THE UNIVERSITY OF TURKISH AERONAUTICAL ASSOCIATION
INSTITUE OF SCIENCE AND TECHNOLOGY

SPAM FILTER BASED ON NAÏVE BAYES AND LEVY-FIREFLY ALGORITHM

MASTER THESIS

Ahmed J. H. ALHALLAQ

THE DEPARTMENT OF INFORMATION TECHNOLOGY
THE PROGRAM OF INFORMATION TECHNOLOGY

SEPTEMBER 2019

THE UNIVERSITY OF TURKISH AERONAUTICAL ASSOCIATION
INSTITUE OF SCIENCE AND TECHNOLOGY

SPAM FILTER BASED ON NAÏVE BAYES AND LEVY-FIREFLY ALGORITHM

MASTER THESIS

Ahmed J. H. ALHALLAQ

1403660017

THE DEPARTMENT OF INFORMATION TECHNOLOGY
THE PROGRAM OF INFORMATION TECHNOLOGY

Supervisor: Assist. Prof. Dr. Shadi AL SHEHABI

Türk Hava Kurumu Üniversitesi Fen Bilimleri Enstitüsü'nün 1403660017 numaralı Yüsek Lisans öğrencisi **"Ahmed J. H. ALHALLAQ"** ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı **"SPAM FILTER BASED ON NAÏVE BAYES AND LEVY-FIREFLY ALGORITHM"** başlıklı tezini aşağıda imzaları bulunan jüri önünde başarı ile sunmuştur.

Supervisor : **Assist. Prof. Dr. Shadi AL SHEHABI**
Türk Hava Kurumu Üniversitesi ....

Assist. Prof. Dr. Meltem İMAMOĞLU
Türk Hava Kurumu Üniversitesi ....

Jury Members : **Assist. Prof. Dr. Abdül Kadir GÖRÜR**
Çankaya Üniversitesi ....

Assist. Prof. Dr. Shadi AL SHEHABI
Türk Hava Kurumu Üniversitesi ....

Thesis Defense Date: 10/09/2019

STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all the information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

10/09/2019

Ahmed J. H. ALHALLAQ

ACKNOWLEDGMENTS

I would like to express my deepest thanks to almighty GOD for helping me to finish this study.

I would like to express my special thanks to Assist. Prof. Dr. Shadi AL SHEHABI for his direct supervision, supports, encouragement, sharing of experiences and guide me through my research period.

Besides my supervisor, I would like to thank the rest of my research committee; Assist. Prof. Dr. Meltem İMAMOĞLU and Assist. Prof. Dr. Abdül Kadir GÖRÜR for their encouragement, insight full comments and questions.

I would like to express my special thanks to my mother and to the spirit of my father who always encouraged me. I wish he could be by my side as I fulfill this achievement. I dedicate this research to Palestine and to all martyrs of Palestine, especially my cousin who was like a brother to me.

I thank my loving wife “Miray” who has always encouraged me and has been helpful to me

I thank my brothers Raed, Nael, Hassan, Mohammed and my sisters Jihan and Ahlam

I express my gratitude to everyone who supported me and encouraged me in this endeavor.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

LFA	Levy flight + Firefly Algorithm
DoS	Denial of Service
NB	Naïve Bayesian
MI	Mutual Information
DR	Detection Rate
FAR	False Alarm Rates
ARPANET	Advanced Research Projects Agency Network
MTA	Mail Transfer Agent
MDA	Mail Delivery Agent
DNS	Domain Name System
MAP	Maximum Posterior
SFS	Supervised Feature Selection
KNN	K-Nearest Neighbor
FA	Firefly Algorithm
MS	Microsoft
UCI	University Of California Lrvine

ABSTRACT

SPAM FILTER BASED ON NAÏVE BAYES AND LEVY-FIREFLY ALGORITHM

ALHALLAQ, Ahmed J. H.
Master, Department of Information Technology
Thesis Supervisor: Assist. Prof. Dr. Shadi AL SHEHABI

September-2019, 81 Page

Internet is rapidly developing due to quick innovation in technology and communication. Users and organization can store and effectively access data, applications, and services. Amongst the best and speediest approaches to connect and send/receive data from one place then onto the next is through electronic mail (e-mail). As a result, it became the target for several malicious attacks such as e-mail phishing, spoofing, and spam emails. Protecting various users from spam emails became the subject of several research, both in academia and industries, were several methods and approaches are developed and still being tested. Recently, the use of machine learning is proposed were several algorithms are trained to detect and separate spams from legitimate emails. This thesis proposes a new approach for filtering out spam e-mails through the use of effective hybrid meta-heuristic optimization algorithm (Levy flight + firefly algorithm) with Naïve Bayes classifier. Preliminary results of various experiments conducted on SPAM dataset revealed that the proposed hybrid method detects unwanted spam e-mails with ~95% accuracy compared to only 79% when using Naïve classifier alone, in other words, 16% improvement. Further, the execution time is fast and the proposed algorithm results are fairly stable; proven through standard deviation.

Keywords: Classification; Feature Selection; Firefly Algorithm; Levy flight Algorithm; Naïve Bayesian

ÖZET

NAİF BAYES VE LEVY-FİREFLY ALGORİTMASINA DAYALI SPAM FİLTRESİ

ALHALLAQ, Ahmed J. H.
Yüksek Lisans, Bilişim Teknolojileri Anabilim Dalı
Tez Danışmanı: Yrd. Doç. Dr. Shadi AL SHEHABI

Eylül-2019, 81 Sayfa

İnternet, teknoloji ve iletişimdeki hızlı yenilikler nedeniyle hızla gelişmektedir. Kullanıcılar ve kuruluşlar, verileri, uygulamaları ve hizmetleri depolayabilir ve etkili bir şekilde erişebilir. Bir yerden veri almak ya da bir yere veri göndermek için kullanılan en iyi ve en hızlı araçlardan biri de elektronik posta (e-posta) olmuştur. Ancak bu mecra zaman içinde e-posta dolandırıcılığı (e-kimlik hırsızlığı), sahtekârlık ve spam e-postalar gibi çeşitli kötü amaçlı saldırılar için açık hedef haline gelmiştir.

Kullanıcıları spam e-postalardan korumak amacıyla hem akademide hem de endüstri camiasında çeşitli yaklaşım ve yöntemler geliştirilmiş olup bu konudaki çalışmalar devam etmektedir.

Bu tez, hybrid meta-heuristic optimization algorithm (Levy flight + firefly algorithm) ve Naïve Bayes sınıflandırıcısı kullanılarak spam e-postalarını filtrelemek için yeni bir yaklaşım önermektedir. SPAM veri setinde yapılan çeşitli deneylerin ön sonuçları, önerilen karma yöntemin, yalnızca Naïve sınıflandırıcısını kullanırken yalnızca% 79'la karşılaştırıldığında % 95 doğrulukla istenmeyen spam e-postaları tespit ettiğini ortaya koymuştur. Başka bir deyişle % 16 iyileşme kaydedilmiştir. Ayrıca, uygulama süresi hızlıdır ve önerilen algoritma sonuçları oldukça kararlıdır; bu sonuç standart sapma ile kanıtlanmıştır.

Anahtar Kelimeler: Classification; Feature Selection; Firefly Algorithm; Levy flight Algorithm; Naïve Bayesian.

CHAPTER ONE

INTRODUCTION

1.1. Background

Electronic mail (colloquially known as e-mail) is a fast, efficient and cheap method for the exchange of messages through the internet. It is the preferred form of communication (both formal and informal) as it cuts across all continents and is used by more than 2.3 billion people globally. The number of active email users by the year 2016 increased to about 4.3 billion [1]. This increased dependence on and use of e-mail has elicited several problems arising from illegal and illegitimate usages in the form of spam emails [2].

The Text Retrieval Conference (TREC) defines the term spam as any unsolicited or unwanted email which is indiscriminately sent. These spam e-mails are often sent in large batches without a request or desire of the recipient. They seek to transmit malware that promotes the propagation of undesired advertisements, phishing messages, explicit content, fraud schemes, and promotions. These spam emails can have negative effects on the recipients. These effects can range from loss of work productivity, individual annoyance, and reduction in reliability of e-mails, misuse of bandwidth, wastage of storage space, inefficient use of computational power, increased susceptibility to viruses, Trojan horses and worms, to monetary losses due to Denial of Service (DoS), directory harvesting attacks, and phishing. [3].

The quantity of these spam emails has been increasing within the past decade and has become a cause of a serious security threat as it continues to cause severe damage to individuals, economies, and businesses.

Amateur marketers and advertisers take advantage of the chance of reaching a vast amount of people via email to cause or initiate several attacks or promotions with little or no cost. One of the most notorious of these, is the penny stock spam, which is one of the most famous spam topics of discussion over the internet. The spammers obtain and use compromised brokerage accounts to purchase sizable quantities of penny stocks and market them through the use of wide e-mail campaigns, pointing share value increase over time. When just a fraction of these spam recipients proceeds to buying the promoted stocks, the fraudsters make gains. The unwitting investors can be fooled into buying the stocks and this results in a spike in the demand price of the shares. As soon as the spam emails are delivered, the fraudster immediately sells their stocks at the increased price while the fooled investors are left trying to sell their own. The stock spam is a long used strategy that gained more interest in the year 2013. It was conveyed by the Security Threat Report 2014 that about 50 % of the total spam emails are penny stock or ‘pump and dumps’ emails [4][5].

The concept of spam is broad and appears in many forms, and yet it is to be understood completely. Generally, there are numerous forms of spam, these include chat spams which is special to chat rooms, blog spams which mainly target blogs and bloggers (splogs), web spams which work by misdirecting search engines in a process known as ‘spamdexing’ or ‘search engine spamming’, and social spams which affect social systems. The focus of this thesis is solely on email spam and its different forms; not spams in general.

Multiple forms of anti-spam filters are currently available, and they are designed to work as manual patterns. These filters are made up of matching rules which need to be adapted to each incoming e-mail message. While these filters work, they require experience and time to be developed efficiently. Furthermore, the features and content of all unwanted (spam) messages (e.g. Marketed products and frequently used terms) change with time and as a result it requires the matching rules to be updated periodically. However, for a new system to be feasible, significant benefits must be obtained over the existing system by any system that offers automatic classification of spam from real and wanted messages.

In literature, machine learning algorithms have been utilized in classification of texts. These algorithms aid in the classification of documents into different categories. The classification is made based on the content of these texts or documents. These algorithms are developed by first training them on manually classifying documents. These same kinds of algorithms are also implemented in e-mail threads, in classifying e-mails to different folders, and in identifying interesting news and many others [7, 8].
The Naive Bayesian (NB) classifier is a growing area of interest in the sub-field of data mining which contributes to categorizing (spam and useful) messages manually, reporting excellent precision, and recalling the messages that are unseen. It might be of benefit to know that the task of text categorization can be more efficient with anti-spam filtering, compared to any other task of text categorization. It is the behavior of sending many e-mail messages quickly and rashly that makes these e-mails ‘spam’ messages and not the actual content of the email. However, it is clear that the languages used in spam messages are unique in that spam messages rarely have the topic mentioned in the body portion of the message. Thus it makes it easy to train a text classifier to filter out spam messages.[6, 9].

In certain cases of feature selection, a slight advance has been shown. The selection of suitable characteristic of text messages is defined as Mutual Information (MI). MI is computed for each word in all classes; the probability of a word appearing in e-mail messages of each class is defined as p (of that word). The classifier that is used to decide whether or not a message arriving in the inbox is of use to the recipient is the spam filter. This method was developed exploiting the existence of recurrent words or phrases which are mutual to all or most junk e-mails. There also exist other spam filtering methods which utilize ‘black’ and ‘white’ lists [1]. E-mail filters can be implemented to examine either the message content (consist of several features that are a group of words) or the message’s subject field. Therefore, the input for email classification may be seen in the form of two-dimensional matrix, of which the axes consist of the features and messages.

The main task of e-mail classification can be further divided into many smaller sub-tasks. The first of which is the collection of data and the representation or modelling of a specific problem, for example; e-mail messages. The second sub-task is the selection and reduction of the features of the e-mail to lower the dimensionality of the problem, for instance, the number of features used for the other sub-task steps [10][11].

Naive Bayes classifier was proposed to identify spam in 1998. Bayesian classifier operates on the basis of relied events, the probability of such an event happening in the future, and furthermore it could also be identified from its past occurrence. This technique can be used for classifying spam, in this context, word probabilities can perform the leading role. If any words always occur as it does in spam, but not as it occurs in the useful e-mail, it follows that this e-mail may be a spam email. The NB classifier technique has become a widely used method in e-mail filtering software. However, this filter needs to be trained to classify e-mails efficiently [12, 13] . The Bayesian classifiers ascertain attributes such as common keywords or phrases and allocates to them probabilities. Then, each word has an allocated specific occurrence probability in spam or useful e-mail stored in its database. If the total computed probabilities of the words used in the email exceeds a certain defined threshold, then, the filter will assign the e-mail to one of the two categories (either spam or useful). Generally, all of the statistics-based spam filters utilize Bayesian probability computation to collect the individual token statistics into a single unified outcome. The Bayesian filtering decision is contingent on this unified outcome [10, 12].

1.2. Problem Statement

Preceding studies have reported that the efficacy of a spam filter model primarily is dependent on the improvement of the identification of classifiers, and the retraining of the reference models. Choosing a feature is a significant issue in the filtering of spam. Feature selection comprises of the selection of sub-feature to characterize a class of data. Feature selection has two important aspects which are to first filter out the noise and then to eliminate redundant features that may result in a significant loss in the detection accuracy. The Levy flight-Firefly [13] algorithm will be evaluated and validated in this thesis and more emphasis will be placed on the feature selection process using the suggested algorithms as an alternative to the conventional approaches. The outcome of this study will determine both the possibility of optimizing the selected algorithms and its accuracy as a detection system.

1.3. The Aim of Thesis

This thesis proposes the use of a feature selection algorithm as an enhancement of Naïve Bayes classifier for filtering spam emails. Enhancing NB classification accuracy is met through selecting the relevant features by applying levy flight-firefly optimization algorithm (LFA). The proposed filtering hybrid approach consists of NB classifier and LFA aim to improve the detection rate (DR), accuracy and decrease false alarm rates (FAR).

1.4. Objectives

1. Using LFA algorithm as feature selection algorithm to enhance Naïve Bayesian classifier for spam filtering.

2. To analyze the proposed algorithm and validate its performance with other published work.

1.5. The Scope

It is evident that the term *spam* has evolved to and used to refer to unsolicited messages in blogs, websites and text messages. However, this thesis will only focus on spam email messages, specifically with bulk spam and phishing messages. Spear phishing, a highly targeted and focused attack will not be analyzed. The criteria for selecting a classifier will be based on finding a classifier with an adequate performance and accuracy, and widely known to academic literatures.

CHAPTER TWO

LITERATURE REVIEW

2.1. Introduction

Spam e-mails can contain viruses, chain letters, advertisements, political advocacy or fraud attempts. Spam emails are also widely called junk or unsolicited e-mails. In 1975, the problem of junk emails had grown to a point that it h ad become an issue, urging Joe Postel to request for comments regarding the issue. The issue has been ever growing since then, to the extent that today each user is familiar with what junk or spam emails are. Among the different classifiers that exist, the Naïve Bayes (NB) classifier which is based on the Bayes theorem is a simple probabilistic classifier with a strong independence on the hypothesis. The NB classifier supposes the existence or absence of a single class of feature is not relevant to the existence or absence of any other feature, rather it is independent. The NB algorithm is conditional probability-dependent and utilizes the Bayes theorem to compute the probability by tallying the value frequencies and combinations. The Bayes theorem is capable of computing the probability of an event occurring by using the probability of another event which occurred in the past [5][14][15]. This chapter elaborates on the history of e-mail, spam, spam filtering methods/systems, NB classifier usage as a spam filter, feature selection methods, the description of the spam base dataset used in the proposed study, and the evaluation of the measurement performances.

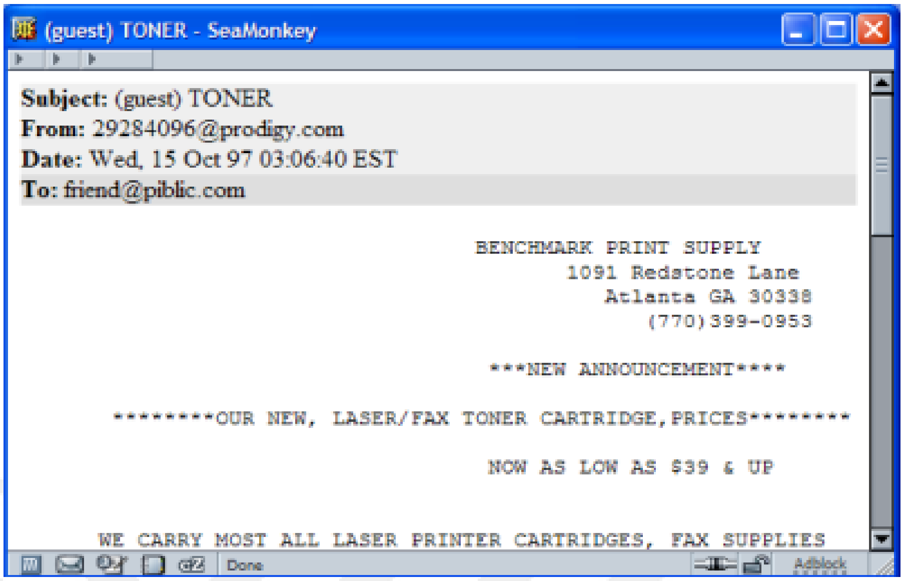
2.2. Electronic Mail (E-mail) History

E-mail is a widely-used communication system both in the professional and personal worlds. It is effective, enjoyable and is being constantly improved with developments of the internet.

It allows users to connect and contact with each other regardless of the distance between them or the time of the day. Moreover, it allows the user to contact another user whether or not that user is available at the time of communication. [15]. By January 2007, the estimated number of global Internet users was more than one billion with each of these users having one or more e-mail accounts. By December 2007, the estimated number of global Internet users had risen to more than four billion. This number is a drastic increase from the number of active Internet users documented in 1971 when Ray Tomlinson of the Department of Defense sent an email message of his work on the Advanced Research Projects Agency Network (ARPANET). The messaging applications available on the ARPANET systems were primarily used for local messaging. Since the launch of ARPANET over the internet, e-mail has become a globally used messaging system both for business and individuals. Owing to the small size of the ARPANET and with it being a trusted system, there was a demand for its security. With the proliferation of Internet usage, the demand for security has greatly increased in the messaging platforms [1][15][19].

2.3. Spam E-mail

Spam is defined as either one message or a sequence of messages which are all unsolicited and are sent as part of a larger collection of messages. All these messages share the same markedly identical contents. It was suggested by the Association of Direct Marketing that messages with certain contents such as pornography be tagged as spam, however this idea did not gain traction as it seen by many as an effort to legalize and legitimize the other forms of spam [15][20][21]. Another significant class of spamming is phishing, which is the hunt for sensitive and protected information such as but not limited to credit card numbers and passwords, by mimicking formal requests from trusted authorities or companies such as banks, service providers, marketing, and server administrators. These kinds of spams are conducted by utilizing the company’s defining characteristics such as the logo, fonts, and colors. Figure 2.1 shows a sample of such a spam email [15][22].



Subject: (guest) TONER
From: 29284096@prodigy.com
Date: Wed, 15 Oct 97 03:06:40 EST
To: friend@piblic.com

BENCHMARK PRINT SUPPLY
1091 Redstone Lane
Atlanta GA 30338
(770)399-0953

NEW ANNOUNCEMENT*

********OUR NEW, LASER/FAX TONER CARTRIDGE,PRICES********

NOW AS LOW AS $39 & UP

WE CARRY MOST ALL LASER PRINTER CARTRIDGES, FAX SUPPLIES

Figure 2.1 Marketing Spam

Fraudulent e-mails contain links which redirect their recipients to a website made by the fraudulent entity. Most fraudulent e-mails also contain links which redirect to specific sections of the legitimate company website. Moreover, it coaxes the recipients to carry out certain actions by using an e-mail address that resembles that of the company they are emulating (such as @paypal.com, @ebay.com) [15][22]. Another kind of malicious spam content are viruses; these are sometimes used to disturb the work of the mail server. All in all, the sender of spam messages uses one of the following strategies: imitates the offer of goods or ideas, sending of malicious software to obtain private information, and cause a temporary crash of the mail server [21][24].

Spam emails come in multiple forms, these include [20][23][24]:

1. **Penny Stock spam**: This spam promotes a cheap stock for people to buy
2. **Online Casino spam**: This type of spam promotes online gambling through online casinos
3. **Pirate Software spam:** This type of spam offers fake products at a much cheaper price compared to the original products
4. **Fake Degree spam**: The spammers promote fake academic certificates that are for sale

5. **419 scams:** These kind of spams usually ask for help to recover money or assets from a foreign country by first paying a small sum of money
6. **Lottery spam:** These spams are similar to the 419 scams as they consist of asking the people to pay money or assets to reclaim a ‘supposed’ lottery win

The spam base dataset used in this thesis was gathered from personal e-mails and spam e-mails. Spam e-mails are of the aforementioned classes. Spam e-mails can be subdivided into many topics and genres according to the different types of legitimate e-mails they mimic such as order confirmations, memos, and letters. The features of spam traffic are different from that of legitimate emails, specifically relating to the time of delivery. While legitimate e-mails are typically delivered during day time, spam e-mails are delivered at a consistent rate regardless of the time of day. Multiple methods are used by spammers to conceal their identity, most often they harvest e-mail addresses from websites with an open identity. It follows then that the recognition of the act of identity harvesting would aid in the identification of spammers [2][19][21].

2.4. Spam Filter System

Spam filters can be implemented at all layers; firewalls are deployed at the front of e-mail servers’ ad also at mail transfer agents (MTA). E-mail servers offer built-in solutions to protect against viruses and spams, and fully protects e-mail at the at the boundary level of the network to prevent unwanted emails from arriving at the network. Spam filters can also be integrated into the mail delivery agent (MDA) level to serve the customers directly. Integration at this level allows users to install personalized spam filters that can automatically filter any e-mail according to user selected criteria [25][26]. The different spam filter techniques can be categorized as follows: it is composed of e-mail header analysis, static algorithm, keyword checking, IP based filtering and list-based filtering. The list-based filtering can be further classified into three categories which are Whitelist, Blacklist, and Grey list. The static algorithm can be divided into rule-based and content-based filtering. Lastly, the IP-based filtering includes reverse-lookup [27–29]. Details are as follow:

E-mail header checking is a widely-known and used method. It is based on a defined set of rules that are used to check e-mail headers. If the e-mail header matches the rules, the mail server resends the messages back to the sender. The rules are: ‘From’ field is blank and ‘To’ field is packed with numerous addresses from the same source in a single e-mail, and there exists a large number of digits in the e-mail addresses (commonly used method to generate false addresses). Likewise, it can return the messages if the code of the language stated in the message header is matched with the set of rules.

Keyword checking is greatly used for spam filtering. It operates by scanning the subject and body of an email. To improve such a method, keyword combinations are used, and it is regarded as a suitable solution. Word combinations that frequently occur in spam are assigned to a list, and any messages that contains these word combinations can be blocked. Such a list however needs to be updated regularly.

