field that is becoming of an important source for marketing and profiting in a wide range of aspects. This encourages spammers to use SMS as an indispensable way to achieve their goals. However, as this process is an ongoing procedure, researchers are never to give up using all means to prevent and protect the mobile phone users from such delema.

The suggestions of future research may include the use of different types of feature selection methods. The classifiers can be adopted to use Wrapper instead of Filters. Additionally, the program processing can be used with the addition of stemming. Different kinds of classifier can be set using more than 26attributes.

In the implementation of algorithms, we will use cross-validationwithout dividing the data to 80% for exercise and 20% for testing. Besides, Implement the classification of SMS dataset on other filters such as deep learning, support vector machines.

APPENDIX-A

View code

A.1 Inthe following statement in C sharp (C#) program has been represent the set of steeps of `Feature extraction`

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.OleDb;
using System.Text;
using System.IO;
using System.Text.RegularExpressions;
using System.Linq;
```

```
namespace SMS_Classfication
{
class Feature
   {
public static List<string> TEXTS = new List<string>();
private List<string> _text = new List<string>();
public static double[,] FeaturesV;
public static double[,] NormlizedF;
public static int[] output;
private static double[] UpperCaseLetters = new double[5574];
private static int[] UpperCaseWords = new int[5574];
      private static string[] FName = { "Words", "Length","Letters", "Digits",
"Symbols", "money", "price", "prize", "win", "winner", "won", "call", "congratulation",
"http", "www", "sms", "txt", "urgent", "cash", "code", "subscribe", "guarntee", "offer",
"service", "customer", "xxx", "sex" };
private static string[] SpamWords = { "money", "price", "prize", "win", "winner",
"won","call","congratulation","http","www","sms","code","urgent","cash","code","subs
cribe","guarntee","offer","service","customer","xxx","sex" };
```

#region StopWords

```
#region StopWords
staticstring[] StopWord = { "a", "about", "above", "across", "afore", "aforesaid", "after",
"again", "against", "agin", "ago", "aint", "albeit", "all", "almost", "alone", "along",
"alongside", "already", "also", "although", "always", "am", "american", "amid", "amidst",
"among", "amongst", "an", "and", "anent", "another", "any",   "anybody", "anyone", "anything",
"are", "aren't", "around", "as", "aslant", "astride", "at", "athwart", "away", "b", "back",
"bar", "barring", "be", "because",  "been", "before", "behind", "being", "below", "beneath",
"beside", "besides", "best", "better", "between", "betwixt", "beyond", "both", "but", "by", "c",
"can", "cannot", "can't", "certain", "circa", "close", "concerning", "considering", "cos",
"could", "couldn't", "couldst", "d", "dare", "dared", "daren't", "dares", "daring",
"despite","did", "didn't", "different", "directly", "do", "does", "doesn't", "doing", "done",
"don't", "dost", "doth", "down", "during", "durst", "e", "each", "early", "either",
"em","english", "enough", "ere", "even", "ever", "every", "everybody", "everyone", "everything",
"except", "excepting", "f", "failing", "far", "few", "first", "five", "following","for", "four",
"from", "g", "gonna", "gotta", "h", "had", "hadn't", "hard", "has", "hasn't", "hast", "hath",
"have", "haven't", "having", "he", "he'd", "he'll", "her", "here","here's", "hers", "herself",
"he's", "high", "him", "himself", "his", "home", "how", "howbeit", "however", "how's", "I",
"id", "if", "ill", "i'm", "immediately", "important","in", "inside", "instantly", "into", "is",
"isn't", "it", "it'll", "it's", "its", "itself", "i've", "j", "just", "k", "l", "large", "last",
"later", "least", "left", "less","lest", "let's", "like", "likewise", "little", "living",
"long", "m", "many", "may", "mayn't", "me", "mid", "midst", "might", "mightn't", "mine",
"minus", "more", "most","much", "must", "mustn't", "my", "myself", "n", "near", "'neath",
"need", "needed", "needing", "needn't", "needs", "neither", "never", "nevertheless", "new",
"next", "nigh","nigher", "nighest", "nisi", "no", "no-one", "nobody", "none", "nor", "not",
"nothing", "notwithstanding", "now", "o", "o'er", "of", "off", "often", "on",
"once","one","oneself", "only", "onto", "open", "or", "other", "otherwise", "ought", "oughtn't",
"our", "ours", "ourselves", "out", "outside", "over", "own", "p", "past", "pending",
"per","perhaps", "plus", "possible", "present", "probably", "provided", "providing", "public",
"q", "qua", "quite", "r", "rather", "re", "real", "really", "respecting", "right","round", "s",
"same", "sans", "save", "saving", "second", "several", "shall", "shalt", "shan't", "she",
"shed", "shell", "she's", "short", "should", "shouldn't", "since","six", "small", "so", "some",
"somebody", "someone", "something", "sometimes", "soon", "special", "still", "such", "summat",
"supposing", "sure", "t", "than", "that","that'd", "that'll", "that's", "the", "thee", "their",
"theirs", "their's", "them", "themselves", "then", "there", "there's", "these", "they",
"they'd", "they'll", "they're","they've", "thine", "this", "tho", "those", "thou", "though",
"three", "thro'", "through", "throughout", "thru", "thyself", "till", "to", "today", "together",
"too","touching", "toward", "towards", "true", "'twas", "'tween", "'twere", "'twill", "'twixt",
"two", "'twould", "u", "under", "underneath", "unless", "unlike", "until", "unto",
"up", "upon", "us", "used", "usually", "v", "versus", "very", "via", "vice", "vis-a-vis", "w",
"wanna", "wanting", "was", "wasn't", "way", "we", "we'd", "well", "were",
"weren't", "wert", "we've", "what", "whatever", "what'll", "what's", "when", "whencesoever",
"whenever", "when's", "whereas", "where's", "whether", "which", "whichever",
```