The Blacklist filter is comprised on a sender server IP address and domain name which are stored in a list called as a ‘Blacklist’. Any e-mail coming from an IP address and domain name included in the ‘Blacklist’ will be directly sent to spam folder or deleted. Conversely, this method of filtering has some limitations; the first is that spammers may use different IP addresses and multiple domain names to circumnavigate this filter. It may become difficult to frequently update the ‘Blacklist’ to keep up with spammers. Secondly, the Blacklist filtering could result in a legitimate e-mail being mis-identified as spam due to a minimum control of the blacklisting.

The Whitelist filtering is the polar opposite of the Blacklist. In this method each user saves the trusted e-mail contacts in the ‘Whitelist’. Each email coming from a sender in the ‘Whitelist’ will be accepted while those not in the list will be rejected. This technique also has its limitations; when addresses are added to the list, it requires rewriting of the old records and thus steady updating is necessary. Another issue is that if the spammer e-mail address was once added to the whitelist, it can provide access to all other addresses listed in the ‘Whitelist’.

Greylist filtering is another method used for fileting which can be installed on the reciever mail server and/or on the personal reciever side as an anti-spam application. This method works by firstly rejecting all received emails. This policy stops spammers as they do not resend the rejected e-mail as it wastes their time, rather they search for other e-mail addresses that is not found in the ‘Grey list’. In addition, based on the spammer behavior, the Grey list provides the contact list of the mail servers using this approach; as a result, the spammer would avoid sending future messages to these servers after the initial rejection. The spammer can detect the structure of the filtering systems used in email servers, in order to bypass the filter, the spammer has to update their operating techniques. However, the most significant problem with this method is the possibility of losing legitimate messages.

The content-based filtering technique utilizes machine learning to filter out spam e-mails. In order to get adequate results, the administrator of the mail server must train the filters to execute the required tasks properly. This filtering system operates by relying on some predefined words after completely receiving the e-mail. These predefined words are gathered through statistical reports that are based on the words accumulated from previous known spams.

The rule-based and content-based filtering systems share a lot of similarities but are different. The rule-based filtering system utilizes some set rules and regulars to categorize the messages which are accepted or blocked. The main issue with the content-based and rule-based filtering methods is the possibility of verifying the rules and words by the programmer which may result in variable restrictions. The first of the restrictions is that the policies and databases have to be updated periodically. Secondly, the spammers are aware of how the filters operate and thus they will constantly attempt to send the spam e-mails by adjusting characters in an attempt to validate their e-mails. Finally, these techniques require the whole e-mail content to make the classifications, thus the mail server needs to receive the whole email and operate on it which may increase the needed time for classification.

The reverse lookup method is also known as the reverse domain name system (DNS) lookup where the host can have a specified IP address. With this method, the receiver can confirm the identity of the sender domain. However, this technique is not effective for mobile users and those users with an invalid IP address.

2.5. Naive Bayesian Theorem

The Bayesian theorem was named after the 18th century English nonconformist priest named Thomas Bayes, who did early work on decision theory and probability. The Bayesian classifiers can predict the class membership probabilities such as the probability of a given tuple which fit into a given class, as such they are viewed to be statistical classifiers. The probability is calculated from the Bayesian expression as given in equation 2.1 [30–32]

$$\mathrm{P(H|X)} = \frac{\mathrm{P(X|H)P(H)}}{\mathrm{P(X)}} \tag{2.1}$$

Where P (H) = $\frac{\mathrm{H|}}{\mathrm{N}}$ = the predetermined h probability when |h| and N were known and “|h|” is defined as the number of the pattern in class "H", and "N" is the total number of patterns. Likewise, it is assumed that all the hypotheses are relatively equal to P(X|H) which is defined as the conditional probability of "X" , in other words; the condition on h, and P(X) = the prior probability of "X" (note: this value is a constant). "X" is the "evidence" in the Bayesian expression and it is typically governed by the measurement of a set of n attributes. Assume that "H" is a specific hypothesis (for instance, the data tuple "X") and it belongs to a specific class "C". Problems of classification will merit the determination of P (H|X) (Eq. 2.1). The hypothesis probability denoted as "H" contains the observed data tuple denoted as "X"(or "evidence"), in other words, searching for the probability that tuple "X" fits to class "C" if the attribute of "X" is known. The term P(H|X) denotes the post-determined ‘H’ probability conditioned on "X" [32]. The hypothesis of maximum posterior (MAP) is used to signify that class h has maximum P(h|X). It can be formulated using Eq. 2.2 [31]:

$$h_{MAP} = \arg {}^{max}_{h\varepsilon H} = \arg {}^{max}_{h\varepsilon H} P(x|\, h)P(h) \tag{2.2}$$

The NB classifier shows any provided pattern X as an "n-dimensional" vector of attribute values [a_1, a_2,..., a_n]. The length of the provided classes L are C_1, C_2, . . . , C_L; the classifier is capable of predicting any unknown sample "X" as fitting to the class due to it having the uppermost post-determined probability which is conditioned on X (this is to say, "X" is allotted to the class C_i if and only if shows in Eq 2.3 [33]:

$$\mathrm{P}(\mathrm{C_i}|\mathrm{X}) > P(\mathrm{C_j}|\mathrm{X}) \qquad (2.3.)$$

For 1< j< i and j ≠ i

With regard to Eq. 2.1.

$$\mathrm{P}(\mathrm{C_i}|\mathrm{X}) = \frac{\mathrm{P}(\mathrm{X}|\mathrm{C_i})\mathrm{P}(\mathrm{C_i})}{\mathrm{P}(\mathrm{X})} \qquad (2.4.)$$

Nonetheless, to decrease the required computational expenses, the classifier makes a Naive or a simple assumption of n features being independent of each other. Therefore, status of the layer is independent and can be formulated as:

$$\mathrm{P}(\mathrm{X}|\mathrm{C_i}) = \Pi_{\mathrm{j=1}}^{\mathrm{n}}\mathrm{P}(\mathrm{a_j}|\mathrm{c_i}) \qquad (2.5.)$$

As P(X) is constant for each class and $\mathrm{P}(\mathrm{C_i}) = \frac{|\mathrm{C_i}|}{\mathrm{N}}$, therefore, it necessitates $\mathrm{P}(\mathrm{X}|\mathrm{C_i})$ to be maximized which as a result greatly decreases the cost of computation since it only counts the distribution of class [30-33].

2.5.1. Reasons for using Naive Bayesian in Spam Filtering

Spam is an ever-growing problem which is becoming more and more problematic each day. Studies have demonstrated that more than 50% of all the current e-mails are spam. The Radicati Group predicted that this percentage will increase to 70% by 2007 and by the year 2017 over 60% of customers will be affected. Spammers are becoming cleverer as time passes and they constantly figure out ways to circumvent the filters that are in use. Static techniques are being widely used in filtering software, meaning that it is simple to dodge the filters by changing the message a little. As a result, spammers examine the existing anti-spam techniques and develop methods of dodging or escaping them [25].

A spam filter is a program that prevents any spam messages from reaching the inbox by blocking it. They are used to prevent the flooding of the inbox with unsolicited messages. The existing spam filters can be classified into two main groups. These are server-side and client-side spam filtering systems. The server-side filtering systems are located on the e-mail servers and they deal with the incoming mail. Majority of the e-mails are marked as spam before the user downloads them or they will be deleted entirely. The client-side filter works on the already downloaded e-mails by examining the e-mail and making a decision about what should be done [36]. For numerous reasons, the NB classifier is considered as the best filter and the following are some of the declarants [12][33][34]:

1. The Bayesian method analyzes the entire message and recognizes and identifies keywords used in spam. It finds out words that identify valid e-mails; for instance, not all email messages which have the word ‘free’ or ‘money’ is a spam mail. The advantage offered by the Bayesian method is that it considers only the most relevant words which are obtained from their divergence from the mean. It makes use of probability to decide on the class of a given message. The Bayesian method is able to recognize the name of the business contact that sent the e-mail and classifies their messages as legitimate. Even if the classifier finds the interesting words ‘free’ or ‘money’ in that e-mail, it allows the words to balance each other in order to decide on the e-mail category.

2. The Bayesian filters are continuously self-adapting. They improve themselves by learning from new spams and legitimate e-mails. They evolve and adapt to counter-act new spam techniques. For example, when a spammer starts with the word ‘m-o-n-e-y’ instead of money, they can dodge keyword verification until the word ‘m-o-n-e-y’ is added to the keyword database. On the other hand, the Bayesian filter can automatically recognize such strategies as the word ‘m-o-n-e-y’ does not show up in legitimate e-

mails. Thus when the word ‘m-o-n-e-y’ shows up in an email, it is an indicator to the NB classifier that the e-mail is spam.

3. This technique is very sensitive to the user. It can understand and adapt to the habit of the specific e-mail users. As an example, the word ‘mortgage’ can be an indicator for spam however if a financial institution that regularly deals with mortgages runs this filter, e-mails containing the word ‘mortgage’ will never be flagged or filtered.

4. This technique is not restricted to a single language, it is an international multi-lingual anti-spam filter which can be used for any language. Generally, all the keyword lists are offered in English only and as a result become completely useless in non-English speaking regions. Moreover, this filter can also account for the language deviations or different usages of specific words in different regions even if the language spoken was the same. Due to this filter’s intelligence, it can catch more spam mails than other static filters.

5. It is not a simple task to evade this filter unlike the keyword filter. A smart spammer who seeks to subvert the Bayesian filter can choose to either use fewer words, which often signal the email is spam, or more words which are commonly used in legitimate e-mails. Implementing the latter case is practically impossible as the smart spammer has to be aware of the e-mail recipient profile which is unique to each recipient and tailor the message to each of them. As spammers work by taking advantage of bulk messaging, their tactics become useless against a Bayesian filter.

2.5.2. Bayesian Classification as Spam Filter

The Bayesian classification assumes that if a message contains specific words, it will decidedly tell whether or not a message is spam. This basic idea can be made general by using the probability theorem as in equation 2.6. This idea is constructed on two categories: S denoting spam, and L denoting legitimate e-mails. In addition, it is based on the message’s probability distribution, or to put it more accurately, the feature vectors given to the messages corresponding to each class. $P(X|C)$ represents the probability of receiving a message with feature vector "X" from class "C". Typically, the user of the Bayes classifier should be informed of how the distribution works and should be able to understand the given message "X" and its category "C" "produced". Thus, the probability is $\mathrm{P(C|X)}$. This is exactly what the user of the classifier obtains if they use the Bayes' rule.

$$\mathrm{P(C|X)} = \frac{\mathrm{P(x|c)P(C)}}{\mathrm{P(X)}} = \frac{\mathrm{P(x|c)P(C)}}{\mathrm{P(x|s)P(S)+P(x|l)P(L)}} \qquad (2.6)$$

Where $\mathrm{P(X)}$ represents the predetermined probability of the message "X". $\mathrm{P(C)}$ denotes the predetermined probability of the class "C" (in other words, the probability of a message showing up in each class). Hence, if the user of the classifier gets the values $\mathrm{P(C)}$ and $\mathrm{P(X|C)}$ (for $\mathrm{C} \in \{S, \mathrm{L}\}$), it will become possible to compute $\mathrm{P(C|X)}$, which is a useful accomplishment that then enables the application of the proceeding classification rule:

$$\text{If } \mathrm{P}(S|\mathrm{X}) > P(L|X) \qquad (2.7)$$

In other words, if the post-determined probability of "X" being a spam is larger than the post-determined probability of "X" being a legitimate e-mail, then, "X" will be classified as a spam, else, it will be classified as a legitimate e-mail. This is referred to as the rule of maximum posterior probability (MAP). The Bayes formula can be revised to the following form to be used for the filtering of spam:

$$\mathrm{X} = \begin{cases} \frac{\mathrm{P(X|S)}}{\mathrm{P(X|L)}} > \frac{\mathrm{P(L)}}{\mathrm{P(S)}} & \text{"spam"} \\ \text{otherwise} & \text{"legitimate"} \end{cases} \qquad (2.8)$$

The target of a Bayesian classifier is to identify spams, to do this it has to find the probabilities $\mathrm{P(X|C)}$ and $\mathrm{P(C)}$ for any "X" and "C". Although, the user of the classifier neither has the exact $\mathrm{P(X|C)}$ nor $\mathrm{P(C)}$ but can utilize the training data to approximate them. For example, the term $\mathrm{P}(S)$ can be estimated by the ratio of the spam message to total messages from the data used for training. The approximation of

$P(X|C)$ is more complicated and is dependent on how the classifier selects the feature vector "X" for message "M".

2.5.3. Advantages of Naive Bayesian

This section presents some advantages of the NB classifier [12][34].

- It permits the programming of interdependencies between the variables and predicting of events utilizing the combined prior knowledge and the given data.

- The NB classifier is able to simplify the computations and works efficiently and quickly when applied to large databases.

- The Bayesian classifier provides theoretical justification for other classifiers that do not employ Bayes theorem.

2.5.4. Disadvantages of Naive Bayesian

The disadvantages involved with NB are [12][34]:

- The results obtained from the NB are similar to the results of threshold-based systems however a significantly larger computational effort is required when Bayesian method is employed.

- The NB indicated imprecise hypotheses for its use, such as class-conditional independence.

- NB requires probability data which are not available.

- The NB method assumes the conditional independence of data attributes even though this may not always be the case. Nevertheless, it should be mentioned that in spite of this imprecise assumption, the NB classifier provides suitable results as it places emphasis on the introduction of the classes, for instance, not for the precise probabilities).

These disadvantages of the NB classifier can be mitigated by hybridizing it with the Firefly Optimization to improve the spam filtering system. The NB assumes that the data attributes are conditionally independent whereas the Firefly Algorithm does not make this assumption (i.e. data attributes are not considered conditionally independent).

2.6. Feature Selection

The dimensionality of the data used in machine learning and data mining has significantly increased over the previous three decades. Significant issues exist in the learning methods with comparatively high dimensionality. These learning models, having multiple features, tend to over fit and presents unacceptable performance. To solve the foundational cause of data dimensionality, numerous techniques for the simplification or reduction of the data dimensionality have been studied in literature and these field has become an significant aspect of machine learning and data mining researches [35][36]. Among these techniques, one of the most commonly used is feature selection. Feature selection attempts to represent a large data set, by using a smaller set of appropriate subsets based on evaluation criteria. This procedure typically boosts learning performance by lowering the computational cost and providing better model interpretability. Depending on the labeling of a training set, there are different classifications of feature selection algorithms. These are supervised and unsupervised feature selection algorithms, and semi-supervised feature selection algorithms [37–40].

Likewise, the supervised feature selection (SFS) algorithms can be further categorized into the wrapper, filter, and embedded models. In the filter model, feature selection is detached from the classifier learning, this is done so that there is no interference of the bias of the two operation algorithms. It is dependent on the dimension of the general features of training data (such as the distance, correlation, dependency, consistency and information. Examples of the filter model include the Relief, Fisher score, and Information Gain Based methods. The wrapper model defines the quality of selected features by utilizing the predictive accuracy of a predetermined learning algorithm. These methods become computationally expensive to run data with several features. Based on the weaknesses of each model, a proposition for an embedded model was presented to connect the gap which presently exists between the wrapper and filter models. The embedded model makes use of statistical criteria similar to the filter model for the selection of individual feature subsets with a certain cardinality. Furthermore, it selects those subsets with the optimal classification accuracy. As a result, the embedded model is able to achieve a similar accuracy to the wrapper model with equivalent efficiency to the filter model. In the embedded model, feature selection is performed during the learning process, i.e. feature selection and model fitting is completed simultaneously. Significant attention has been given to the development of the unsupervised feature selection models in literature [29][35][38].

Search problems can become less constrained in the absence of class labels depending on the measured clustering quality for unsupervised feature selection [40]. It is capable of evaluating multiple similarly valid feature subsets. A difficulty that may be faced by unsupervised feature selection is the recovery of relevant features when dimensionality is high but without considering further constraints. A further issue is how to objectively measure the obtained feature selection results [13][41]. In this thesis, unsupervised feature selection is outside the scope but additional information and a detailed review of the unsupervised feature selection can be found in literature [42].

The supervised feature selection review feature relevance with the aid of the Labeled information, but a good selection demand numerous labeled data although this can be time consuming. Alternatively, unsupervised feature selection can operate with unlabeled data, however the evaluation of the relevant features becomes hard. A dataset with a small labeled-sample size but a high level of dimensionality can be executed. Such a dataset allows for a large hypothesis space, but with few constraints. This combination of data features (i.e. high-level dimensionality with small labeled-sample size) poses a new research problem. By making the supposition that both the unlabeled and labeled data are obtained from a common population which is generated by the target concept, both unlabeled and labeled data are used to estimate the relevance of the features in semi-supervised feature [12][13][29].

Feature weighting is considered as the generalization of feature selection [29], however in feature selection, a binary weight is given to each feature, where 1 denotes a chosen feature and 0 represents otherwise. In feature weighting however, values assigned to each feature are usually in the range of [0,1] or [-1,1]. A greater value represents a more prominent or significant feature. Majority of the algorithms for the assigning of feature weight utilize a combined (global) weight for each feature over all occurrences. Meanwhile, there may be variations in the relative importance, noise and relevance across the different dimensions with data locality. In a few local feature selection algorithms, the local feature selection is exclusively performed to a specific test instance. This is mutual to so called 'lazy learning algorithms' like KNN [43][44]. The idea is to conduct feature selection or feature weighting during the process of classification (instead of during the training process) as a knowledge of the test instance will improve the model's ability for feature selection.

There are four distinctive steps in feature selection [38], they are the generation of the subset, evaluation, stopping criterion, and validation of results. Throughout the subset generation step, candidate feature subsets are selected depending on a provided search strategy and directed to the second step where evaluation is carried out based on a particular evaluation criteria. Those subsets that fits most closely fits the evaluation criterion are chosen from the evaluated candidate features after reaching the stopping criterion. In the last step, the chosen subsets are validated by the use of a validation set or domain knowledge.

2.6.1. Feature Selection for Classification

The supervised learning method is necessary in multiple real-world problems of classification where the core class probabilities and class-conditional probabilities are unknown, and each occurrence has a class label. Frequently in the real-world, there is very limited knowledge of relevant features. Therefore, for an improved representation of the domain, there is an introduction of multiple candidate features which causes there to be irrelevant and superfluous features to the target concept. Relevant features are neither irrelevant nor superfluous to the target; an irrelevant feature has no direct relation to the target, but it may impact the learning process. Likewise, redundant features do not contribute in any way to the target. The learning of good classifiers is continually difficult in many classification problems due to the large data size even before the removal of the unwanted ones. The computational time of the learning algorithms can be decreased by decreasing the number of the superfluous features and this can result in more general classifiers. It is beneficial to get improved knowledge of the basic concept of the classification of real-world problems [45][46][47].

Figure (2.2) shows the general framework of the feature selection methods for classification. The process of feature selection primarily impacts the training phase of classification. After the features have been generated, the subsets will be chosen through feature selection and data processing using the specific features of the learning algorithm instead of immediately processing the data with the complete features of the learning algorithm. The phase of feature selection may differ from filter models in that it may be independent from learning algorithms. However, the phase may repetitively deploy the learning algorithm execution for evaluation of the type of chosen features as in the wrapper models. Once the selected features are obtained, the classifier is triggered to the prediction phase.

Feature selection for classification typically tries to choose the feature subset with the least size corresponding to the following criteria:

- The accuracy of classification is not considerably reduced.

- Only using the values of the selected features, the resulting class distribution is effectively the same (as much as possible) as the original class distribution [45][46][47].

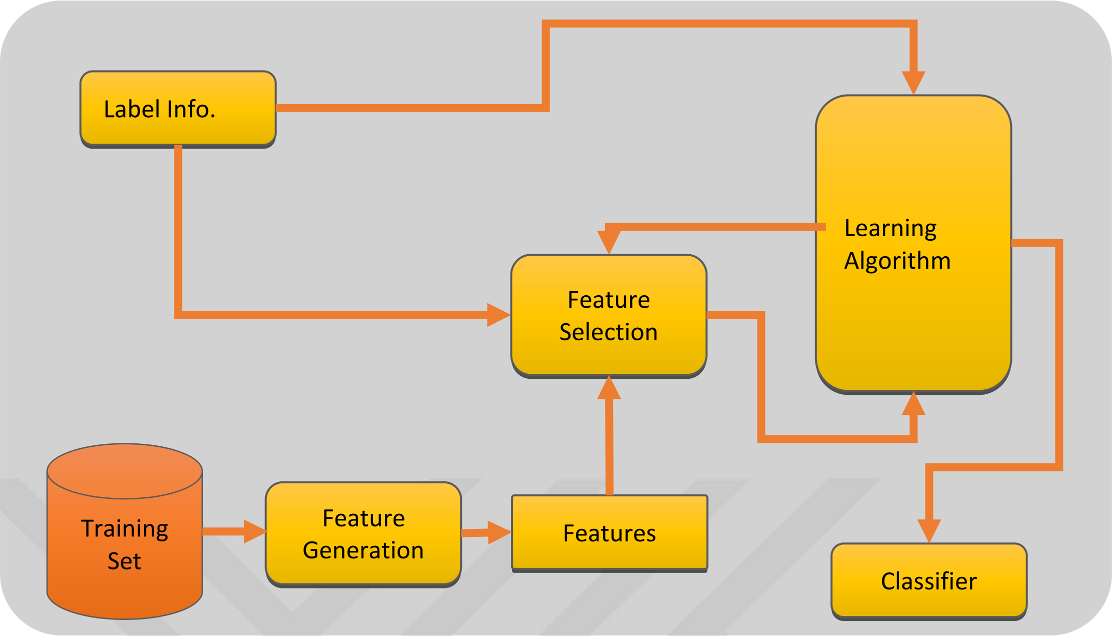


Figure 2.2 A Generalized Framework of Feature Selection for Classification

Feature selection methods ideally attempt to find the best feature subset by searching through the subsets of features for the optimal amongst the contending 2m subsets using defined evaluation functions. This process is exhaustive because it seeks to find the best subset only. The cost of such a process may be high and essentially prohibitive in practice even for medium-sized features (m). Other heuristics or random search-based algorithms concede their performance in exchange for a reduced computational complexity. They require a stopping criterion in order to avoid an exhaustive search for a subset. Herein this chapter the feature selection for classification was broken down into three classes based on the feature structure, these are: methods for flat features, streaming features, and for structured features as shown in Figure 2.3. The coming sections will offer a review of the three aforementioned classes, each with a representative algorithm. Before continuing, it is needed to introduce the notations which have been adopted in this chapter. Let F = {f1, f2, . . . , fm} and C = {c1, c2, . . . , cK} denote the feature set and the class label set correspondingly, where m and K represent the feature numbers and labels, correspondingly. X = {x1, x2, . . . , x3} ∈ R m×n denotes the data, where n is the number of instances, and the label information of the *i*-th instance xi is represented as y_i [47].

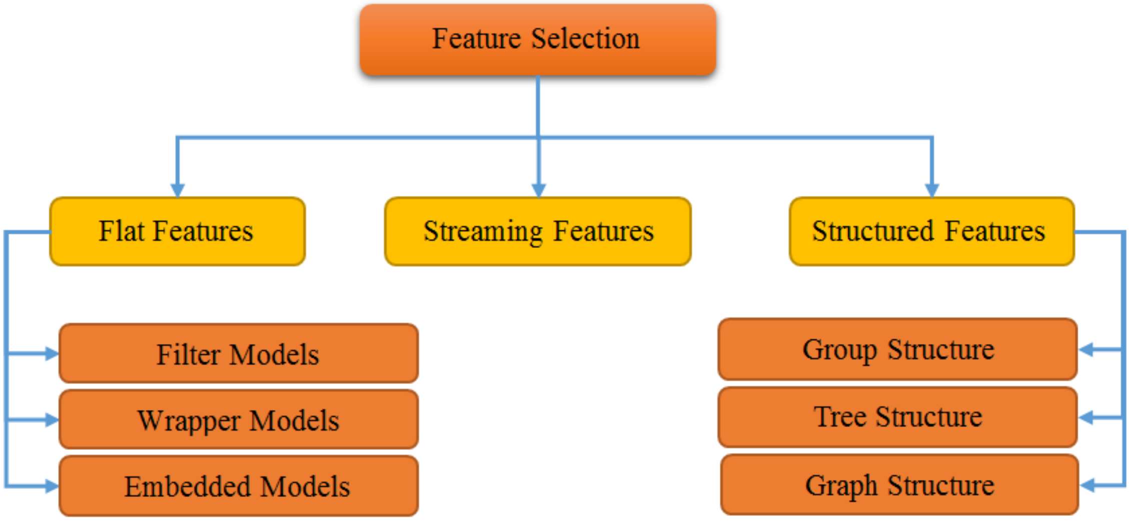


Figure 2.3 Sort the algorithms to select the selection for the label

2.6.1.1. Filter Approach

In the filer method, feature subsets are assessed using independent measures without the necessity of learning an algorithm. This approach is computationally fast and efficient. Meanwhile, the filter methods can miss the features that are redundant by themselves but are relevant when combined with other features. Figure 2.4 shows a flowchart representation of the filter model.

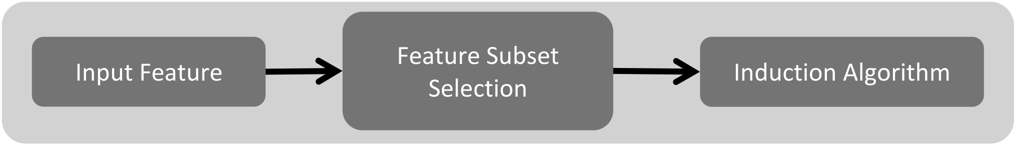


Figure 2.4 Filter Model

2.6.1.2. Wrapper Model

The filter and wrapper approaches can only be differentiated through an evaluation criterion. In the wrapper approach a learning algorithm is used to evaluate subsets. Figure 2.5 shows a flowchart representation of the wrapper model. Different wrapper algorithms can be generated by varying the subset generation (X_g) and subset evaluation measures (through the use of dependent criterion). The wrapper approach results in the selection of the best subsets which suit the learning algorithm, and generally this approach gives better performance.

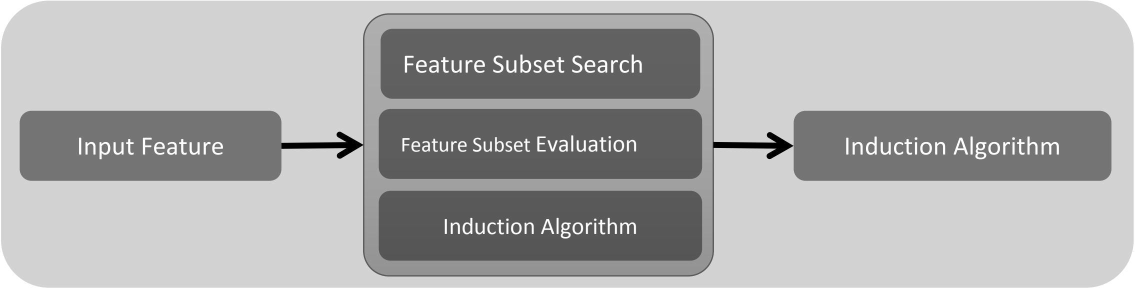


Figure 2.5 The wrapper models

2.6.1.3. Embedded Approach

Relative to the wrapper model, the embedded model interacts with the learning algorithm at a lower computation cost. This model is capable of capturing the dependency of the features, it also considers the relationship between an input feature and its output feature coupled with locally searching for features that give a better local discrimination. It utilizes the independent criteria while deciding optimal subsets for a given cardinality. The learning algorithm is then employed for the selection of the final optimal subsets across various cardinality.

2.7. Firefly Algorithm (FA)

The Firefly Algorithm (FA) is a stochastic global nature-inspired optimization method developed by Yang [16]. The FA mimics the mating and information exchange mechanism of fireflies through flashes of light. In this section the behavior of fireflies will be discussed. In addition, the Binary firefly, the artificial firefly, and the FA fore feature selection will be discussed.

2.7.1. The Behavior of Fireflies

In the world there are more than 2000 Firefly species, of which a majority radiate short and rhythmic flash patterns [16][18]. The main function of these flashes are the attracting partners for mating through communication, attracting prey, and as a warning mechanism. There are two factors which contribute to only allowing fireflies to be visible at short distances. The first factor is that the intensity of light at a distance r from the light source obeys the inverse square law. This is to say that the intensity of light I is inversely proportional to the square of the distance (i.e. $I \;\propto \frac{1}{r^2}$). The second factor is due to the absorption of light in air which reduces the intensity for an increase in distance. The preceding manners and rules are devised mathematically in the artificial firefly which will be detailed in the proceeding section.

2.7.2. Artificial Fireflies

Yang devised three idealized rules to depict the behavior of artificial fireflies and they are:

- All fireflies are unisex and can be attracted to each other regardless of the sex.

- A specific firefly's degree of attractiveness is proportional to its luminosity, meaning the brighter fireflies typically attract those with lower luminosity. It also means the attractiveness decreases with the increase of distance between the fireflies.

- The luminosity of a firefly is governed by the topography of the objective function. In a maximization problem the luminosity or brightness can be basically proportional to the objective function's value.

The degree of attractiveness of one firefly i towards another one with more intensity j is governed by:

$$X_i = \beta_0 e^{-\gamma r^2}\left(X_j - X_i\right) + \alpha\,(rand - \tfrac{1}{2}) \qquad (2.1)$$

Where *i* and *j* denote the attraction and randomization. α *denotes the* randomization parameter and *rand denotes the r*andom number selected from a uniform distribution in [0*;* 1]. Therefore, the expression (*rand* - 0*:*5) ranges from [-0.5,0.5] to permit positive and negative variations. β_0 is typically set to 1 and α ϵ [0*;* 1]. α *denotes* the environmental noise which may influence the transmission of light. In the artificial algorithm, there is the option to select this parameter to permit variation in the solution and therefore, it offers more solutions. The randomization term can be modified to account for a normal distribution with a mean of 0 and variance of 1; *N* (0*;* 1) to account for changes in the rate of noise within the environment. γ *denotes the* variation in degree of attractiveness, and the value of it is significant in the determination of the speed of convergence and also in the determination of the behavior of FA. The term γ *varies from a value of 0.01* to 100 for majority of the applications. The distance between f_i *,* f_j*; is* represented asf_{ij}*;* and it is defined in equation 2.2.

$$f_{ij} = \| \ X_i - \ X_j \ \| \qquad (2.2)$$

Where X_i = the position of firefly *i.*

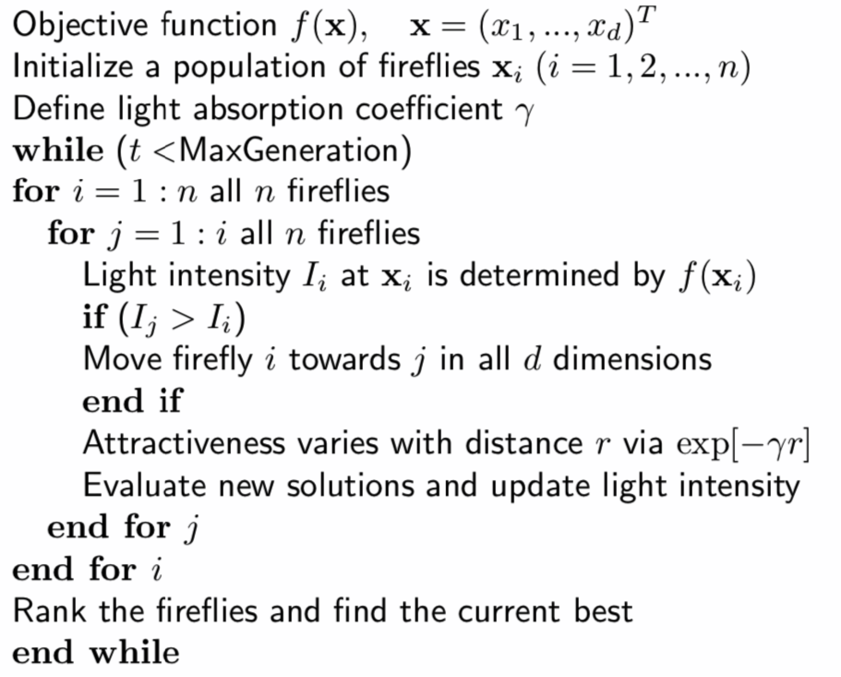
It is important to note that in the updated equation $\beta_0 e_{ij}^{-\gamma r^2}$, the attractiveness coefficient is used to approximate the loss in the light intensity due to distance as detailed in the idealized rules. This behavior can be modeled as a monotonically decreasing function. Likewise, it was mentioned that dust and the environment can also impact light intensity, this effect can be modelled using the random term in the equation. Therefore, the FA which took inspiration from the behavior of fireflies can be devised as in the proceeding pseudo-code; refer to Algorithm 2.1, flowchart in figure 2.6. The properties of the FA can be expressed as follows:

- The FA is a swarm intelligent method which raises the merits of swarm optimization.

- The FA is capable of handling multi-model problems with ease owing to its automatic subdivision of population, where an individual firefly's scope of vision is constrained to permit the formation of sub-swarms in the search space.

- The convergence speed of the algorithm can be developed by dynamically adjusting the attraction and randomness parameters of the FA during the process of iteration.

Given below, is the detail of the Fire Fly algorithm [16] and flowchart shown in Figure 2.6.



```
Objective function $f(\mathbf{x})$,    $\mathbf{x} = (x_1, ..., x_d)^T$
Initialize a population of fireflies $\mathbf{x}_i$ $(i = 1, 2, ..., n)$
Define light absorption coefficient $\gamma$
while ($t <$MaxGeneration)
for $i = 1 : n$ all $n$ fireflies
   for $j = 1 : i$ all $n$ fireflies
      Light intensity $I_i$ at $\mathbf{x}_i$ is determined by $f(\mathbf{x}_i)$
      if $(I_j > I_i)$
      Move firefly $i$ towards $j$ in all $d$ dimensions
      end if
      Attractiveness varies with distance $r$ via $\exp[-\gamma r]$
      Evaluate new solutions and update light intensity
   end for $j$
end for $i$
Rank the fireflies and find the current best
end while
```

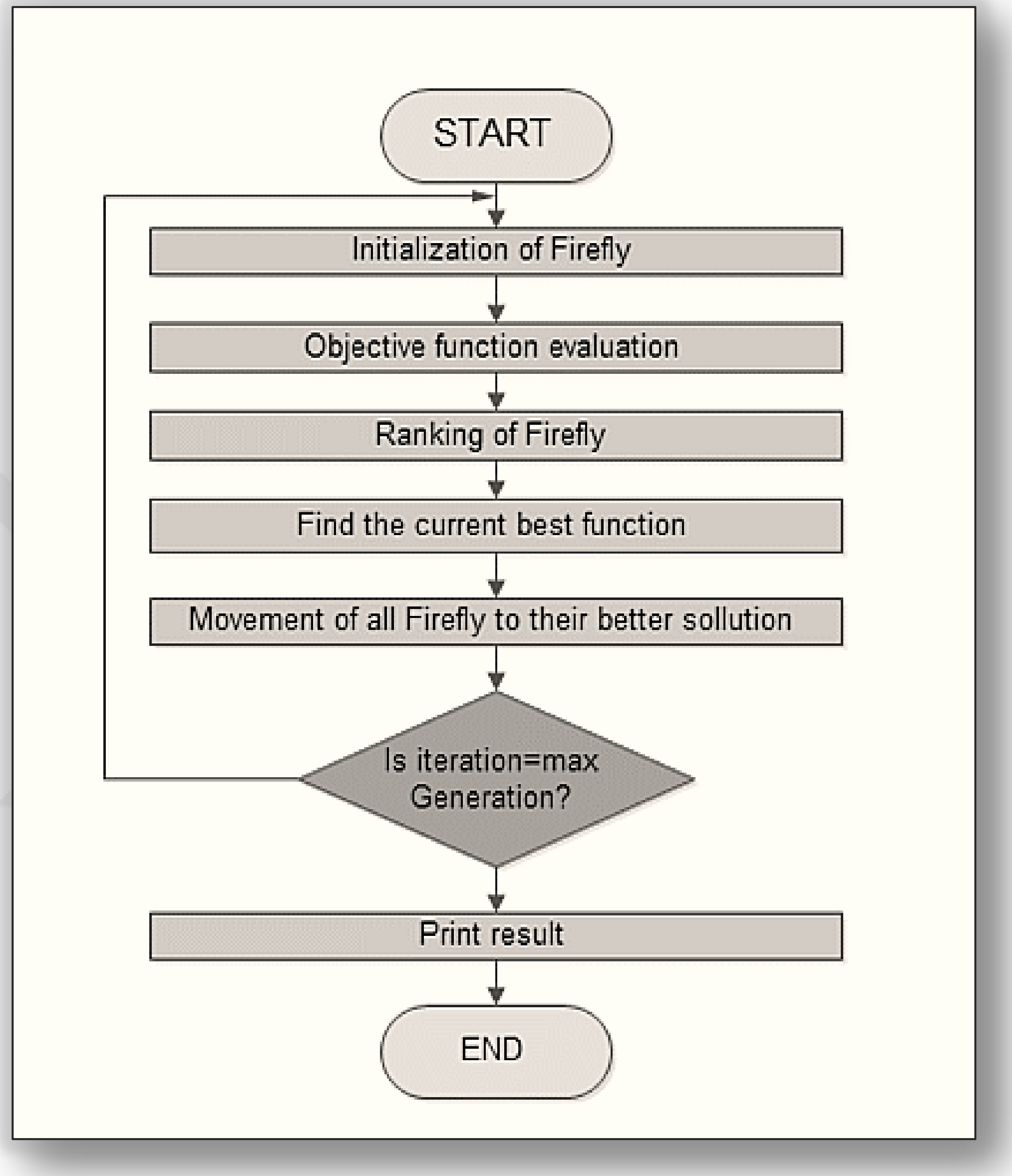
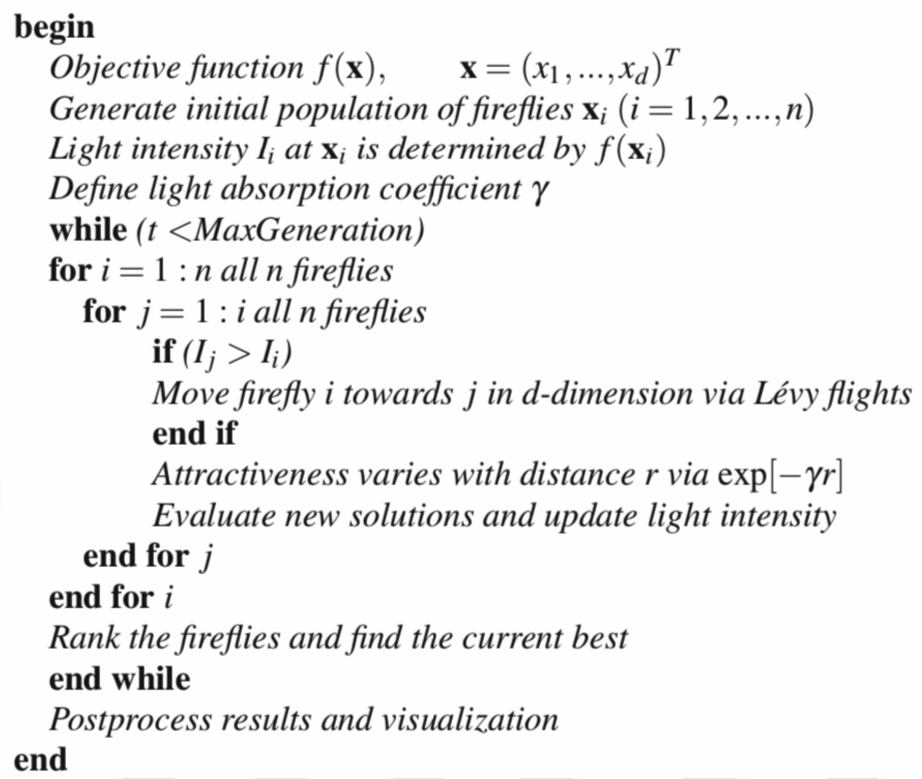

START
Initialization of Firefly
Objective function evaluation
Ranking of Firefly
Find the current best function
Movement of all Firefly to their better sollution
Is iteration=max Generation?
Print result
END

Figure 2.6 Flowchart of the Firefly algorithm

2.8. LFA (Levy flight – Firefly Algorithm)

From the same author of firefly algorithm, new metaheuristic nature-inspired algorithm was proposed in [49], combining Levy flight with search strategy of fire flight algorithm. From the three rules of Firefly (explained in [49]) and the characteristics of Levy flights, Yang proposed the following LFA pseudo code [49]:



```
begin
    Objective function $f(\mathbf{x})$,        $\mathbf{x} = (x_1, ..., x_d)^T$
    Generate initial population of fireflies $\mathbf{x}_i$ $(i = 1, 2, ..., n)$
    Light intensity $I_i$ at $\mathbf{x}_i$ is determined by $f(\mathbf{x}_i)$
    Define light absorption coefficient $\gamma$
    while (t <MaxGeneration)
    for $i = 1 : n$ all n fireflies
        for $j = 1 : i$ all n fireflies
                if $(I_j > I_i)$
                Move firefly i towards j in d-dimension via Lévy flights
                end if
                Attractiveness varies with distance r via $\exp[-\gamma r]$
                Evaluate new solutions and update light intensity
        end for j
    end for i
    Rank the fireflies and find the current best
    end while
    Postprocess results and visualization
end
```

From the formulated validation tests in [49], Yang showed the LFA is more superior in performance and accuracy (fining global optimum solution) when solving optimization problems (such NP-hard problems). They study compared LFA to particles swarm optimization algorithm and genetic algorithm.

CHAPTER THREE

DESIGN AND IMPLEMENTATION

3.1. Introduction

The main target of this thesis is to improve the performance of the spam filtering system by the selection of the most relevant feature subsets. In this chapter, the design and implementation of the suggested spam filtering system will be expanded on in detail. The proposed system is specialized on spam filtering applications. It utilizes Naïve Bayesian Classifier based on LFA. This chapter is grouped into three sub chapters. The first of which explains the proposed spam filtering system. The second sub chapter elaborates on the conversion of the data set and how it is processed. The data set is converted from Weka file format into a database format. The last sub chapter expounds on the steps of utilizing the FA wrapper feature selection method for the use in a Spam Filtering system.

3.1.1. Data Set Information

The content and purpose of spam is very diverse. These can range from product or website advertisements, get rich quick schemes, serial letters to pornographic content. For this study, the unsolicited emails were collected from the mail manager and individuals who contributed unsolicited messages. The group of unsolicited messages used in this study came from deposited business and also personal emails. Such indicators are beneficial when a custom spam filter is being developed, but for developing a general-purpose spam filter such unwanted indicators must be neglected or a significantly large set of spam must be utilized [48].

3.1.2. Attribute Information

The last column of the spam base data denotes whether the email is considered spam. Spam is denoted by (1) and non-spam messages are denoted by (0). Majority of attributes specify whether a certain character or word occurs frequently in legitimate email. Length qualities (55-57) measure the distance between two successive capital letters. Factual measures of each characteristic are presented at end of this record. The associated meanings of the attributes are as follows:

1 - 48 persistent genuine [0,100] qualities of sort word_freq_WORD are equal to the level of words appearing in email that match WORD, which can be formulated as, 100 * (occurences of WORD in the email)/Total number of words in email

2 – 'Word' in this case is defined as any string of alphanumeric characters constrained by non-alphanumeric numerals or the end of string.

3 - Six nonstop genuine [0,100] qualities of sort char_freq_CHAR are equal to the level of characters occurring in email that match CHAR, which can be formulated as:. 100 * ((occurrences of 'CHAR' in email)/total number of characters in email).

4 - 1 ceaseless genuine [1,...] quality of sort capital_run_length_average is equal to the regular length of continuous successions of capitalized letters. normal length of continuous successions of capital letters.

5 - 1 nonstop whole number [1,...] property of sort capital_run_length_longest is equal to the length of the longest continuous and successive occurrence of capitalized letters.

6 - 1 ceaseless whole number [1,...] property of sort capital_run_length_total is equal to the cumulative length of continuous progression of capital letters, i.e. = total number of capitalized letters in the email.

7 - 1 ostensible {0,1} class trait of sort spam indicates whether or not the email is classified as spam (1) or not (0), for instance, an unprompted work email.

3.2. The Proposed Spam Filtering System

Figure (3.1) and figure (3.2) show the NB classifier with two configurations, the first configuration is the NB classifier with all features included, and the second configuration is the NB classifier with the wrapper feature selection method (LFA-NBC) proposed in this study. The system can be generalized as follows for both configurations:

1. Preparation and pre-processing of the dataset that will be used for training and classification.

2. In both the configurations, the NB classifier is the algorithm used for spam filtering and it is used to train and classify the dataset into spam and non-spam categories.

a. The first configuration includes all the features present in the ‘spambase’ dataset with the NB classifier as shown in (3.1).

b. The second configuration utilizes the LFA-NBC algorithm with NB classifier and is implemented as a fitness function on all features as shown in Figure (3.2).

It is vital to note that the LFA is not employed inside the NBC, rather, the LFA receives all the features after the completion of the training stage then it selects the relevant features and sends it to the testing process to evaluate them. The training stage is conducted in two steps: Frequency and Probability calculation. Both steps are executed only once concluding the training stage. This stage is followed by the second stage which is testing. Moreover, it is necessary that preprocessing stage must be completed before the training stage is started. The following section will detail the preprocessing stage.

3.2.1. Converting WEKA file format into Access DB

As aforementioned, the ‘spambase’ dataset is the main file utilized in this study. The dataset was obtained from the UCI website. [48]. The file as downloaded from the website is in WEKA file format with ‘*.arff’ extension. For the data to be accessible to our designed software, the file was converted into a MS Access Database format. Figure 3.3 presents a snippet of the original downloaded file.