```
public static DataTable DBConnect()
        {
            OleDbConnection con = new
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=DataSet.accdb");
con.Open();
string sql = null;
sql = "SELECT * from Dataset" ;
            DataTable dt = new DataTable();
            DataSet ds = new DataSet();
            System.Data.OleDb.OleDbDataAdapter da = new
System.Data.OleDb.OleDbDataAdapter();
ds.Tables.Add(dt);
da = new System.Data.OleDb.OleDbDataAdapter(sql, con);
da.Fill(dt);
con.Close();
return dt;
        }
public static String StopWords_Removal(string Text, int x)
        {
            Text = Text.ToLower();
for (int i = 0; i < StopWord.Length; i++)
                Text = Text.Replace(" " + StopWord[i] + " ", " ");
return Text;
        }
public static void Pre_processing()
        {
            DataTable DT = DBConnect();
string T;
for (int i = 0; i < DT.Rows.Count; i++)
            {
                T = StopWords_Removal(DT.Rows[i].ItemArray[1].ToString(), i);
TEXTS.Add(T); // Add the texts after cleaning
            }
        }
public static void Extraction()
        {
```

```
        FeaturesV = new double[5574, FName.Length];
int f = 0;
foreach(string Tx in TEXTS)
        {
FeaturesV[f, 0] = Tx.Split(' ').Length ;
FeaturesV[f, 1] = Tx.Length;   //  The length property of the message
FeaturesV[f, 2] = CountLetters(Tx);
FeaturesV[f, 3] = CountDigits(Tx);
FeaturesV[f, 4] = CountSymbols(Tx);
            //FeaturesV[f, 0] = CountWord(Tx, SpamWords[0]);
for (int i = 0; i < SpamWords.Length; i++)
FeaturesV[f, i + 5] = CountWord(Tx, SpamWords[i]);
f++;
        }


    }

private static double CountDigits(string T)
    {
      //  The number of digits relative to the length of the text
return (float)T.Count(char.IsDigit) / (float)T.Length;
    }
private static double CountLetters(string T)
{ // The number of characters relative to the message length
return (float)T.Count(char.IsLetter) / (float)T.Length;
    }
private static double CountSymbols(string T)
    {    //  The number of symbols relative to the length of the message
return Regex.Matches(T, "[~!@#$%^&*()_+{}:\"<>?.',-]").Count / (float)T.Length;
    }
private static int CountWord(string T, string W)
    {
        T = T.ToLower();
int x = 0;
        T = T.ToLower();

if (T.Contains(W))
            {
x++;
            }
return x;
    }
public static void Normlization()
    {
        DataTable TB2 = DBConnect();
output = new int[TB2.Rows.Count];
bool ff = false;
double min2;// = double.MinValue;
double max2;// = double.MaxValue;
```