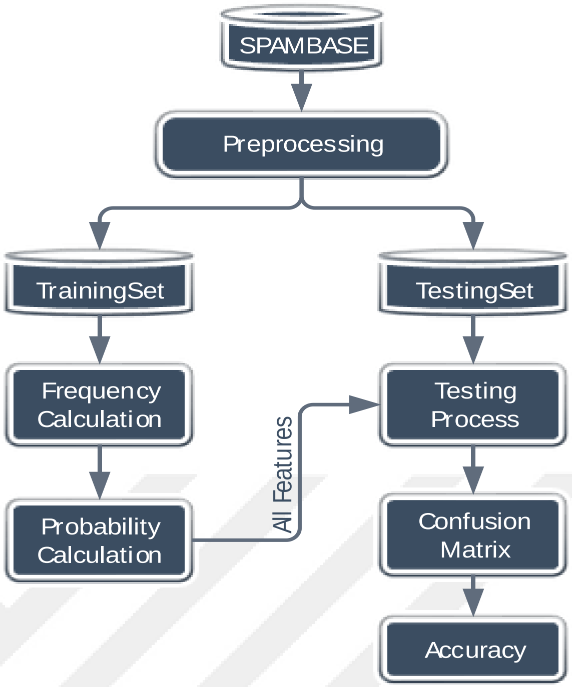


Figure 3.1 Block Diagram of NBC

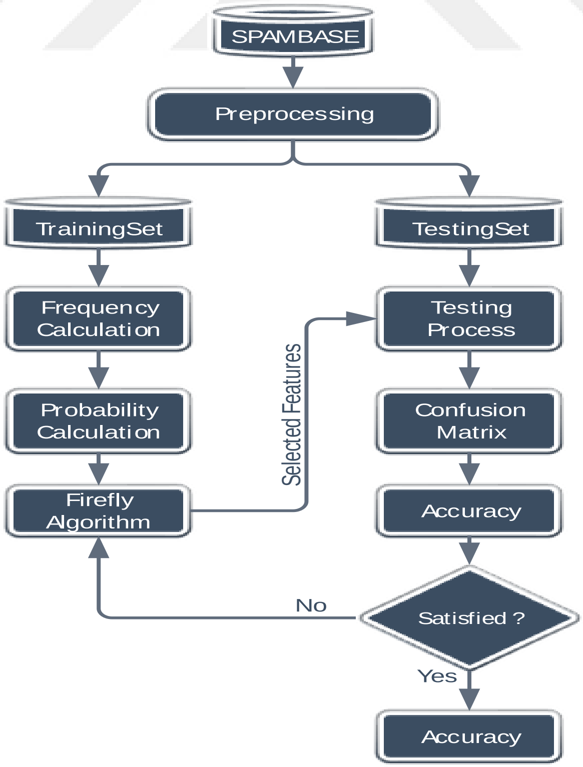
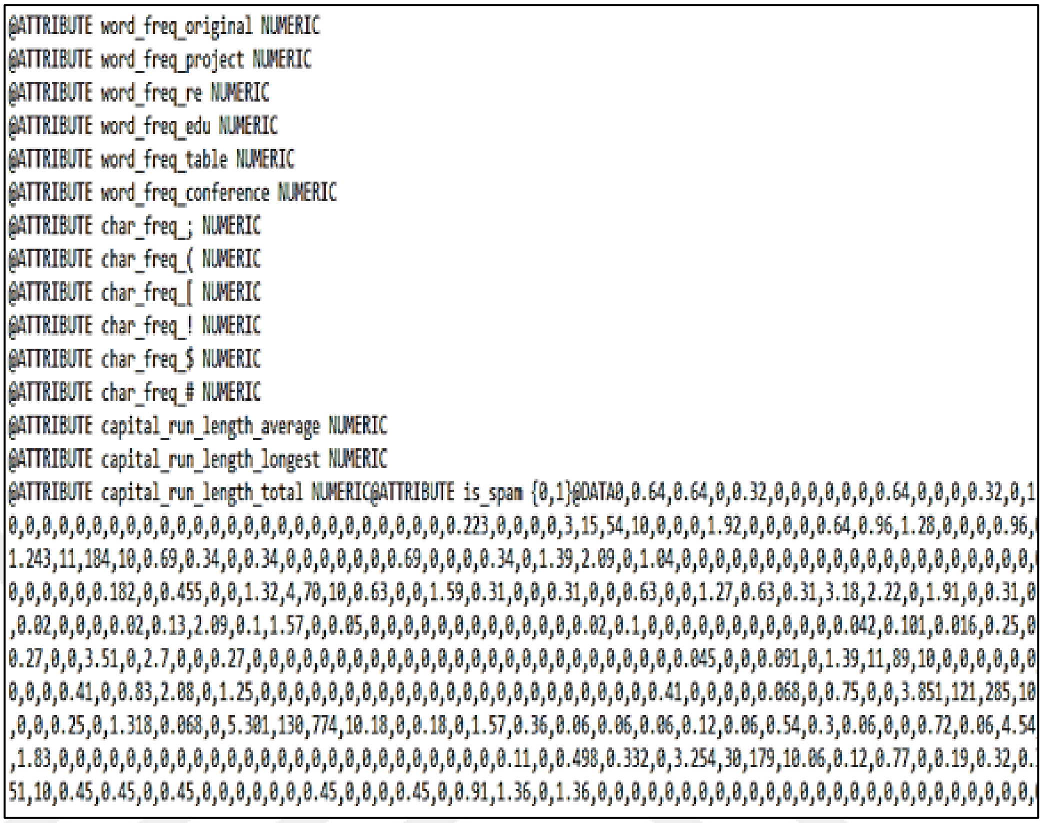


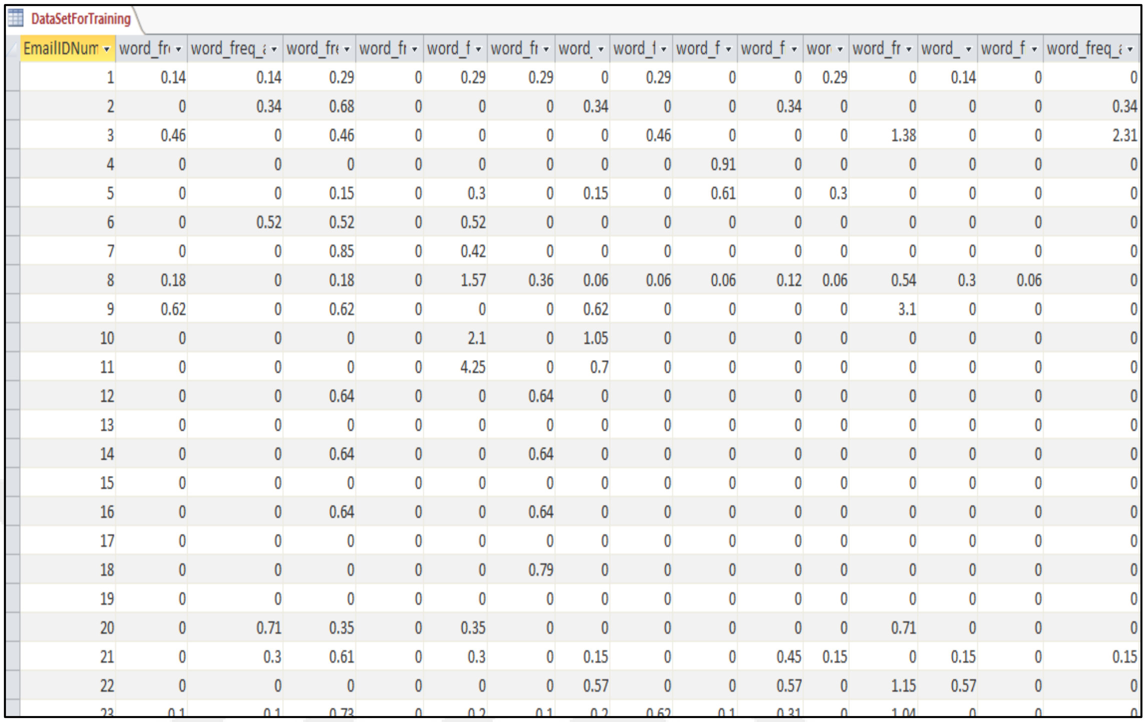
Figure 3.2 Block diagram of LFA-NBC



```
@ATTRIBUTE word_freq_original NUMERIC
@ATTRIBUTE word_freq_project NUMERIC
@ATTRIBUTE word_freq_re NUMERIC
@ATTRIBUTE word_freq_edu NUMERIC
@ATTRIBUTE word_freq_table NUMERIC
@ATTRIBUTE word_freq_conference NUMERIC
@ATTRIBUTE char_freq_; NUMERIC
@ATTRIBUTE char_freq_( NUMERIC
@ATTRIBUTE char_freq_[ NUMERIC
@ATTRIBUTE char_freq_! NUMERIC
@ATTRIBUTE char_freq_$ NUMERIC
@ATTRIBUTE char_freq_# NUMERIC
@ATTRIBUTE capital_run_length_average NUMERIC
@ATTRIBUTE capital_run_length_longest NUMERIC
@ATTRIBUTE capital_run_length_total NUMERIC@ATTRIBUTE is_spam {0,1}@DATA0,0.64,0.64,0,0.32,0,0,0,0,0,0,0.64,0,0,0,0.32,0,1
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.223,0,0,0,0,3,15,54,10,0,0,0,1.92,0,0,0,0,0.64,0.96,1.28,0,0,0,0.96,0
1.243,11,184,10,0.69,0.34,0,0.34,0,0,0,0,0,0,0.69,0,0,0,0.34,0,1.39,2.09,0,1.04,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0.182,0,0.455,0,0,1.32,4,70,10,0.63,0,0,1.59,0.31,0,0,0.31,0,0,0.63,0,0,1.27,0.63,0.31,3.18,2.22,0,1.91,0,0.31,0
,0.02,0,0,0,0.02,0.13,2.09,0.1,1.57,0,0.05,0,0,0,0,0,0,0,0,0,0,0.02,0.1,0,0,0,0,0,0,0,0,0,0,0,0,0.042,0.101,0.016,0.25,0
0.27,0,0,3.51,0,2.7,0,0,0.27,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.045,0,0,0.091,0,1.39,11,89,10,0,0,0,0,0,0
0,0,0,0.41,0,0.83,2.08,0,1.25,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.41,0,0,0,0,0.068,0,0.75,0,0,3.851,121,285,10
,0,0,0.25,0,1.318,0.068,0,5.301,130,774,10.18,0,0.18,0,1.57,0.36,0.06,0.06,0.06,0.12,0.06,0.54,0.3,0.06,0,0,0.72,0.06,4.54
,1.83,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.11,0,0.498,0.332,0,3.254,30,179,10.06,0.12,0.77,0,0.19,0.32,0.1
51,10,0.45,0.45,0,0.45,0,0,0,0,0,0,0.45,0,0,0,0.45,0,0.91,1.36,0,1.36,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Figure 3.3 Sample of SPAMBASE.arff file

The original file (arff file) is comprised of three segments, the first of which holds the dataset information such as title. The second segment holds the feature related information such as feature name and the type of data. The third and final segment holds the data samples or instances. For the purposes of this study, the first two segments were removed from the file and the third segment was imported into a MS Access database. Figure 3.4 presents a snippet of data after the completion of the conversion process.



DataSetForTraining

EmailIDNum	word_fr	word_freq_a	word_fre	word_fr	word_f	word_fr	word_	word_f	word_f	word_f	wor	word_fr	word_	word_f	word_freq_a
1	0.14	0.14	0.29	0	0.29	0.29	0	0.29	0	0	0.29	0	0.14	0	0
2	0	0.34	0.68	0	0	0	0.34	0	0	0.34	0	0	0	0	0.34
3	0.46	0	0.46	0	0	0	0	0.46	0	0	0	1.38	0	0	2.31
4	0	0	0	0	0	0	0	0	0.91	0	0	0	0	0	0
5	0	0	0.15	0	0.3	0	0.15	0	0.61	0	0.3	0	0	0	0
6	0	0.52	0.52	0	0.52	0	0	0	0	0	0	0	0	0	0
7	0	0	0.85	0	0.42	0	0	0	0	0	0	0	0	0	0
8	0.18	0	0.18	0	1.57	0.36	0.06	0.06	0.06	0.12	0.06	0.54	0.3	0.06	0
9	0.62	0	0.62	0	0	0	0.62	0	0	0	0	3.1	0	0	0
10	0	0	0	0	2.1	0	1.05	0	0	0	0	0	0	0	0
11	0	0	0	0	4.25	0	0.7	0	0	0	0	0	0	0	0
12	0	0	0.64	0	0	0.64	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0.64	0	0	0.64	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0.64	0	0	0.64	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0.79	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0.71	0.35	0	0.35	0	0	0	0	0	0	0.71	0	0	0
21	0	0.3	0.61	0	0.3	0	0.15	0	0	0.45	0.15	0	0.15	0	0.15
22	0	0	0	0	0	0	0.57	0	0	0.57	0	1.15	0.57	0	0
23	0.1	0.1	0.73	0	0.2	0.1	0.2	0.62	0.1	0.31	0	1.04	0	0	0

Figure 3.4 Sample of data in MS Access

3.2.2. Data Normalization

The magnitude of the feature values present in ‘spambase’ dataset has widely different ranges. This weakens the classification performance, as such a normalization process was implemented on the dataset to scale the data to a uniform range between [0,1]. The normalization was carried out by the use of the following expression: =

$$f_i^{new} = \frac{f_i^{old} - \min(F)}{\max(F) - \min(F)}, \qquad (3.1)$$

Where f_i signifies the current value of feature, the terms $max(F)$ and $min(F)$ signify the maximum and minimum values of the feature respectively. After the normalization is completed, the normalized dataset is split into two datasets: namely the TraningSet and the TestingSet. The TrainingSet is comprised of 3220 samples, whereas the TestingSet is comprised of 1380 samples. Simply, the TrainingSet consists of the majority (70%) of the ‘spambase’ dataset whereas the rest (30%) comprises the TestingSet.

3.3. Proposed Algorithm

The primary driver of building a feature selection algorithm is to identify better subset features for an improved performance accuracy. Traditionally, in wrapper models like the LFA, the fireflies are all initialized with randomly chosen features. In contrast, the proposed model, the fireflies in the swarm is initialized in a binary sequence. The proposed algorithm follows these major steps:

3.3.1. Initialization

In this step, all fireflies in the swarm are initialized with a random number between the range [0,1]. These numbers signify the position of each firefly and is computed using the expression in Eq. (3.2).

$$X = (UB - LB) \times Rand(0,1) + LB \tag{3.2}$$

In this expression, UB and LB denotes the upper bound (1.0) and lower bound (0.0), respectively. The function Rand() denotes a logistic chaotic map given in Eq (3.3). It helps firefly algorithm to start from randomized positions rather than by a uniform distribution. The sequence generated from Eq (3.2) is then converted into a binary sequence by utilizing the sigmoid function as in Eq (3.3).

$$X_{i+1} = \mu X_i \, (1 - X_i) \tag{3.3}$$

Here $X_i \; and \; X_{i+1}$ denote the initial value and the following value respectively, whereas μ denotes ‘mutation’ which is the control parameter.

$$B_i = \begin{cases} 1, \; sigmoid\,(X_i) > \; U\,(0\,,1) \\ 0, \; otherwise \end{cases} \tag{3.4}$$

Here X_i represents the position of a firefly, the sigmoid (X_i) is calculated as 1 / (1 + e $^{-\mathrm{GRI}}$), and U denotes the uniform distribution. *Bi* term denotes the binary sequence, value of 1 means the feature will be selected, and conversely a value of 0 means the feature will not be selected.

3.3.2. Fitness Function

The fitness function of the proposed algorithm to obtain minimum error rate in classification performance over the validation set of the training data yet at the same time maximizing the number of non-selected features (i.e. irrelevant or unnecessary features). The error is computed using the expression in Eq (3.5). A classifier must be utilized to calculate the fitness function. Here, Naïve Bayesian Classifier was employed to obtain the accuracy.

$$\boldsymbol{Error} = \ 100 - A \tag{3.5}$$

Here *A* denotes the classifier accuracy rate; i.e. the 5-fold cross validation error rate obtained following the training of the Naïve Bayesian Classifier. The value of the error is utilized to calculate the intensity of each firefly using Eq (3.6).

$$I\ (F_i) = \frac{1}{1 + Error^2} \tag{3.6}$$

3.3.3. Attractiveness Calculation

The attractiveness β of each firefly is expressed using Eq (3.7).

$$\beta(r) = \ \beta_0 \ \times \ e^{-\gamma r^2} \tag{3.7}$$

Here r denotes the distance between two flies is expressed using Eq (3.8), and β_0 denotes the attractiveness at r = 0 (Initial Attractiveness).

$$r_{ij} \ = \left|X_i - \ X_j\right| \tag{3.8}$$

Here X denotes the real values of the position of the fireflies computed using the information gain ratio equation. The hamming distance method is employed to calculate the distance by subtracting each bit of firefly i from firefly j. In this method the distance is denoted by the difference between the binary strings of the two fireflies. Utilizing this method will enhance the algorithm for operating with the binary sequence (features) rather than operating with continuous values (positions) [16].

3.3.4. Position Updating

Fireflies (Fi) in the swarm are attracted towards brighter fireflies, this is to say that each firefly in the swarm moves toward the brighter firefly. This is called position updating and is determined by using the expression in Eq (3.9) [16].

$$X_i = X_i + \beta \times \left(X_j - X_i \right) + \alpha \times (Rand - \tfrac{1}{2}) \qquad (3.9)$$

Here X_i in the first term of the equation denotes the immediate position, and the second term accounts for the attractiveness between the position of F_i and F_j. The third term expresses the randomization with α, where $\alpha \in [0,1]$. The randomness parameter α is reduced discretely by another constant rate δ, where $\delta \in [0.95 , 0.97]$, such that at the last stage of the optimization process, α has its lowest value as in Eq (3.10).

$$\alpha = \alpha \times \delta \qquad (3.10)$$

The pseudo-code of the proposed algorithm is given in Figure (3.5) shows the pseudo-code of the proposed algorithm.

Algorithm: LFA-NBC