```
        max2 = 0; min2 = 0;
        NormlizedF = new double[FeaturesV.GetLength(0), FeaturesV.GetLength(1)];
int nn = 0;
for (int i = 0; i < FeaturesV.GetLength(1); i++)
        {
FindMAXMIN(i, ref max2, ref min2);
for (int dr = 0; dr < FeaturesV.GetLength(0); dr++)
            {
if (ff == false)
output[dr] = TB2.Rows[dr].Field<string>(2).ToString() == "ham" ? 1 : 0;
if (max2 == 0 && min2 == 0)
NormlizedF[dr, i] = 0;
else
NormlizedF[dr, i] = Math.Truncate((double)(FeaturesV[dr, i] - min2) / (max2 - min2) *
1000) / 1000;
            }
ff = true;
        }
    }
public static void FindMAXMIN(int F, ref double Mx, ref double Mi)
    {
double x = double.MaxValue;
double y = double.MinValue;
for (int i = 0; i < FeaturesV.GetLength(0); i++)
        {
double col = FeaturesV[i, F];
            x = Math.Min(x, col);
            y = Math.Max(y, col);
        }
        Mi = x;
        Mx = y;
    }
public static void DisplayFeatures(System.Windows.Forms.DataGridView D)
    {
for (int c = 0; c < FeaturesV.GetLength(1); c++)
        {
if (c == 0) { D.Columns.Add(c.ToString(), "Sample"); }
else
            {
D.Columns.Add(c.ToString(), FName[c - 1]);
                D.Columns[c].Width = 80;
            }
        }
for (int i = 0; i < FeaturesV.GetLength(0); i++)
        {
D.Rows.Add();
for (int j = 0; j < D.ColumnCount; j++)
            {
if (j == 0) { D.Rows[i].Cells[j].Value = (i + 1).ToString(); }
else { D.Rows[i].Cells[j].Value = string.Format("{0:0.0}", FeaturesV[i, j - 1]); };
```