```
Initialize the swarm by using chaotic logistic map
Convert the swarm to binary
Define the objective function f(x) = The classification accuracy by NBC
While (itr < MaxItr)
    Calculate Attractiveness
    Update Position
    Convert the position to binary
    Evaluate the solutions using NBC
    Rank the swarm and get the best LFA
End While
Print Best LFA
```

Figure 3.5 The pseudo-code of the proposed algorithm.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1. Introduction

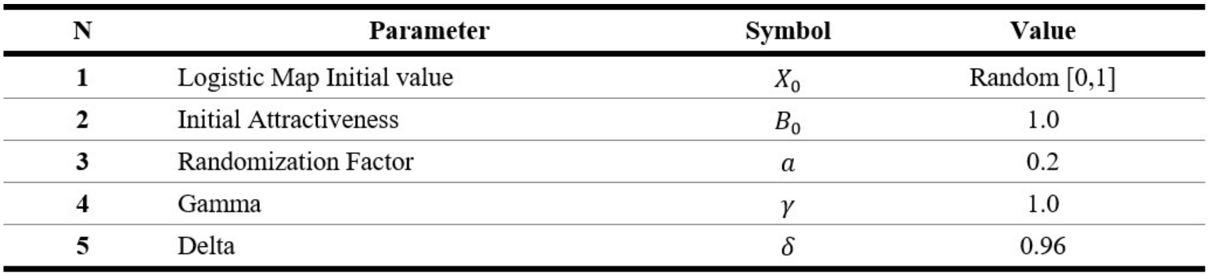
The main contribution of this thesis is the design of feature selection algorithm wrapper based on chaotic levy-firefly algorithm (LFA). This chapter presents detail results and discussion of the proposed system through several testing scenarios. The chapter is divided into two main parts; first part illustrates the experimental setup, while the second part presents the attained results.

4.2. Experimental Results

The proposed chaotic LFA algorithm and Naïve Bayesian classifier have been developed using visual C#.net version 6.0 – Visual Studio 2017 community version. The developed program has been implemented in an environment with the following specification: Operating System is Windows 10 with 64-bit architecture, CPU Intel 2.4 GHz, and RAM 8GB.

The dataset used in this thesis is SPAMBASE dataset, obtained from [1]. The original version of SPAMBASE consists of two classes (Spam and Non-Spam), contains 57 features (detailed in Appendix 1), and 4601 samples. In order to execute the Chaotic LFA, several parameters are required to be initialized. Table 4.1 lists firefly parameters (for the firefly algorithm) with their values according to [2].

Table 4.1 Firefly Algorithm Parameter Settings



N	Parameter	Symbol	Value
1	Logistic Map Initial value	X_0	Random [0,1]
2	Initial Attractiveness	B_0	1.0
3	Randomization Factor	a	0.2
4	Gamma	γ	1.0
5	Delta	δ	0.96

Optimization in general means trying different settings/values of a set of input variables to a given problem. In that, a finite search is performed where several solutions – hundreds or thousands - are evaluated until a particular solution is equal/approximately equal to fitness function. By re-examining the block diagram in Figure 3.2, the proposed approach in this thesis is comprised of LFA and NB; the set of features selected by LFA is used to derive new instances of data from the original dataset and then fed to NB. If NB classification accuracy increased, the selected set of features are kept and further optimized, otherwise, new set is selected.

On the optimization side, LFA algorithm swarm size define the number of flies, similar to population size which defines the number of parents in Genetic Algorithm. The number of flies represent the number solutions to be evaluated.

Therefore, two factors were considered to test the proposed algorithm:

1- **Swarm size**: several swarm sizes are chosen to test the relation of swarm size on NB accuracy. The following five sizes are considered: 10, 20, 30, 40 and 50.

2- **Number of Iterations**: according to Figure 3.2, one iteration represents computing chaotic sequence, selecting set of features and construct new data instance, testing NB classification accuracy, such that it can be repeated up to ***N*** times until no further improvement can be realized. For the reason that there is no defined method in the literatures to calculate the optimum value of ***N***, five scenarios of 100, 200, 300, 400, and 500 iteration will be considered in our test.

Last, the firefly algorithm is initialized randomly through generation of chaotic sequence, it is expected that each runtime shall produce results of different accuracy. Therefore, all proposed scenarios in this thesis have been implemented with 10 run times, then the maximum, minimum, and the average accuracy are measured.

4.3. Results

The below figures compare in details the proposed LFA-NB accuracy with NB accuracy, and the number of selected features for each run. Precision and Recall measures are available in Appendix B.

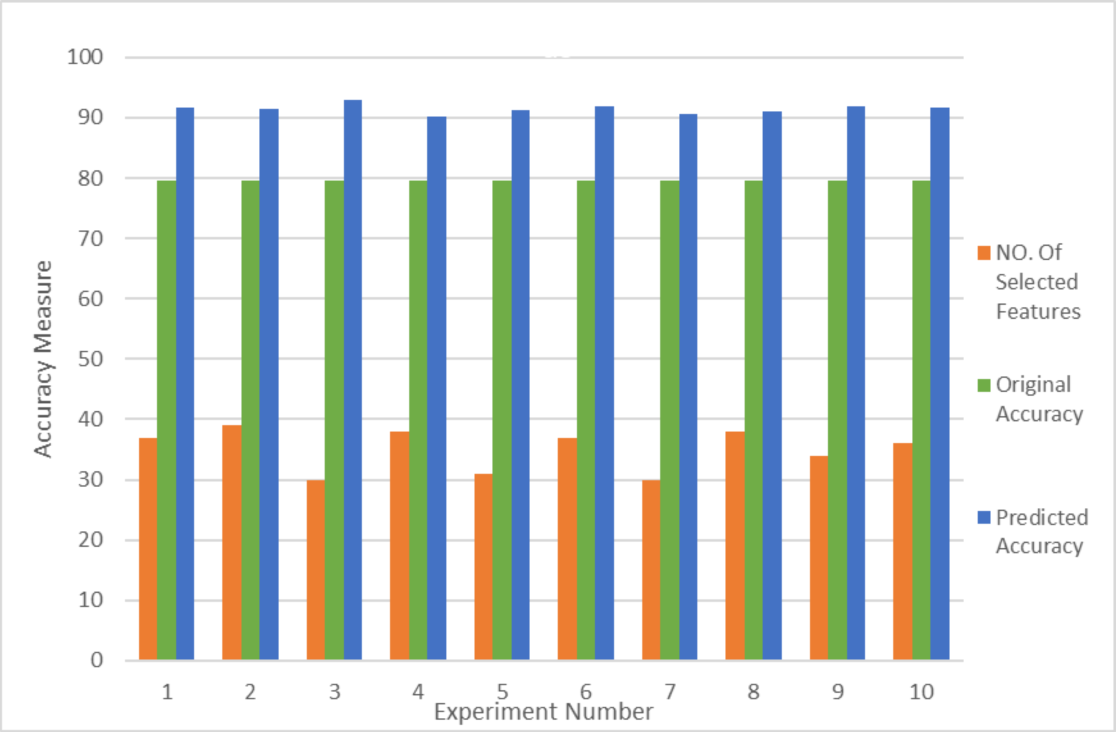


Figure 4.1 Accuracy and Selected Features for 100 iterations, swarm size=10.

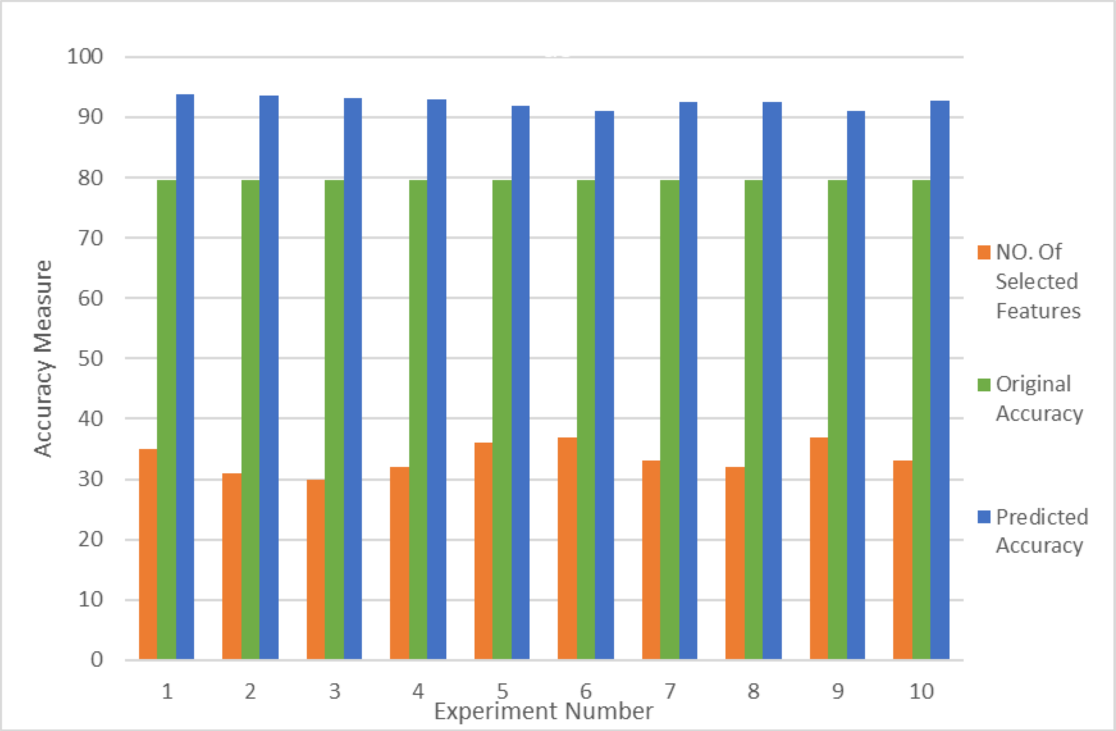


Figure 4.2 Accuracy and Selected Features for 100 iterations, swarm size=20.

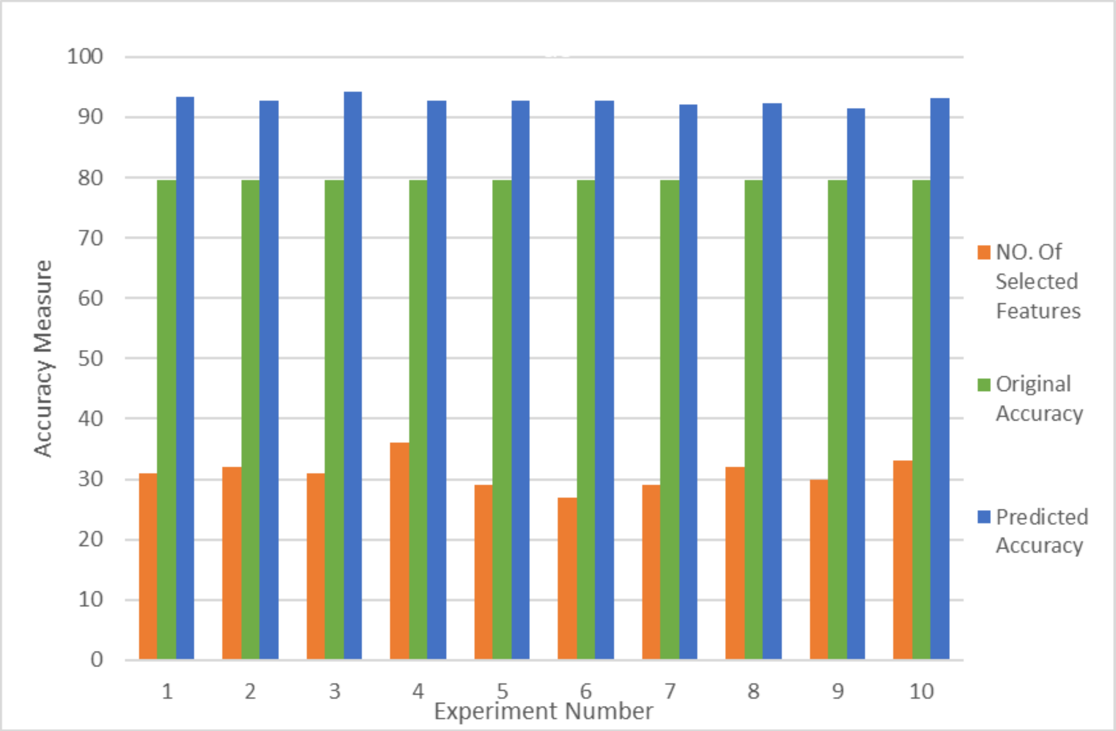


Figure 4.3 Accuracy and Selected Features for 100 iterations, swarm size=30.

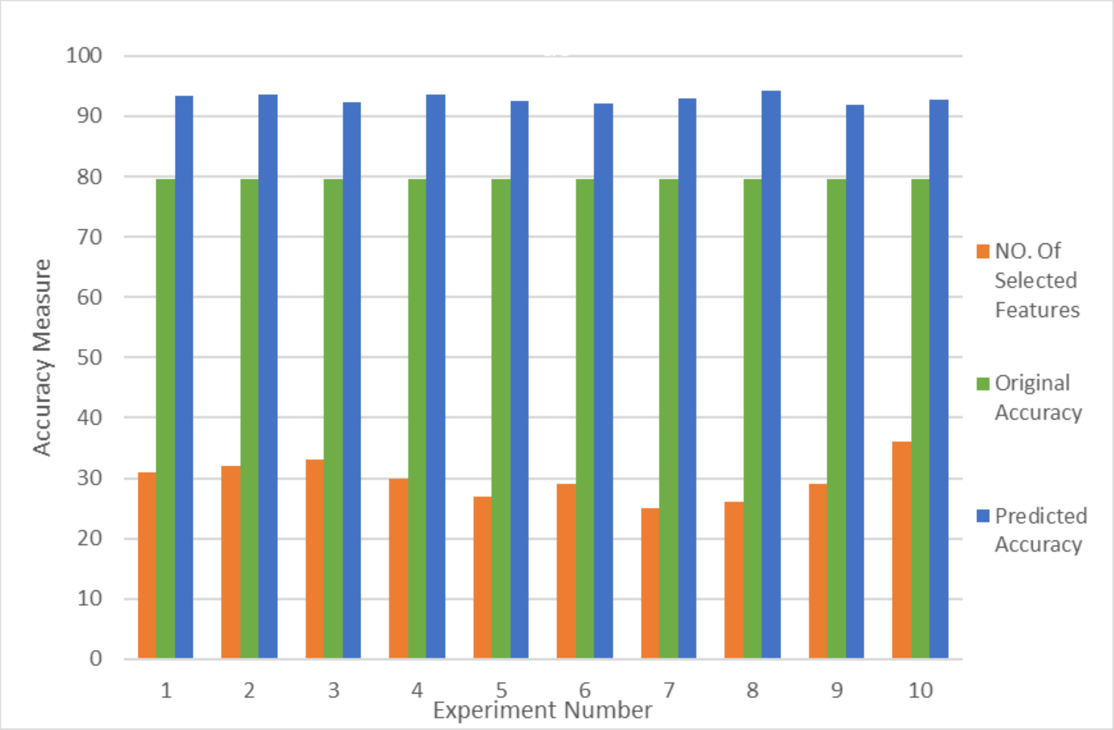


Figure 4.4 Accuracy and Selected Features for 100 iterations, swarm size=40.

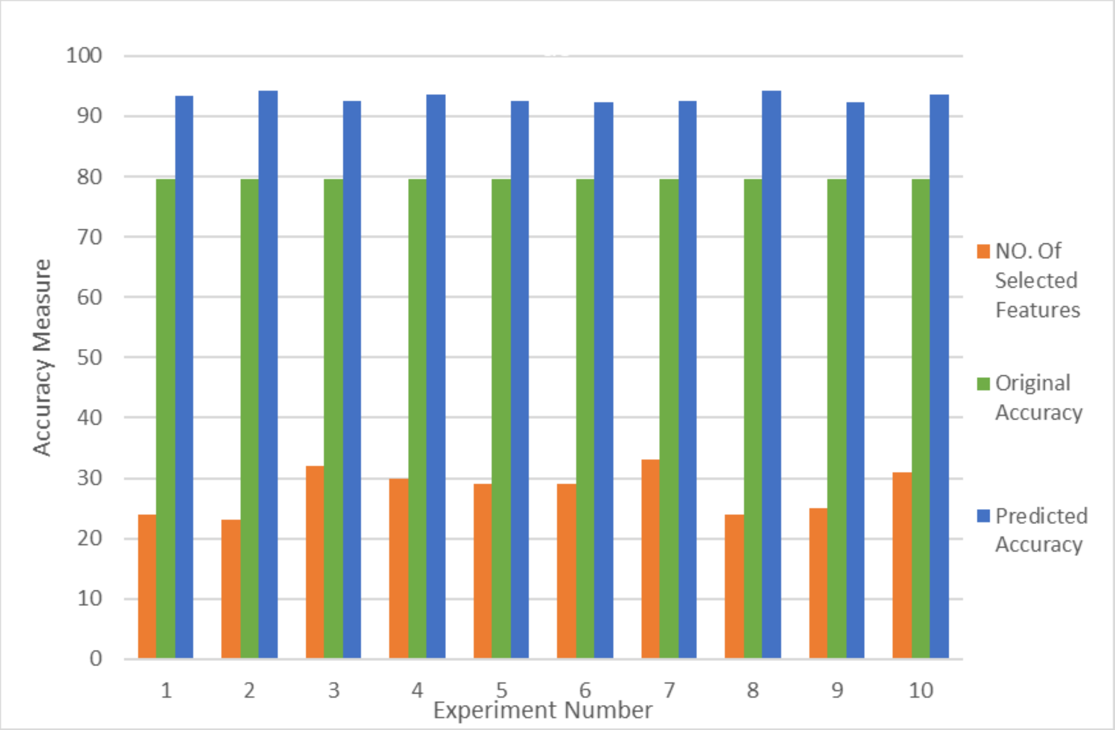


Figure 4.5 Accuracy and Selected Features for 100 iterations, swarm size=50.

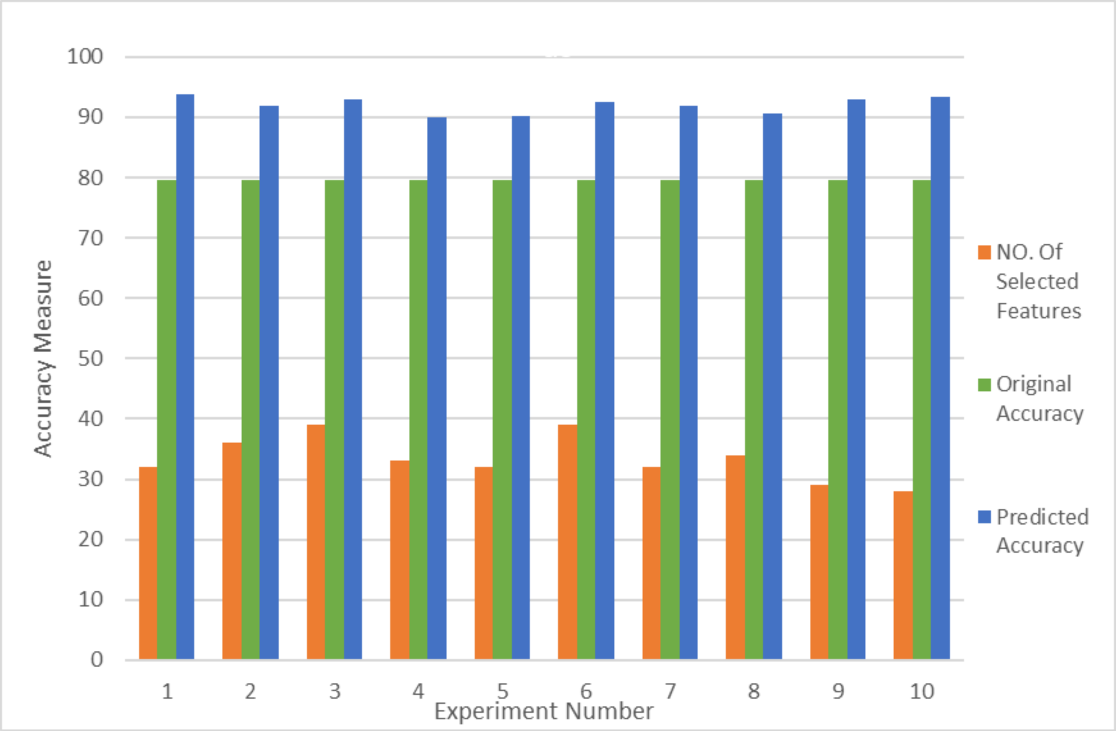


Figure 4.6 Accuracy and Selected Features for 200 iterations, swarm size=10.

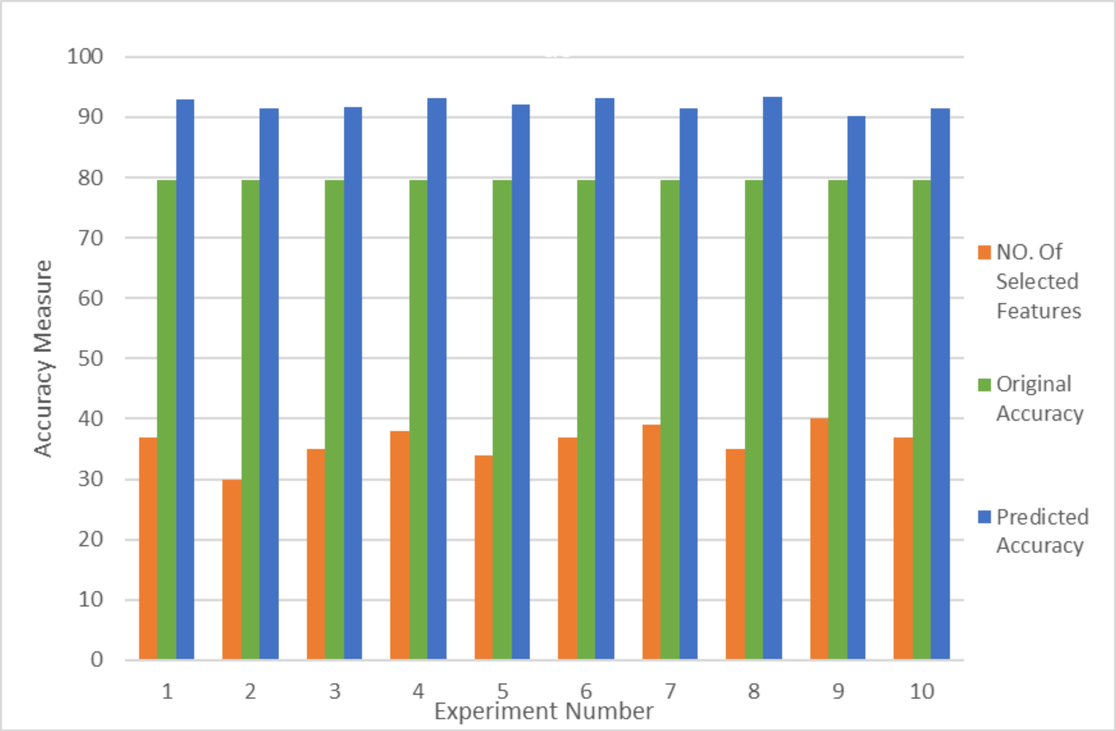


Figure 4.7 Accuracy and Selected Features for 200 iterations, swarm size=20.

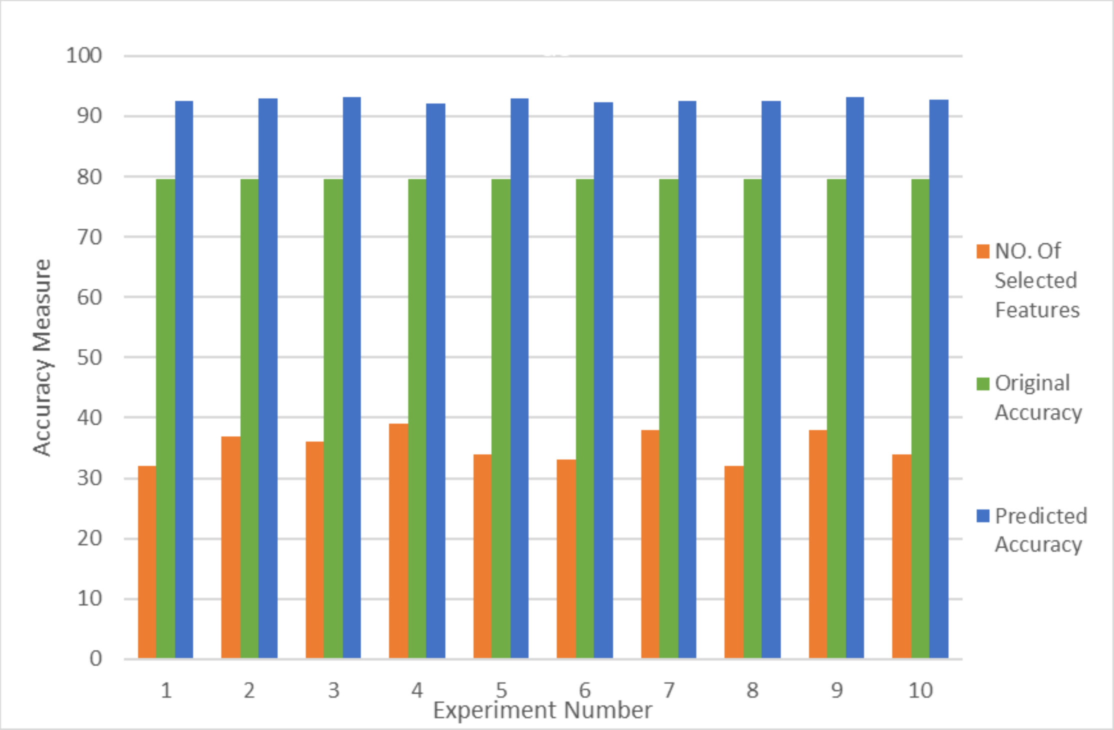


Figure 4.8 Accuracy and Selected Features for 200 iterations, swarm size=30.

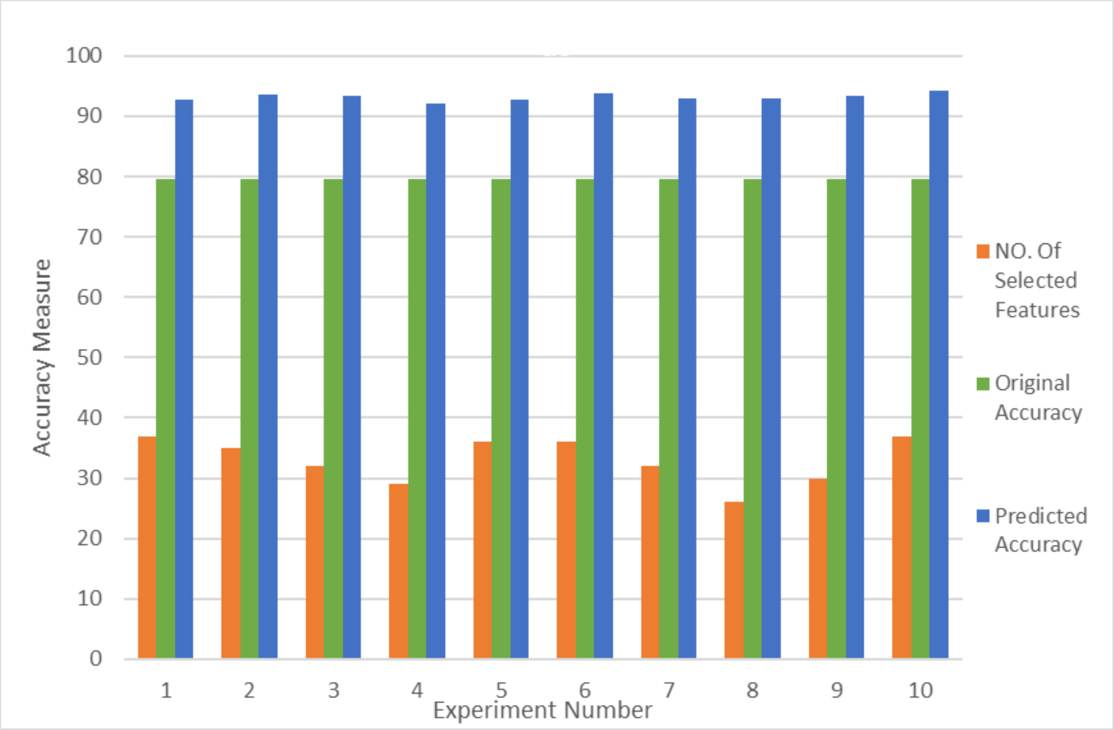


Figure 4.9 Accuracy and Selected Features for 200 iterations, swarm size=40.

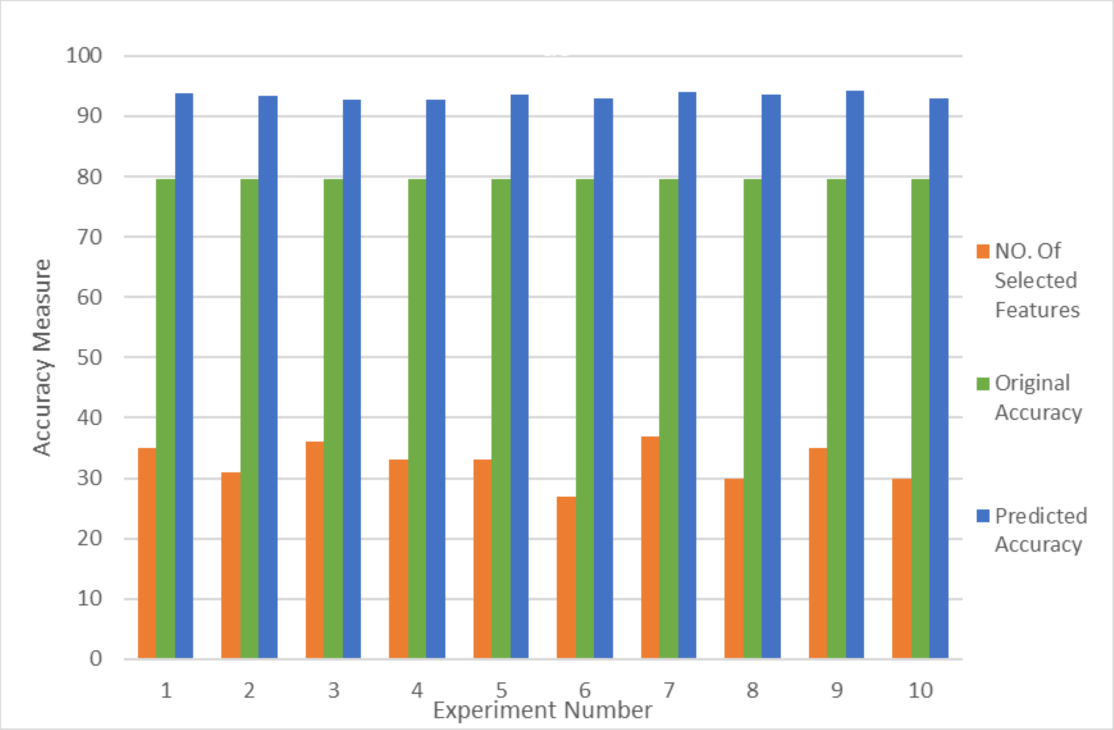


Figure 4.10 Accuracy and Selected Features for 200 iterations, swarm size=50.

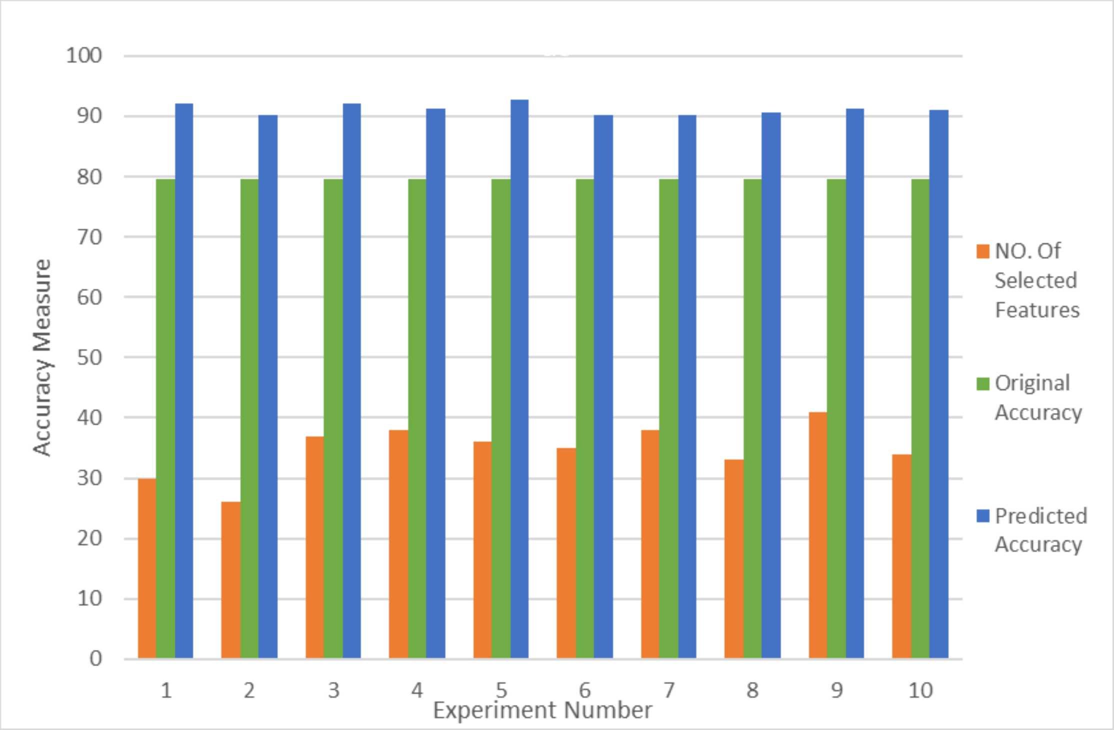


Figure 4.11 Accuracy and Selected Features for 300 iterations, swarm size=10.

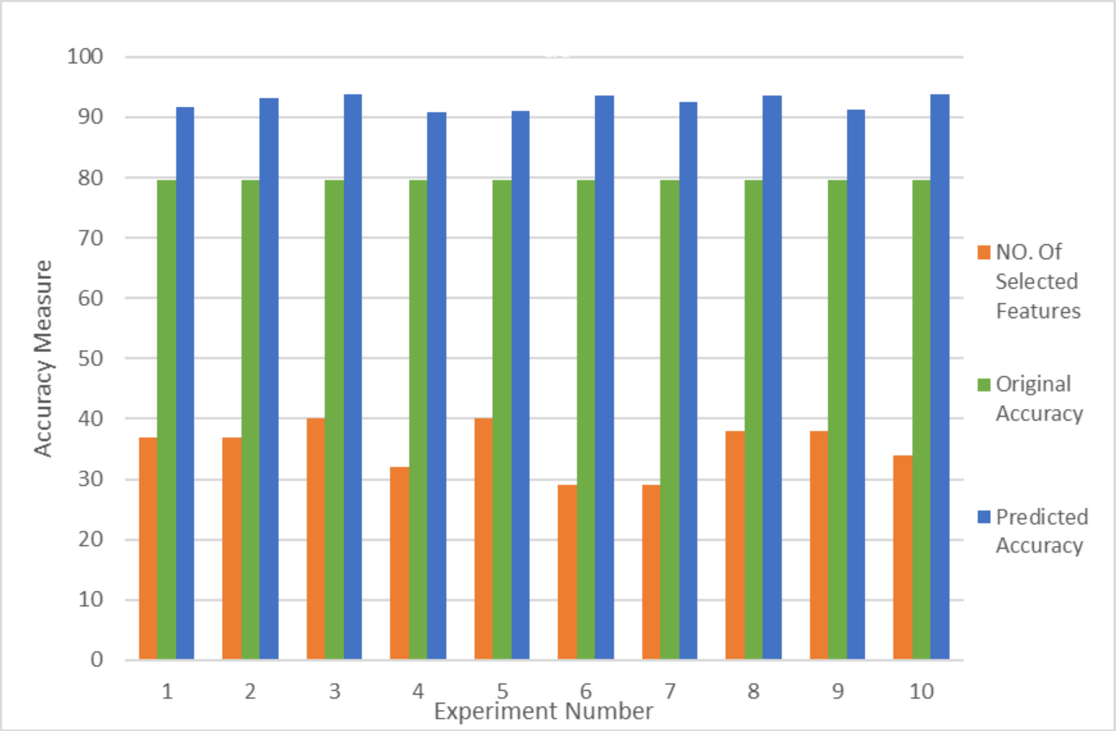


Figure 4.12 Accuracy and Selected Features for 300 iterations, swarm size=20.

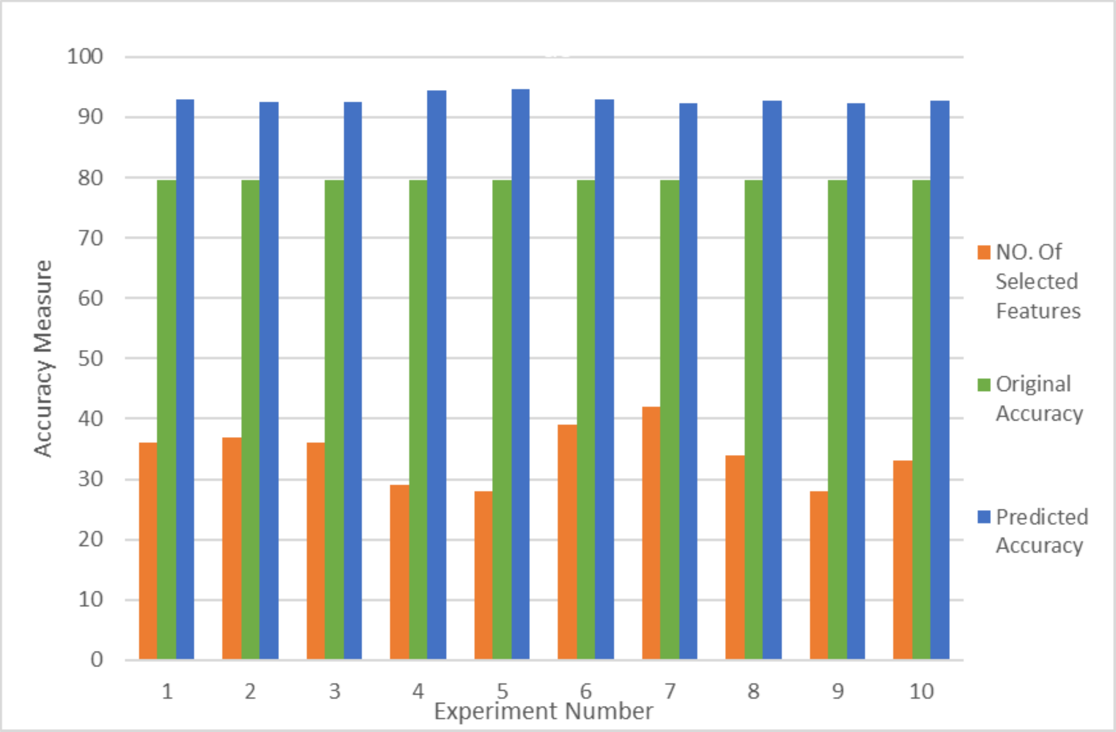


Figure 4.13 Accuracy and Selected Features for 300 iterations, swarm size=30.

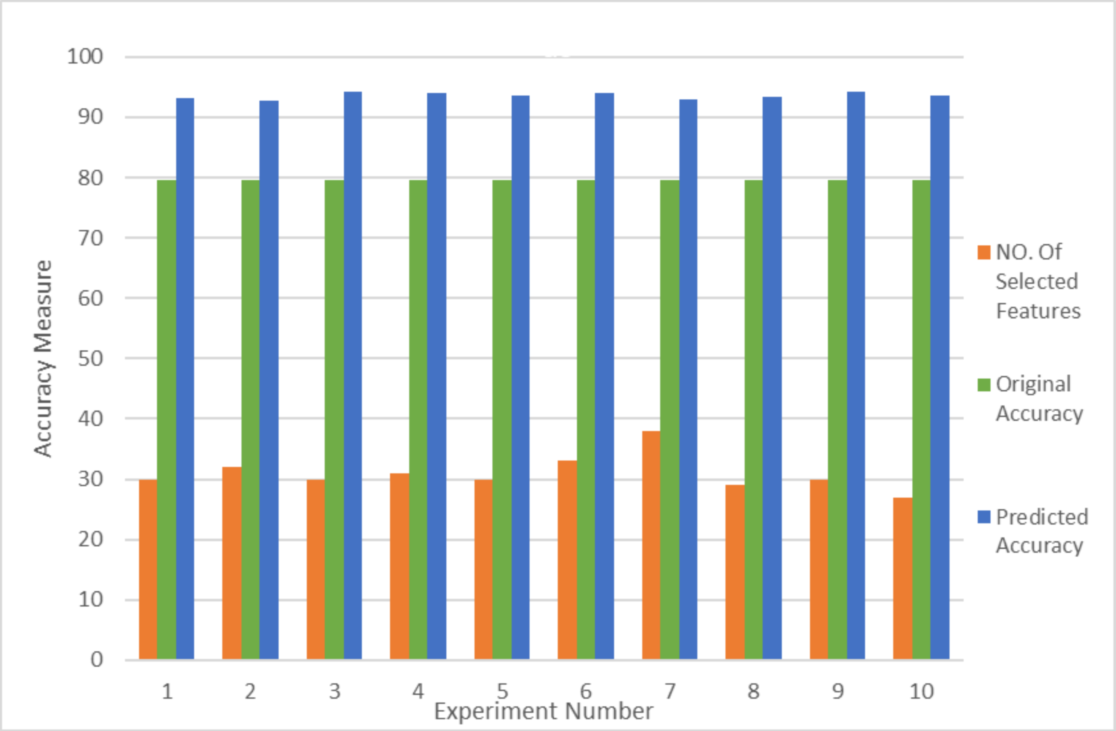


Figure 4.14 Accuracy and Selected Features for 300 iterations, swarm size=40.

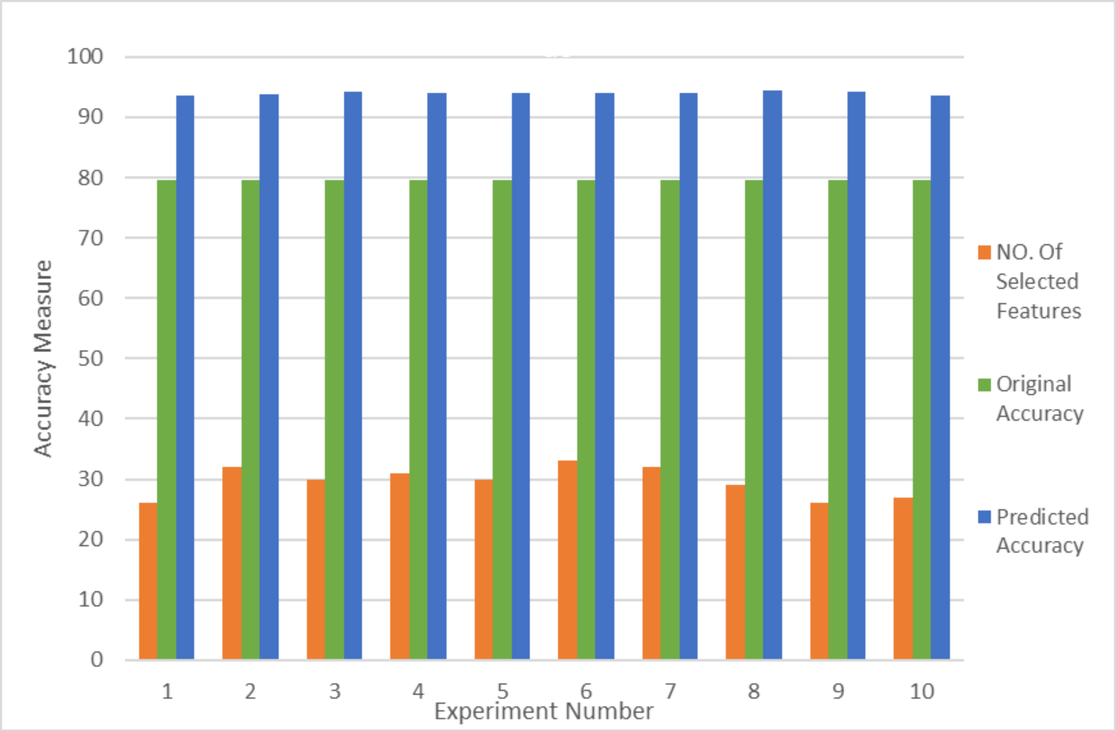


Figure 4.15 Accuracy and Selected Features for 300 iterations, swarm size=50.

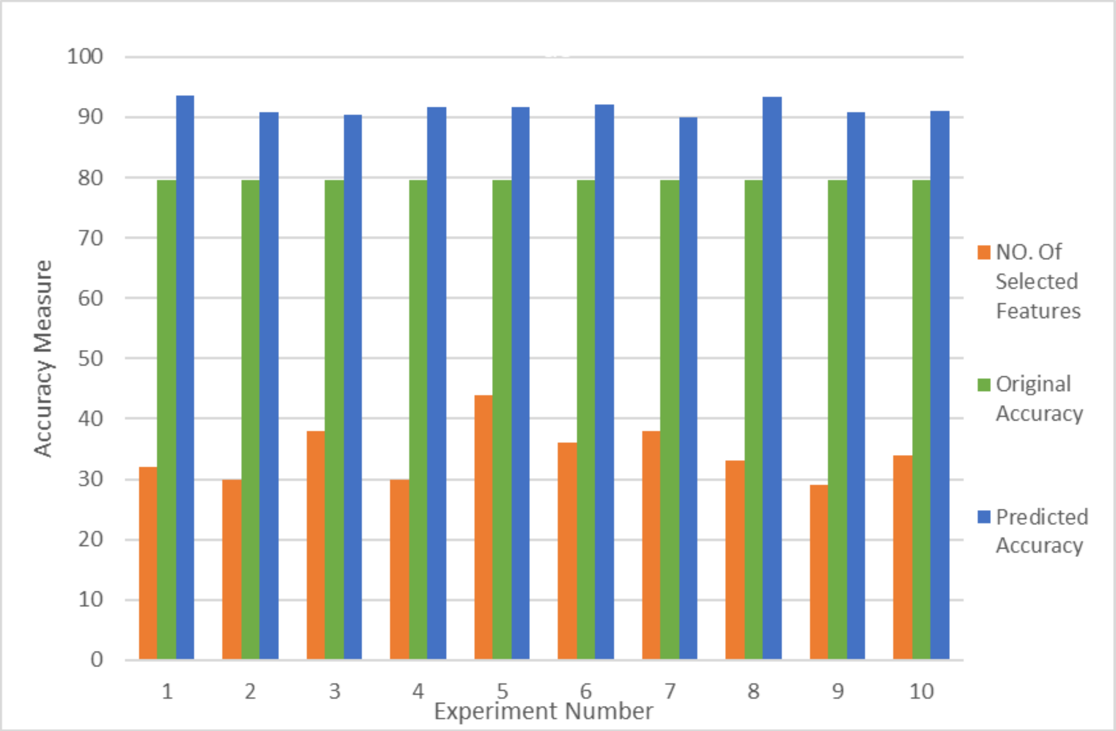


Figure 4.16 Accuracy and Selected Features for 400 iterations, swarm size=10.

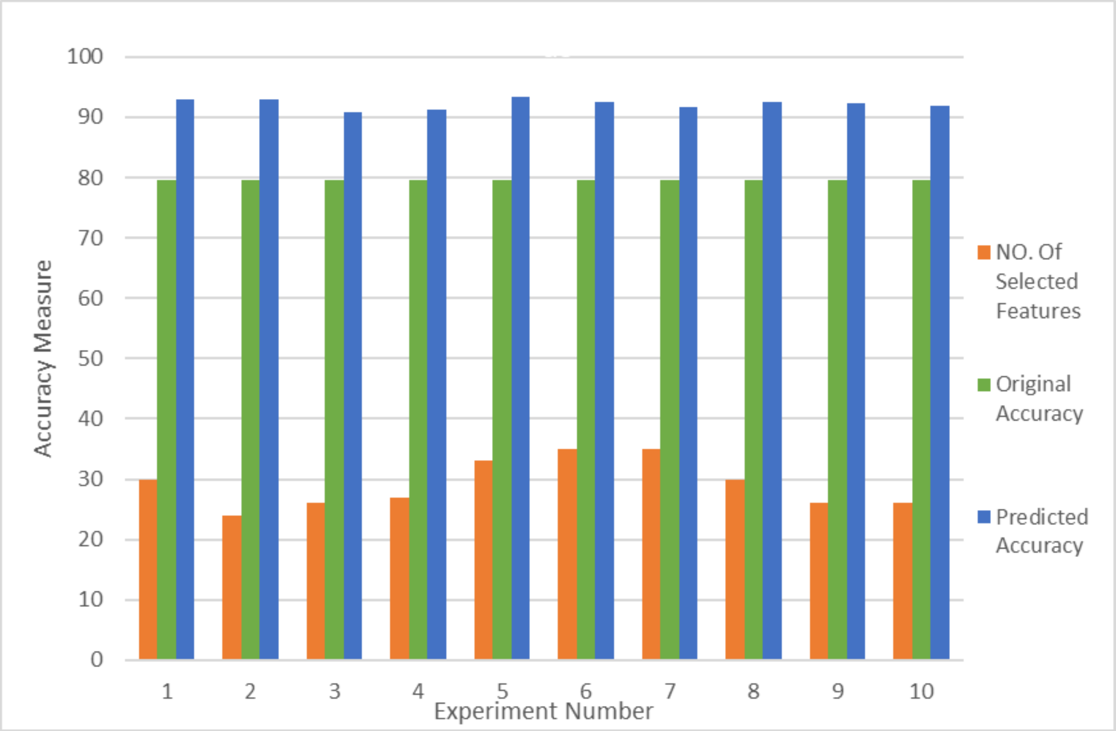


Figure 4.17 Accuracy and Selected Features for 400 iterations, swarm size=20.

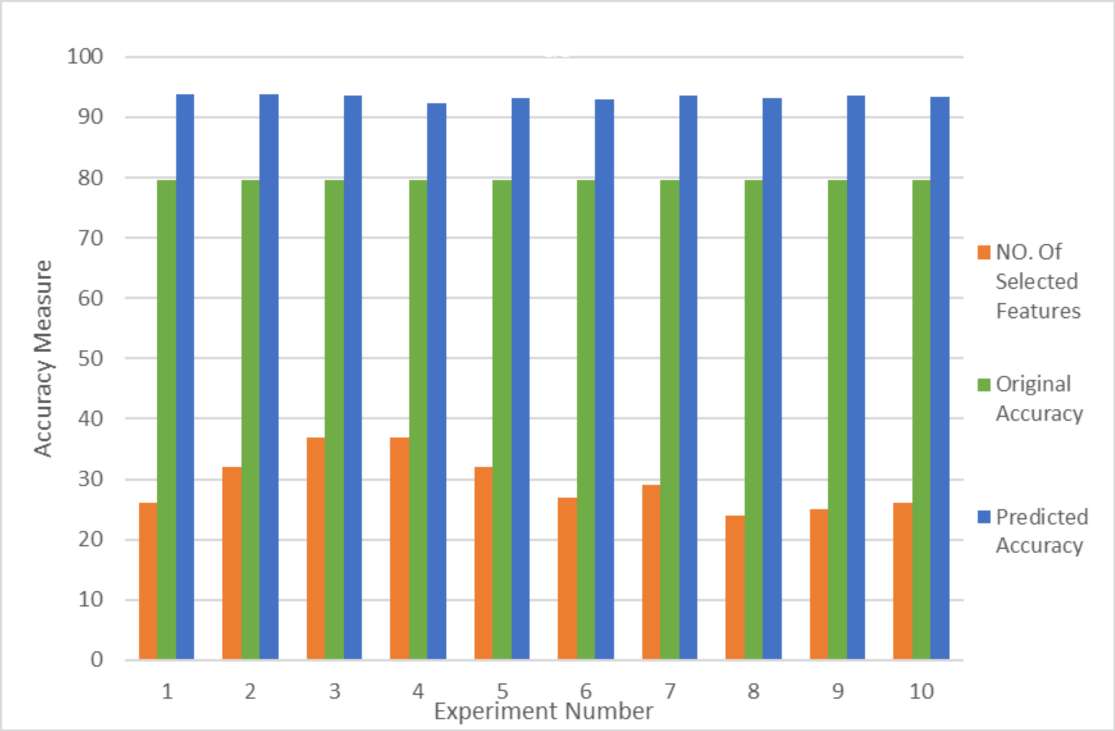


Figure 4.18 Accuracy and Selected Features for 400 iterations, swarm size=30.

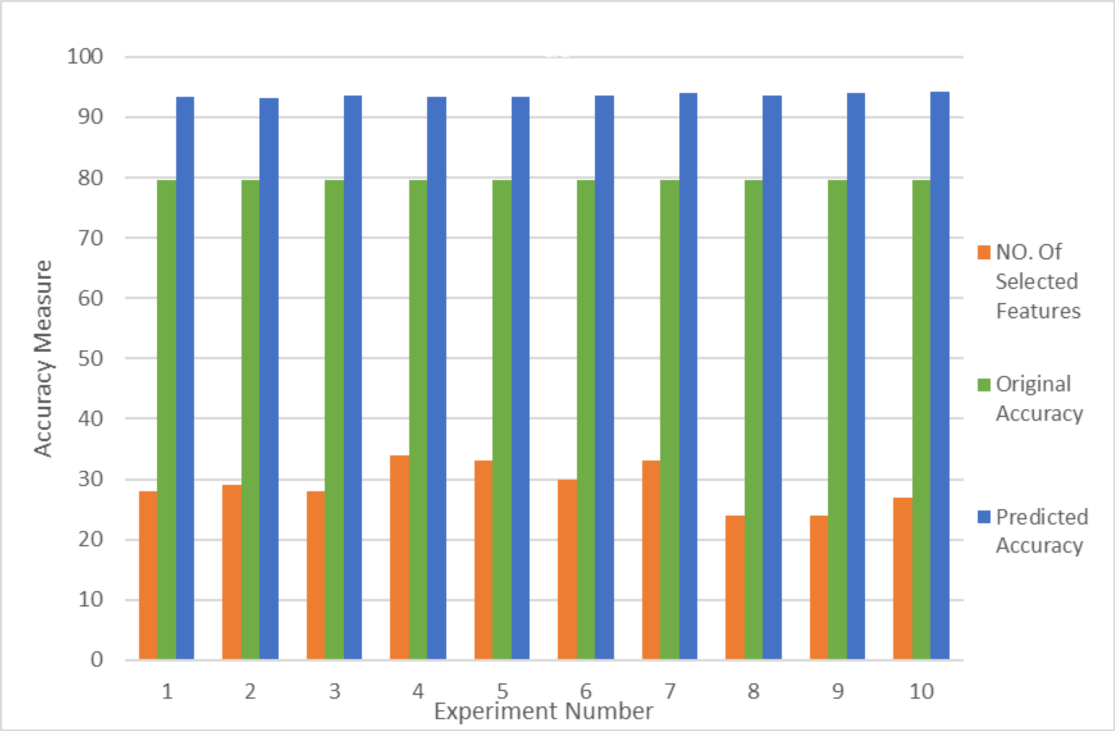


Figure 4.19 Accuracy and Selected Features for 400 iterations, swarm size=40.

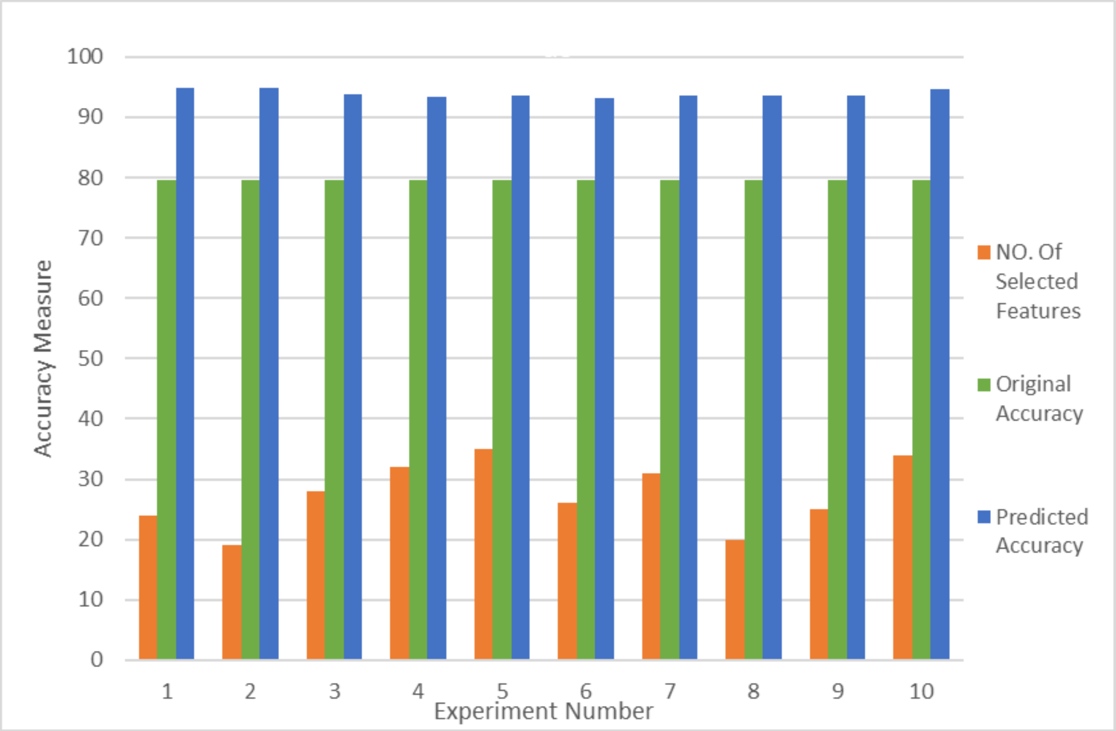


Figure 4.20 Accuracy and Selected Features for 400 iterations, swarm size=50.

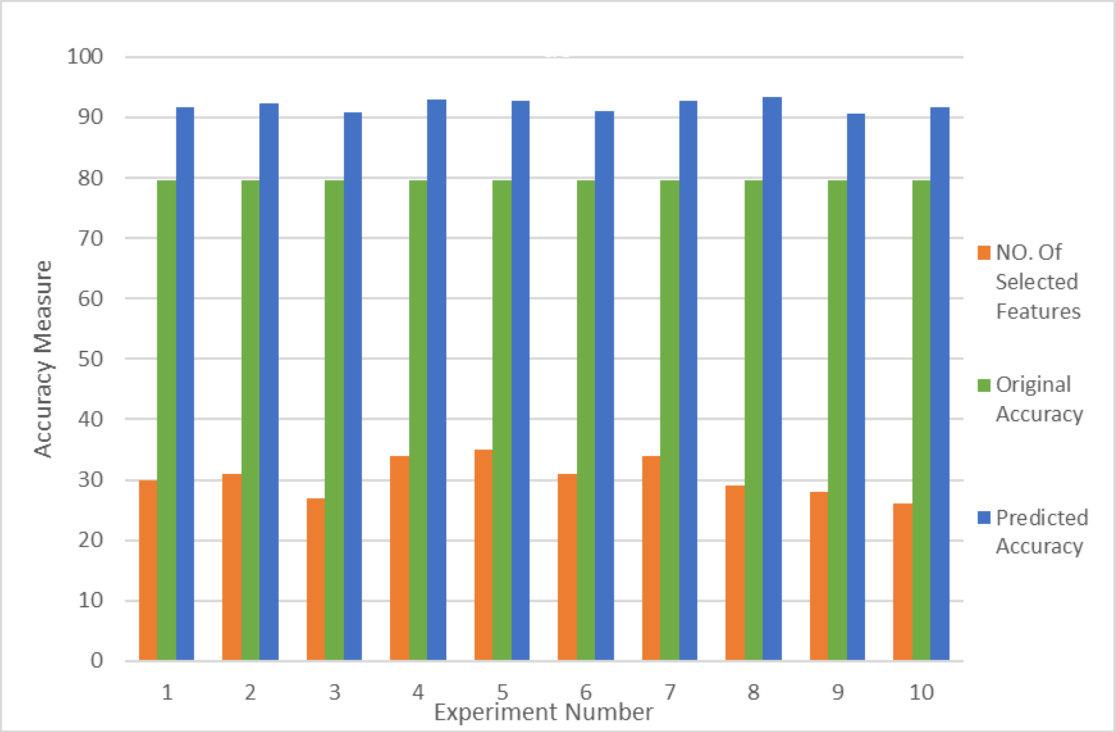


Figure 4.21 Accuracy and Selected Features for 500 iterations, swarm size=10.

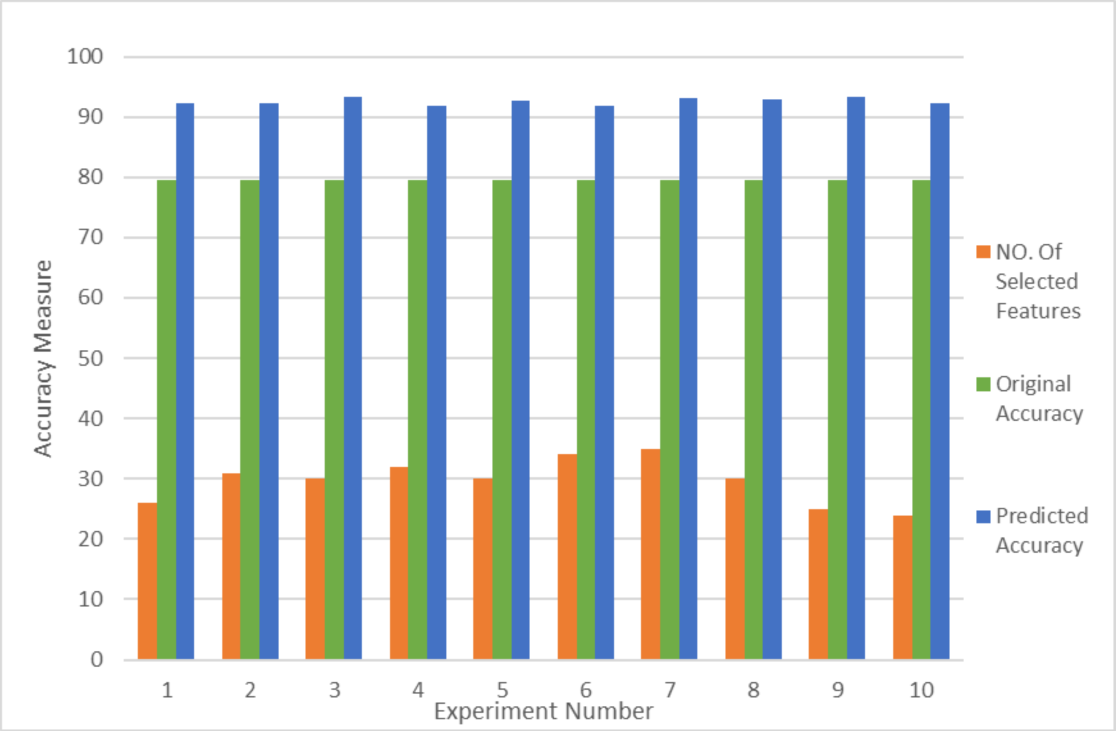


Figure 4.22 Accuracy and Selected Features for 500 iterations, swarm size=20.

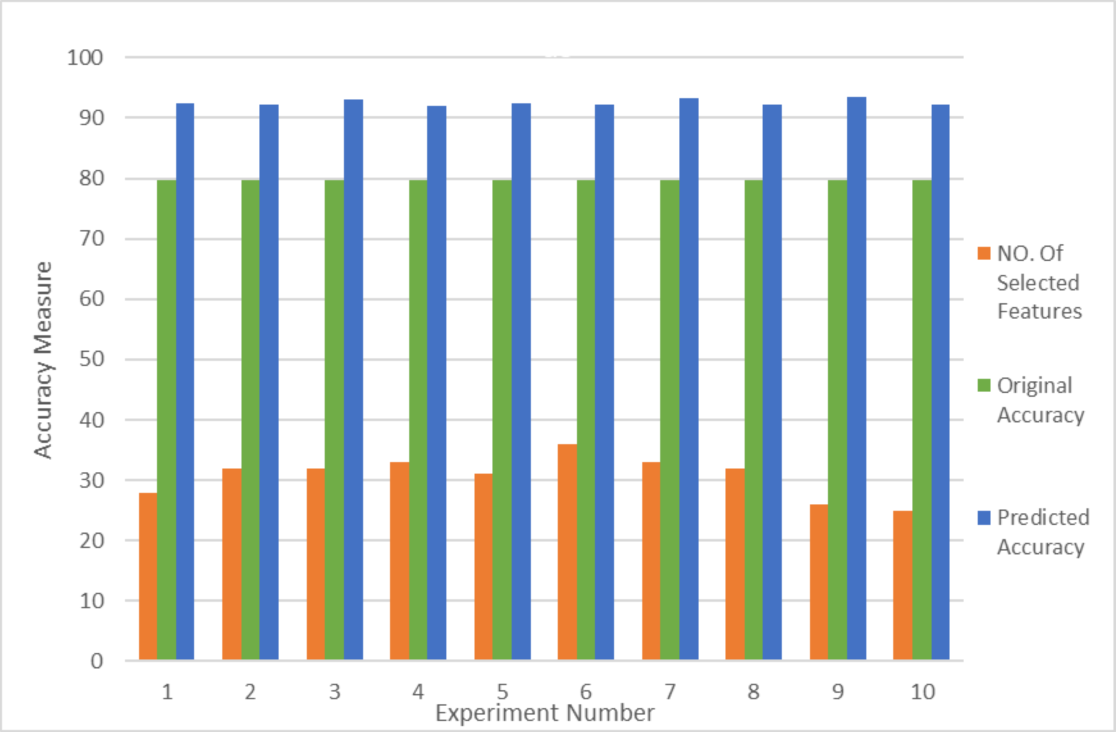


Figure 4.23 Accuracy and Selected Features for 500 iterations, swarm size=30.

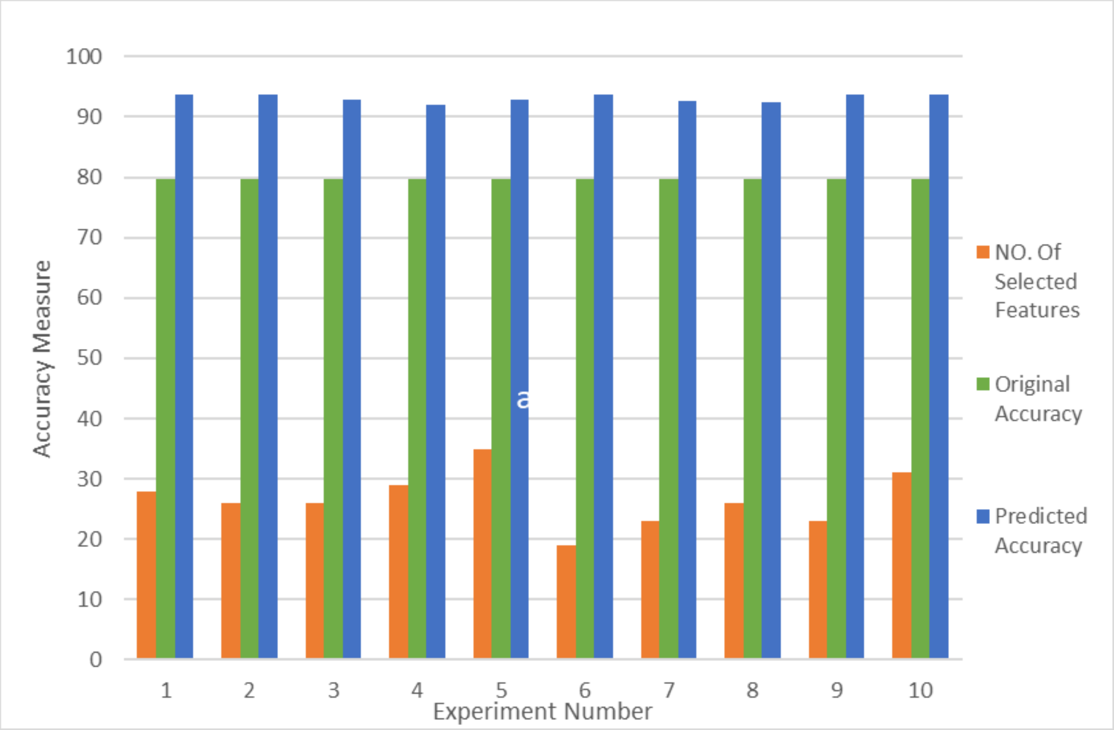


Figure 4.24 Accuracy and Selected Features for 500 iterations, swarm size=40.

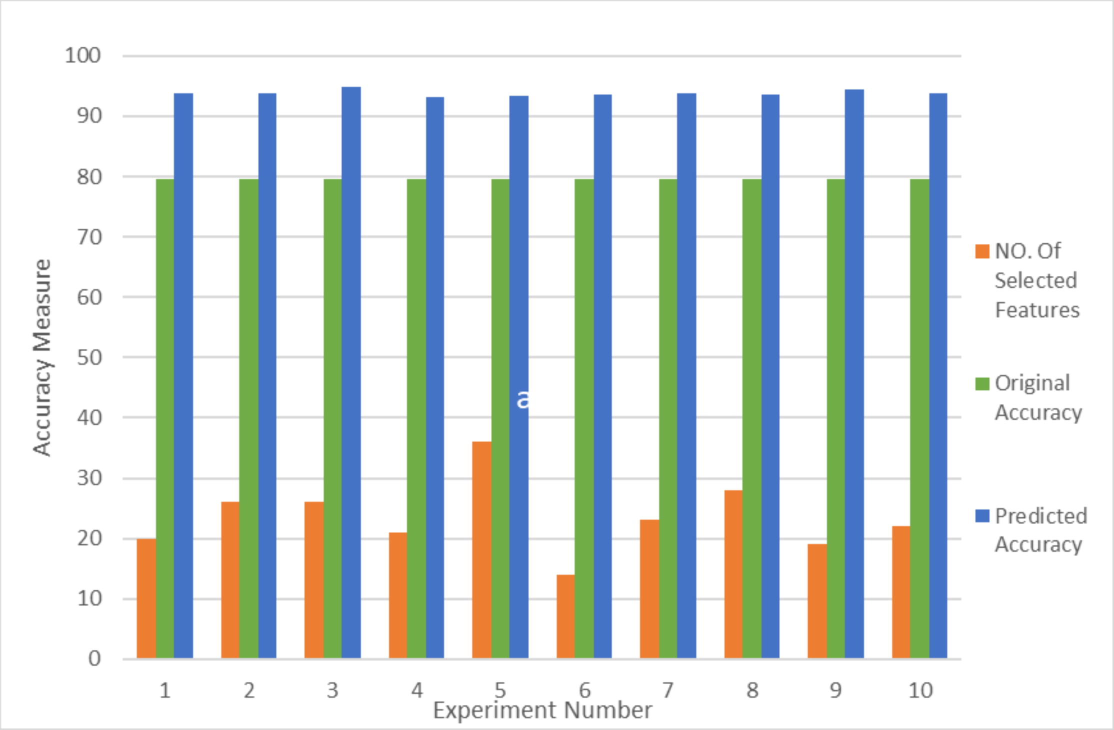


Figure 4.25 Accuracy and Selected Features for 500 iterations, swarm size=50.

4.4. Discussion

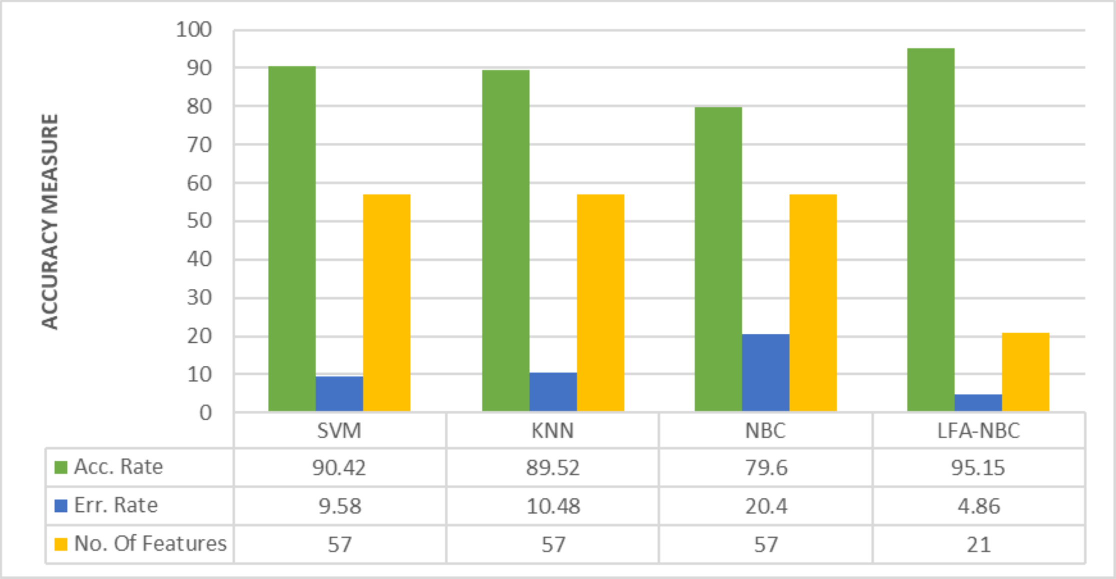
The previous section presented the attained results after implementing the proposed LFA algorithm with Naïve Bayes, where different swarm sizes are tested with 10 runs for each. Overall, increasing swarm size of LFA algorithm positively increased classification accuracy of Naïve Bayes towards Spam Emails. It was also evident from the results that increasing the number of iterations of the classifier also increased the prediction accuracy, but with less observed effect when compared to swarm size. Table 4.2 summarizes the best accuracy, worst accuracy, average accuracy, and standard deviation of NB classification results.

Table 4.2 Summarized Results

No. of Iterations	**Swarm Size**	**Best Accuracy**	**Worst Accuracy**	**Average Accuracy**	**Standard Deviation**
100	10	93.01	90.19	91.6	0.712
	20	93.1	91.02	92.06	0.921
	30	94.23	91.52	92.875	0.71
	40	93.98	92.1	93.04	0.721
	50	94.12	92.34	93.23	0.723
200	10	93.134	90.05	91.592	1.199
	20	93.4	92.036	92.718	1.014
	30	93.7	92.637	93.168	0.432
	40	94.08	93.158	93.619	0.587
	50	94.31	93.386	93.848	0.503
300	10	92.67	90.32	91.495	0.784
	20	93.78	90.9	92.34	1.12
	30	94.32	92.32	93.32	0.81
	40	94.8	92.71	93.755	0.47
	50	94.89	93.52	94.205	0.257
400	10	93.63	90.05	91.84	1.124

	20	93.4	91.01	92.205	0.73
	30	93.67	92.87	93.27	0.428
	40	94.1	93.21	93.655	0.321
	50	94.89	93.3	94.095	0.610
500	10	93.3	91.01	92.155	0.784
	20	93.42	90.8	92.11	0.520
	30	93.4	92.1	92.75	0.642
	40	94.87	93.1	93.985	0.461
	50	95.15	93.61	94.38	0.48

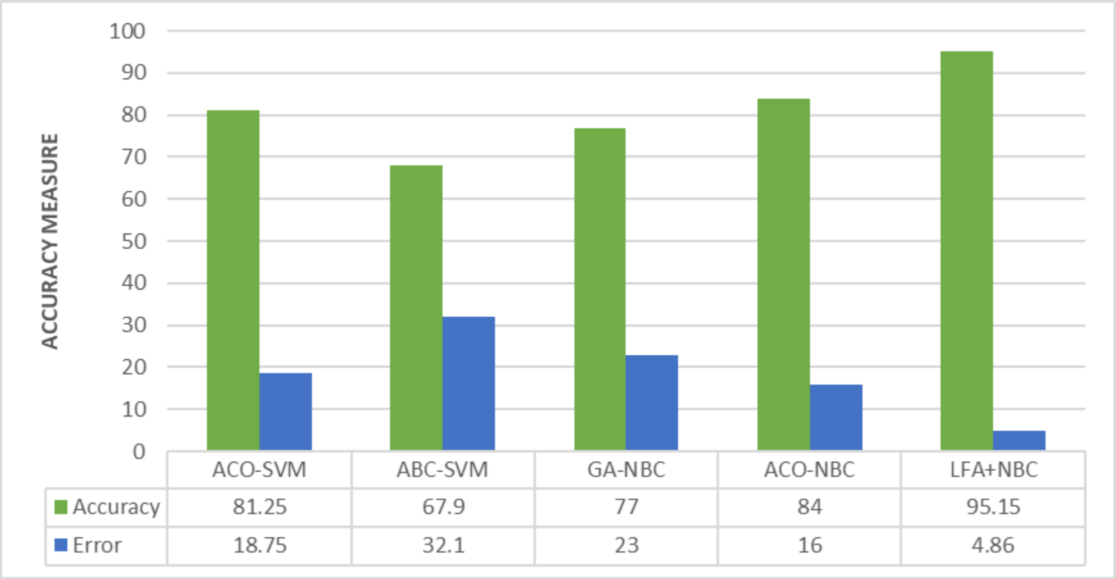
Mathematically, standard deviation is the measure of how spread/distributed a group of numbers around a single line. In machine learning, it is used to show how stable the performance of a given algorithm over several testing scenarios. From standard deviation measure given in Table 4.2, it is evident that LFA-NB is stable, with best measured in 100 and 500 iterations. Last, the proposed LFA-based Naïve Bayes classifier is compared with three of the widely used standard classification models, namely: Support Vector Machine (SVM), K-nearest neighbor (KNN), and Naïve Bayesian Classifier (NBC). When applying the suggested algorithms on the dataset, the total number of features (i.e. 57) was used compared to small subset of features (only 21, given in Appendix C) selected by the proposed LFA selection algorithm. As a result, the selection of the important features helped in enhancing the prediction accuracy to 95.15. Figure 4.26 shows the comparison of classification accuracy of the proposed algorithm and three standard classification models. It is worth to mention that the standard NBC attained 79.6 with all features, while it attained 95.15 when the features selection algorithm – levy+firefly algorithm – is applied.



	SVM	KNN	NBC	LFA-NBC
🟩 Acc. Rate	90.42	89.52	79.6	95.15
🟦 Err. Rate	9.58	10.48	20.4	4.86
🟨 No. Of Features	57	57	57	21

Figure 4.26 Comparison of Proposed Algorithm and Three Standard Models

For further evaluation, the proposed algorithm is compared in Figure 4.27 with other published feature selection algorithms, ACO-SVM [3], ABC-SVM [4], GA-NBC [6], and ACO-NBC [5].



	ACO-SVM	ABC-SVM	GA-NBC	ACO-NBC	LFA+NBC
🟩 Accuracy	81.25	67.9	77	84	95.15
🟦 Error	18.75	32.1	23	16	4.86

Figure 4.27 Comparison of Proposed Algorithm and Published Work

CHAPTER FIVE

CONCLUSION AND FUTURE WORK

5.1. Introduction

E-mails are a widely-used form of communication owing to their low-cost and efficient nature, however their misuse can result in numerous threats such as spam, or unsolicited emails. Spam e-mails can be described as mass e-mails sent randomly containing commercial contents to a certain targeted group of recipients. The task of managing spam is much costlier than the task of sending spam e-mails and therefore there is a resultant wastage of network resources and storage space. Moreover, spam emails contribute to traffic congestion and waste employees' time resulting in lower productivity. These spam e-mails waste an estimated 10 minutes per day on average to the sorting of spam messages, and an estimated cost of billions of dollars is spent annually to manage spam.

Multiple types of anti-spam filters exist today that are designed to filter emails manually. These systems operate on the basis of matching rules that require manual adjustment to each incoming e-mail. Time and experience are needed to operate such a system efficiently and even still it requires constant regular updating of the features of all unwanted messages. On the other hand, the automatic spam filtering systems are considerably more advantageous compared to the manual systems.

Machine learning frameworks have been implemented to classify texts and documents into categories based on its content. These algorithms are initially trained manually on document classification and they can be implemented on e-mail classification into folders and also to identify relevant news content. The Naïve Bayesian Classifier is an intriguing part of data mining, it aids in the classification of messages at which it has excellent precision, and also in recalling unseen messages.

It is worthy to note that the process of text categorization is efficient with anti-spam filtering systems, but it may be defective without anti-spam filtering. E-mails are classified as spam considering their mode of transmission (bulky and hastily) rather than their actual content. However, it seems the language used in spam e-mails, are different from legitimate e-mails; spam e-mails rarely contain real messages which makes it simple to train a text classifier for the purpose of spam filtering.

In this work, the Levy flight + Firefly algorithms (LFA) were implemented to choose the most relevant features that will improve the measured accuracy and predictive performance of the Naïve Bayesian Classifier (NBC). The LFA was initialized using a chaotic logistic map prior to converting the positions to a binary system using the sigmoid function, where 0 denotes the unselected features and 1 denotes the selected features. The NBC was implemented in the proposed algorithm as a fitness function for the evaluation of the solutions.

In general, the proceeding conclusions were reached from the study:

1- The NBC obtained a low accuracy (79.5%) in comparison to the standard KNN or SVM.

2- The LFA algorithm when implemented mostly enhanced the accuracy of the NBC to a minimum accuracy in excess of 95%, suggesting an improved performance of the proposed algorithm in comparison to the standard SVM and KNN.

3- The experiments showed that the performance of the LFA algorithm was influenced by the swarm size; i.e. an increased number of swarms also increased the accuracy of the classifier.

4- Likewise, the amount of iterations was found to only slightly affect the classifier’s performance.

5- The comparisons demonstrated the proposed algorithm outperforms the benchmarking algorithms like ACO, ABC and GA.

5.2. Future works

Future works will target the following areas:

1. Designing and implementing of an online system that will allow the work to be carried out on a server rather than an offline system to allow all incoming e-mails to be classified automatically.
2. Gathering and assembly of a new dataset of e-mail messages rather than using a spam-based dataset that was gathered for research purposes.
3. Implementation of the proposed system with other classification systems like disease diagnosis by utilizing a corresponding dataset.
4. Hybridization of the proposed system with metaheuristic or evolutionary methods like the GA or PSO.
5. Implementing other chaotic maps like Piece-wise map or Tent map to initialize the LFA algorithm.

LIST OF APPENDICES

APPENDIX A – SPAMBASE Features Details

```
| SPAM E-MAIL DATABASE ATTRIBUTES (IN .NAMES FORMAT)
|
| 48 CONTINUOUS REAL [0,100] ATTRIBUTES OF TYPE WORD_FREQ_WORD
| = PERCENTAGE OF WORDS IN THE E-MAIL THAT MATCH WORD,
| I.E. 100 * (NUMBER OF TIMES THE WORD APPEARS IN THE E-MAIL) /
| TOTAL NUMBER OF WORDS IN E-MAIL.  A "WORD" IN THIS CASE IS ANY
| STRING OF ALPHANUMERIC CHARACTERS BOUNDED BY NON-
ALPHANUMERIC
| CHARACTERS OR END-OF-STRING.
|
| 6 CONTINUOUS REAL [0,100] ATTRIBUTES OF TYPE CHAR_FREQ_CHAR
| = PERCENTAGE OF CHARACTERS IN THE E-MAIL THAT MATCH CHAR,
| I.E. 100 * (NUMBER OF CHAR OCCURENCES) / TOTAL CHARACTERS IN E-
MAIL
|
| 1 CONTINUOUS REAL [1,...] ATTRIBUTE OF TYPE
CAPITAL_RUN_LENGTH_AVERAGE
| = AVERAGE LENGTH OF UNINTERRUPTED SEQUENCES OF CAPITAL
LETTERS
|
| 1 CONTINUOUS INTEGER [1,...] ATTRIBUTE OF TYPE
CAPITAL_RUN_LENGTH_LONGEST
| = LENGTH OF LONGEST UNINTERRUPTED SEQUENCE OF CAPITAL
LETTERS
|
| 1 CONTINUOUS INTEGER [1,...] ATTRIBUTE OF TYPE
CAPITAL_RUN_LENGTH_TOTAL
| = SUM OF LENGTH OF UNINTERRUPTED SEQUENCES OF CAPITAL
LETTERS
| = TOTAL NUMBER OF CAPITAL LETTERS IN THE E-MAIL
|
| 1 NOMINAL {0,1} CLASS ATTRIBUTE OF TYPE SPAM
| = DENOTES WHETHER THE E-MAIL WAS CONSIDERED SPAM (1) OR NOT
(0),
| I.E. UNSOLICITED COMMERCIAL E-MAIL.
```

| FOR MORE INFORMATION, SEE FILE 'SPAMBASE.DOCUMENTATION' AT THE
| UCI MACHINE LEARNING REPOSITORY:
HTTP://WWW.ICS.UCI.EDU/~MLEARN/MLREPOSITORY.HTML

1, 0. | SPAM, NON-SPAM CLASSES

WORD_FREQ_MAKE:	CONTINUOUS.
WORD_FREQ_ADDRESS:	CONTINUOUS.
WORD_FREQ_ALL:	CONTINUOUS.
WORD_FREQ_3D:	CONTINUOUS.
WORD_FREQ_OUR:	CONTINUOUS.
WORD_FREQ_OVER:	CONTINUOUS.
WORD_FREQ_REMOVE:	CONTINUOUS.
WORD_FREQ_INTERNET:	CONTINUOUS.
WORD_FREQ_ORDER:	CONTINUOUS.
WORD_FREQ_MAIL:	CONTINUOUS.
WORD_FREQ_RECEIVE:	CONTINUOUS.
WORD_FREQ_WILL:	CONTINUOUS.
WORD_FREQ_PEOPLE:	CONTINUOUS.
WORD_FREQ_REPORT:	CONTINUOUS.
WORD_FREQ_ADDRESSES:	CONTINUOUS.
WORD_FREQ_FREE:	CONTINUOUS.
WORD_FREQ_BUSINESS:	CONTINUOUS.
WORD_FREQ_EMAIL:	CONTINUOUS.
WORD_FREQ_YOU:	CONTINUOUS.
WORD_FREQ_CREDIT:	CONTINUOUS.
WORD_FREQ_YOUR:	CONTINUOUS.
WORD_FREQ_FONT:	CONTINUOUS.
WORD_FREQ_000:	CONTINUOUS.
WORD_FREQ_MONEY:	CONTINUOUS.
WORD_FREQ_HP:	CONTINUOUS.
WORD_FREQ_HPL:	CONTINUOUS.
WORD_FREQ_GEORGE:	CONTINUOUS.
WORD_FREQ_650:	CONTINUOUS.
WORD_FREQ_LAB:	CONTINUOUS.
WORD_FREQ_LABS:	CONTINUOUS.
WORD_FREQ_TELNET:	CONTINUOUS.
WORD_FREQ_857:	CONTINUOUS.
WORD_FREQ_DATA:	CONTINUOUS.
WORD_FREQ_415:	CONTINUOUS.
WORD_FREQ_85:	CONTINUOUS.
WORD_FREQ_TECHNOLOGY:	CONTINUOUS.
WORD_FREQ_1999:	CONTINUOUS.
WORD_FREQ_PARTS:	CONTINUOUS.