```
            }
        }
    }
public static void DisplayNormlization(System.Windows.Forms.DataGridView D)
    {
for (int c = 0; c < FeaturesV.GetLength(1); c++)
        {
if (c == 0) { D.Columns.Add(c.ToString(), "Sample"); }
else { D.Columns.Add(c.ToString(), FName[c - 1]); }
            D.Columns[c].Width = 100;
        }
for (int i = 0; i < NormlizedF.GetLength(0); i++)
        {
D.Rows.Add();
for (int j = 0; j < D.ColumnCount; j++)
            {
if (j == 0) { D.Rows[i].Cells[j].Value = (i + 1).ToString(); }
else { D.Rows[i].Cells[j].Value = string.Format("{0:0.000}", NormlizedF[i, j - 1]); }
            }
        }
    }
public static void SaveArff()
    {
        DataTable TB = DBConnect();
using (StreamWriter writer = File.CreateText("Data.arff"))
        {
writer.WriteLine("@relation SMS");
for(int f = 0; f <FeaturesV.GetLength(1);f++)
            {
writer.WriteLine("@attribute A" + f.ToString() + " real");
            }
writer.WriteLine("@attribute C {ham,spam}");
writer.WriteLine();
writer.WriteLine("@data");
for (int i = 0; i < FeaturesV.GetLength(0); i++)
            {
for(int j = 0; j < FeaturesV.GetLength(1); j++)
                {
writer.Write(FeaturesV[i, j].ToString() + ",");
                }
writer.WriteLine(TB.Rows[i].ItemArray[2].ToString());
            }
        }
    }
}
```

APPENDIX-A2

In appendix-A2, the area codes have been usedC # to classifications of data to the three algorithms are performed by calling the WIKA libraries in the code below then the two selection methods IG and IGR, are also shown in the code extension

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using weka.classifiers;
using weka.core;

namespace SMS_Classfication
{
classClssifiers
    {
publicstaticdouble[,] MLP(Instances _inst, int percentSplit,string LA ,  double
LR , int it , refdouble KCV)
        {
```

```
//Instances _inst = new Instances(new java.io.FileReader(path));
            _inst.setClassIndex(_inst.numAttributes() - 1);
java.util.Random Rnd = new java.util.Random(1);
weka.classifiers.functions.MultilayerPerceptron mlp = new
weka.classifiers.functions.MultilayerPerceptron();
weka.filters.Filter myRandom = new
weka.filters.unsupervised.instance.Randomize();
myRandom.setInputFormat(_inst);
            _inst = weka.filters.Filter.useFilter(_inst, myRandom);

int train = _inst.numInstances() * percentSplit / 100;
int test = _inst.numInstances() - train;

Instances TRAIN = newInstances(_inst, 0, train);

            mlp.setLearningRate (LR);
mlp.setMomentum(0.2);
mlp.setTrainingTime(it);
mlp.setHiddenLayers(LA);

mlp.buildClassifier(TRAIN);
Evaluation _eval = newEvaluation(_inst);
            _eval.crossValidateModel(mlp, _inst, 5, Rnd);
            KCV = _eval.pctCorrect();
int tp = 0, tn = 0, fp = 0, fn = 0;
for (int i = train; i < _inst.numInstances(); i++)
            {
Instance cI = _inst.instance(i);
double cP = mlp.classifyInstance(cI);
if (cP == _inst.instance(i).classValue() & cP == 0)
tp++;
elseif (cP == _inst.instance(i).classValue() & cP == 1)
fp++;
elseif (cP != _inst.instance(i).classValue() & cP == 0)
tn++;
elseif (cP != _inst.instance(i).classValue() & cP == 1)
fn++;
            }

//  System.Windows.Forms.MessageBox.Show(c.ToString() + " , " + test.ToString());
return ConfusionMatrix(tp, tn, fp, fn);
        }

publicstaticdouble[,] NB(Instances _inst, int percentSplit, refdouble KCV)//,ref
List<int> Matrix)
        {
//   Instances  _inst = new Instances(new java.io.FileReader(path));
            _inst.setClassIndex(_inst.numAttributes() - 1);
java.util.Random Rnd = new java.util.Random(1);
weka.classifiers.bayes.NaiveBayes nb = new weka.classifiers.bayes.NaiveBayes();
weka.filters.Filter myRandom = new
weka.filters.unsupervised.instance.Randomize();
myRandom.setInputFormat(_inst);
           _inst = weka.filters.Filter.useFilter(_inst, myRandom);

int train = _inst.numInstances() * percentSplit / 100;
int test = _inst.numInstances() - train;

Instances TRAIN = newInstances(_inst, 0, train);

nb.buildClassifier(TRAIN);
Evaluation _eval = newEvaluation(_inst);
```

```
            _eval.crossValidateModel(nb, _inst, 5, Rnd);
            KCV = _eval.pctCorrect();
int tp = 0, tn = 0 , fp = 0 , fn =0 ;
for(int i= train; i< _inst.numInstances();i++)
            {
Instance cI = _inst.instance(i);
double cP = nb.classifyInstance(cI);
if (cP == _inst.instance(i).classValue() & cP == 0)
tp++;
elseif (cP == _inst.instance(i).classValue() & cP == 1)
fp++;
elseif (cP != _inst.instance(i).classValue() & cP == 0)
tn++;
elseif (cP != _inst.instance(i).classValue() & cP == 1)
fn++;
            }

//  System.Windows.Forms.MessageBox.Show(tp.ToString() + " , " + test.ToString()
+ " , " +  tp + " , " + tn + " , "  + fp + " ,"  + fn  ) ;
return ConfusionMatrix(tp, tn, fp, fn);
        }

publicstaticdouble[,] KNN(Instances _inst, int percentSplit, int K, refdouble
KCV)//,ref List<int> Matrix)
        {
//Instances _inst = new Instances(new java.io.FileReader(path));
            _inst.setClassIndex(_inst.numAttributes() - 1);
java.util.Random Rnd = new java.util.Random(1);
weka.classifiers.lazy.IBk knn = new weka.classifiers.lazy.IBk(K);
weka.filters.Filter myRandom = new
weka.filters.unsupervised.instance.Randomize();
myRandom.setInputFormat(_inst);
            _inst = weka.filters.Filter.useFilter(_inst, myRandom);

int train = _inst.numInstances() * percentSplit / 100;
int test = _inst.numInstances() - train;

Instances TRAIN = newInstances(_inst, 0, train);
knn.buildClassifier(TRAIN);
Evaluation _eval = newEvaluation(_inst);
            _eval.crossValidateModel(knn, _inst, 5, Rnd);
            KCV = _eval.pctCorrect();
int tp = 0, tn = 0, fp = 0, fn = 0;
for (int i = train; i < _inst.numInstances(); i++)
            {
Instance cI = _inst.instance(i);
double cP = knn.classifyInstance(cI);
if (cP == _inst.instance(i).classValue() & cP == 0)
tp++;
elseif (cP == _inst.instance(i).classValue() & cP == 1)
fp++;
elseif (cP != _inst.instance(i).classValue() & cP == 0)
tn++;
elseif (cP != _inst.instance(i).classValue() & cP == 1)
fn++;
            }

//System.Windows.Forms.MessageBox.Show(c.ToString() + " , " + test.ToString());
return ConfusionMatrix(tp, tn, fp, tn) ;
        }

publicstaticdouble[,] ConfusionMatrix(int tp,int tn, int fp, int fn)
```