```
WORD_FREQ_PM:                                        CONTINUOUS.
WORD_FREQ_DIRECT:                                    CONTINUOUS.
WORD_FREQ_CS:                                        CONTINUOUS.
WORD_FREQ_MEETING:                                   CONTINUOUS.
WORD_FREQ_ORIGINAL:                                  CONTINUOUS.
WORD_FREQ_PROJECT:                                   CONTINUOUS.
WORD_FREQ_RE:                                        CONTINUOUS.
WORD_FREQ_EDU:                                       CONTINUOUS.
WORD_FREQ_TABLE:                                     CONTINUOUS.
WORD_FREQ_CONFERENCE:                                CONTINUOUS.
CHAR_FREQ_;:                                         CONTINUOUS.
CHAR_FREQ_(:                                         CONTINUOUS.
CHAR_FREQ_[:                                         CONTINUOUS.
CHAR_FREQ_!:                                         CONTINUOUS.
CHAR_FREQ_$:                                         CONTINUOUS.
CHAR_FREQ_#:                                         CONTINUOUS.
CAPITAL_RUN_LENGTH_AVERAGE:                          CONTINUOUS.
CAPITAL_RUN_LENGTH_LONGEST:                          CONTINUOUS.
CAPITAL_RUN_LENGTH_TOTAL:                            CONTINUOUS.