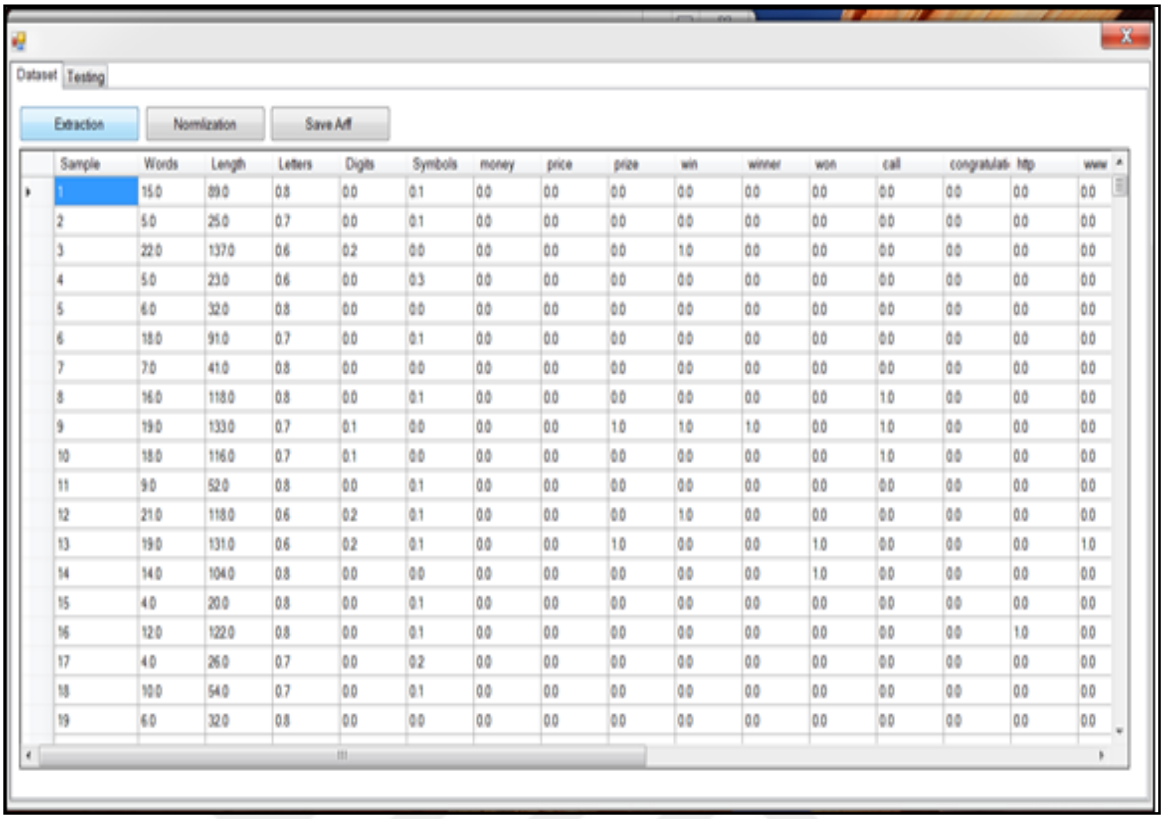
```
        {
double[,] EV = newdouble[2, 2];
EV[0, 0] = 100 * ((double)(tp + fp) / (tp + tn + fp + fn));  // accuracy
EV[0, 1] = ((double)(tp ) / (tp + fp));                 // Precision
EV[1, 0] = ((double)(tp) / (tp + fn ));  // Recall
EV[1, 1] = (2 * EV[0, 1] * EV[1, 0]) / (EV[0, 1] + EV[1,0]); // F-M
return EV;
        }

publicstaticList<double> GainRatio(Instances _insts)
        {

List<double> G = newList<double>();
            _insts.setClassIndex(_insts.numAttributes() - 1);
weka.attributeSelection.GainRatioAttributeEval gr = new
weka.attributeSelection.GainRatioAttributeEval();
gr.buildEvaluator(_insts);
for (int f = 0; f < _insts.numAttributes()-1; f++)
G.Add(gr.evaluateAttribute(f));            // calculate the GR for all attributes
return G;
        }

publicstaticList<double> InfoGain(Instances _insts)
        {
List<double> G = newList<double>();
// Instances _insts = new Instances(new java.io.FileReader(path));
            _insts.setClassIndex(_insts.numAttributes() - 1);
weka.attributeSelection.InfoGainAttributeEval gr = new
weka.attributeSelection.InfoGainAttributeEval();
gr.buildEvaluator(_insts);
for (int f = 0; f < _insts.numAttributes()-1; f++)
G.Add(gr.evaluateAttribute(f));          // Calculate the IG for all attributes
,,,
return G;
        }

publicstaticInstances  FeatureSelection(string FS , int F, Instances _inst)
        {
// FS ... The name of the method ...
// F .. Number of features to be removed ..

List<double> LFS = newList<double>();
// Rank the features by using FS algorithm
if(FS == "IG")
            {
                LFS = InfoGain(_inst);
            }
elseif(FS == "GR")
            {
                LFS = GainRatio(_inst);
            }
int x;
for(int i = 0; i < F; i++)

            {
                x = LFS.IndexOf(LFS.Min());
                _inst.deleteAttributeAt(x);
                LFS.RemoveAt(x);
            }
return _inst;// NB(_inst,80);
        }
```

```
    }
}
```


APPENDIX-B

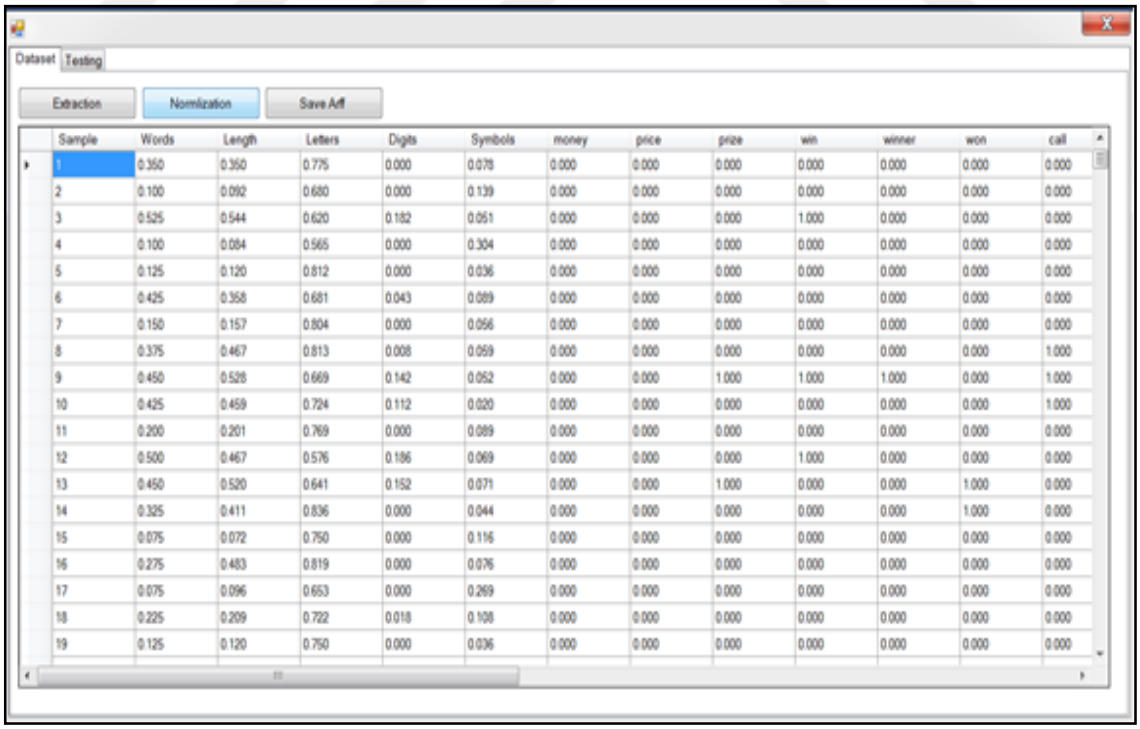
View execution

The implementation results are shown by using the Visual Studio program environment that is shown below for the some required operations.



Sample	Words	Length	Letters	Digits	Symbols	money	price	prize	win	winner	won	call	congratulat	http	www
1	15.0	89.0	0.8	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	5.0	25.0	0.7	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	22.0	137.0	0.6	0.2	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
4	5.0	23.0	0.6	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	6.0	32.0	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	18.0	91.0	0.7	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	7.0	41.0	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	16.0	118.0	0.8	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
9	19.0	133.0	0.7	0.1	0.0	0.0	0.0	1.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0
10	18.0	116.0	0.7	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
11	9.0	52.0	0.8	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	21.0	118.0	0.6	0.2	0.1	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
13	19.0	131.0	0.6	0.2	0.1	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0
14	14.0	104.0	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
15	4.0	20.0	0.8	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16	12.0	122.0	0.8	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
17	4.0	26.0	0.7	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
18	10.0	54.0	0.7	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
19	6.0	32.0	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure B.1 Extracting Feature before Preprocessing



Sample	Words	Length	Letters	Digits	Symbols	money	price	prize	win	winner	won	call
1	0.350	0.350	0.775	0.000	0.078	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	0.100	0.092	0.680	0.000	0.139	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	0.525	0.544	0.620	0.182	0.051	0.000	0.000	0.000	1.000	0.000	0.000	0.000
4	0.100	0.084	0.565	0.000	0.304	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	0.125	0.120	0.812	0.000	0.036	0.000	0.000	0.000	0.000	0.000	0.000	0.000
6	0.425	0.358	0.681	0.043	0.089	0.000	0.000	0.000	0.000	0.000	0.000	0.000
7	0.150	0.157	0.804	0.000	0.056	0.000	0.000	0.000	0.000	0.000	0.000	0.000
8	0.375	0.467	0.813	0.008	0.059	0.000	0.000	0.000	0.000	0.000	0.000	1.000
9	0.450	0.528	0.669	0.142	0.052	0.000	0.000	1.000	1.000	1.000	0.000	1.000
10	0.425	0.459	0.724	0.112	0.020	0.000	0.000	0.000	0.000	0.000	0.000	1.000
11	0.200	0.201	0.769	0.000	0.089	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12	0.500	0.467	0.576	0.186	0.069	0.000	0.000	0.000	1.000	0.000	0.000	0.000
13	0.450	0.520	0.641	0.152	0.071	0.000	0.000	1.000	0.000	0.000	1.000	0.000
14	0.325	0.411	0.836	0.000	0.044	0.000	0.000	0.000	0.000	0.000	1.000	0.000
15	0.075	0.072	0.750	0.000	0.116	0.000	0.000	0.000	0.000	0.000	0.000	0.000
16	0.275	0.483	0.819	0.000	0.076	0.000	0.000	0.000	0.000	0.000	0.000	0.000
17	0.075	0.096	0.653	0.000	0.269	0.000	0.000	0.000	0.000	0.000	0.000	0.000
18	0.225	0.209	0.722	0.018	0.108	0.000	0.000	0.000	0.000	0.000	0.000	0.000
19	0.125	0.120	0.750	0.000	0.036	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Figure B.2 Final Data Format after Normalization

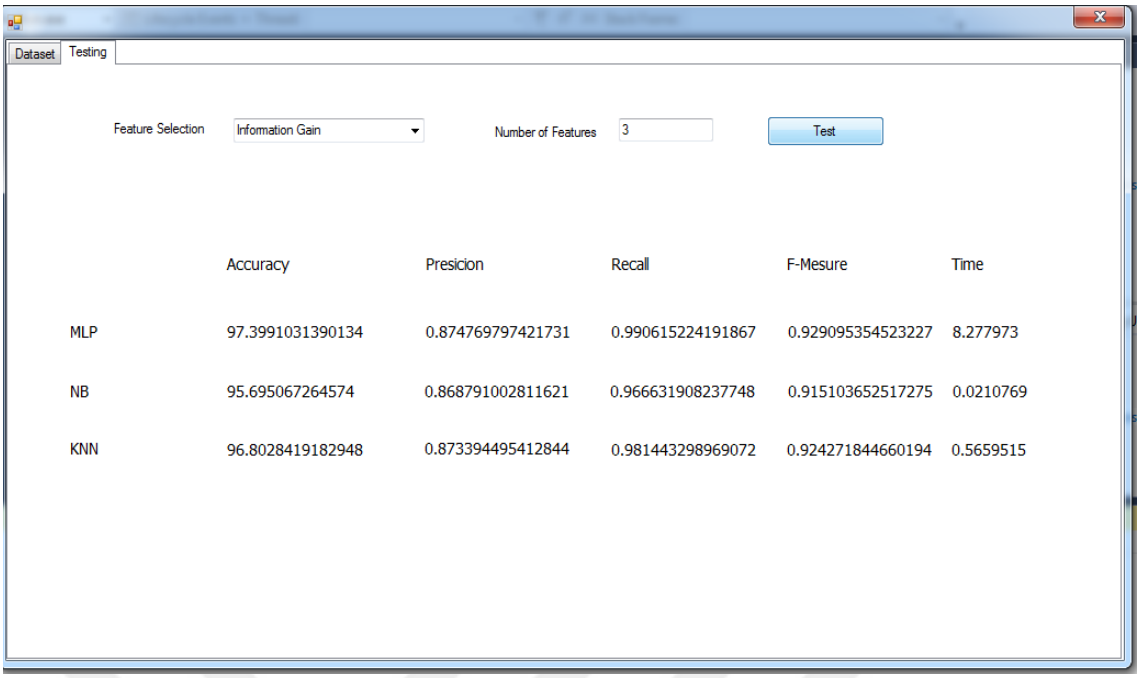


Figure B.3Implementation results when using IG

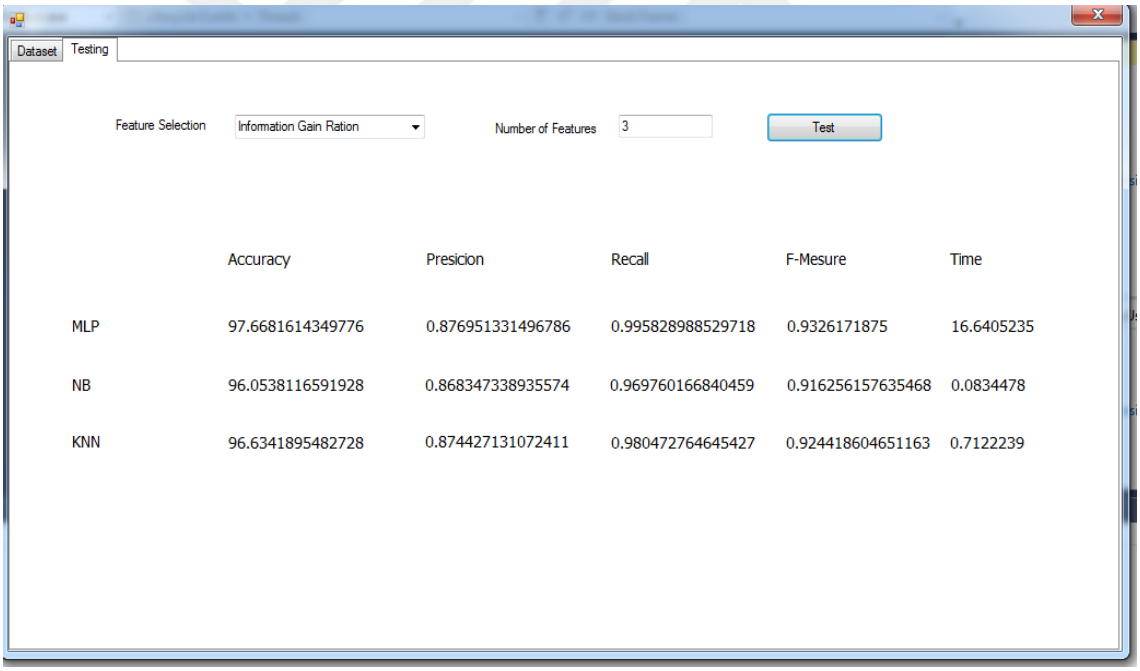


Figure B.4 Implementation results when using IGR