ATTRIBUTE STATISTICS:
  MIN: MAX:   AVERAGE:  STD.DEV: COEFF.VAR_%:
1  0    4.54   0.10455   0.30536  292
2  0    14.28  0.21301   1.2906   606
3  0    5.1    0.28066   0.50414  180
4  0    42.81  0.065425  1.3952   2130
5  0    10     0.31222   0.67251  215
6  0    5.88   0.095901  0.27382  286
7  0    7.27   0.11421   0.39144  343
8  0    11.11  0.10529   0.40107  381
9  0    5.26   0.090067  0.27862  309
10 0    18.18  0.23941   0.64476  269
11 0    2.61   0.059824  0.20154  337
12 0    9.67   0.5417    0.8617   159
13 0    5.55   0.09393   0.30104  320
14 0    10     0.058626  0.33518  572
15 0    4.41   0.049205  0.25884  526
16 0    20     0.24885   0.82579  332
17 0    7.14   0.14259   0.44406  311
18 0    9.09   0.18474   0.53112  287
19 0    18.75  1.6621    1.7755   107
20 0    18.18  0.085577  0.50977  596
21 0    11.11  0.80976   1.2008   148
22 0    17.1   0.1212    1.0258   846
23 0    5.45   0.10165   0.35029  345
```

24	0	12.5	0.094269	0.44264	470
25	0	20.83	0.5495	1.6713	304
26	0	16.66	0.26538	0.88696	334
27	0	33.33	0.7673	3.3673	439
28	0	9.09	0.12484	0.53858	431
29	0	14.28	0.098915	0.59333	600
30	0	5.88	0.10285	0.45668	444
31	0	12.5	0.064753	0.40339	623
32	0	4.76	0.047048	0.32856	698
33	0	18.18	0.097229	0.55591	572
34	0	4.76	0.047835	0.32945	689
35	0	20	0.10541	0.53226	505
36	0	7.69	0.097477	0.40262	413
37	0	6.89	0.13695	0.42345	309
38	0	8.33	0.013201	0.22065	1670
39	0	11.11	0.078629	0.43467	553
40	0	4.76	0.064834	0.34992	540
41	0	7.14	0.043667	0.3612	827
42	0	14.28	0.13234	0.76682	579
43	0	3.57	0.046099	0.22381	486
44	0	20	0.079196	0.62198	785
45	0	21.42	0.30122	1.0117	336
46	0	22.05	0.17982	0.91112	507
47	0	2.17	0.0054445	0.076274	1400
48	0	10	0.031869	0.28573	897
49	0	4.385	0.038575	0.24347	631
50	0	9.752	0.13903	0.27036	194
51	0	4.081	0.016976	0.10939	644
52	0	32.478	0.26907	0.81567	303
53	0	6.003	0.075811	0.24588	324
54	0	19.829	0.044238	0.42934	971
55	1	1102.5	5.1915	31.729	611
56	1	9989	52.173	194.89	374
57	1	15841	283.29	606.35	214
58	0	1	0.39404	0.4887	124

APPENDIX B – Recall and Precision Measures

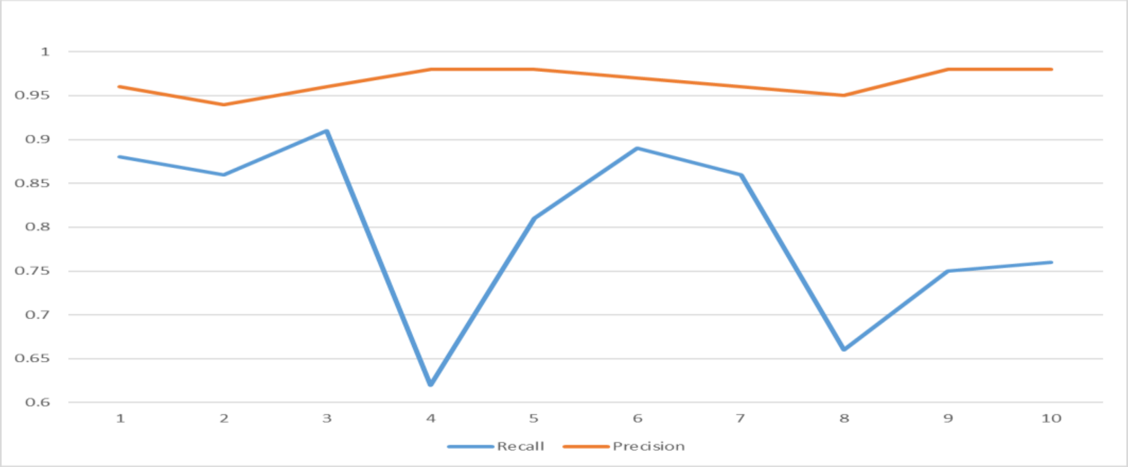


Figure 6.1 The results of swarm size=10 and number of iterations=100

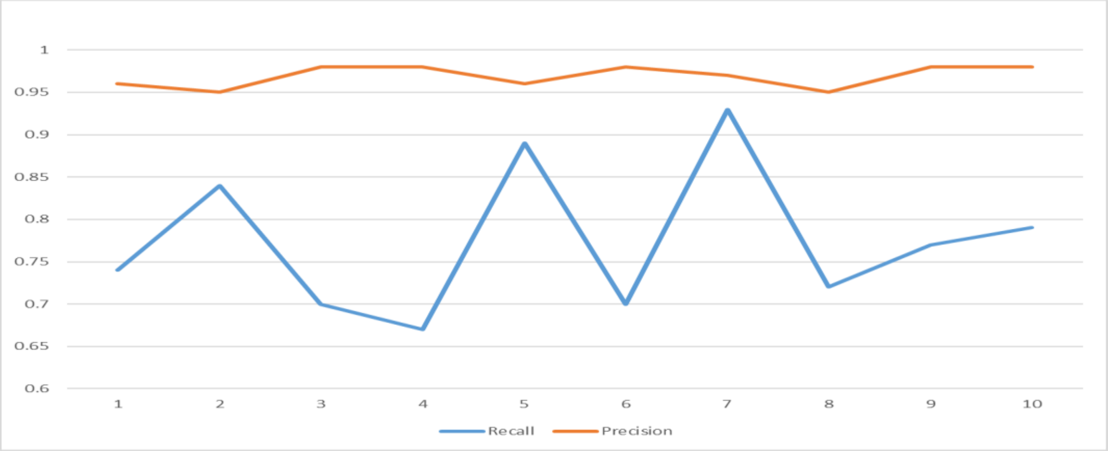


Figure 6.2 The results of swarm size=20 and number of iterations=100

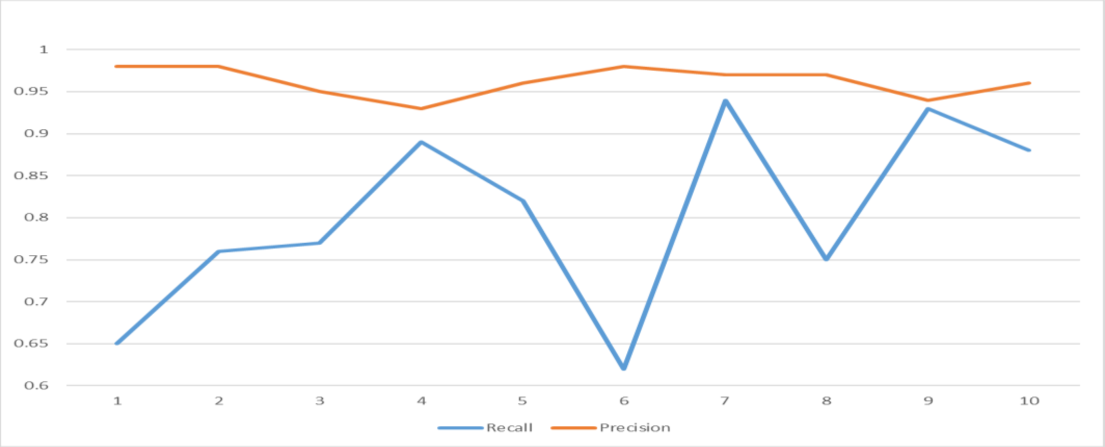


Figure 6.3 The results of swarm size=30 and number of iterations=100

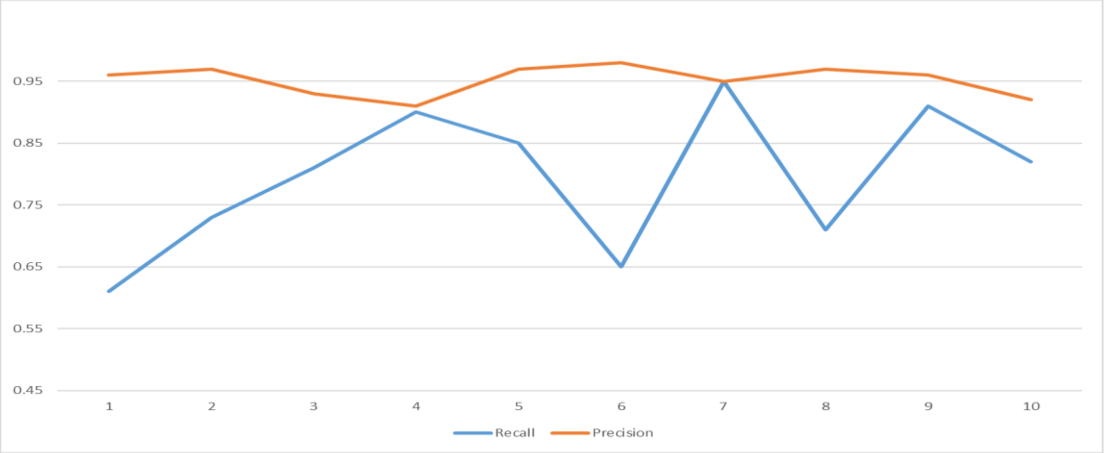


Figure 6.4 The results of swarm size=40 and number of iterations=100

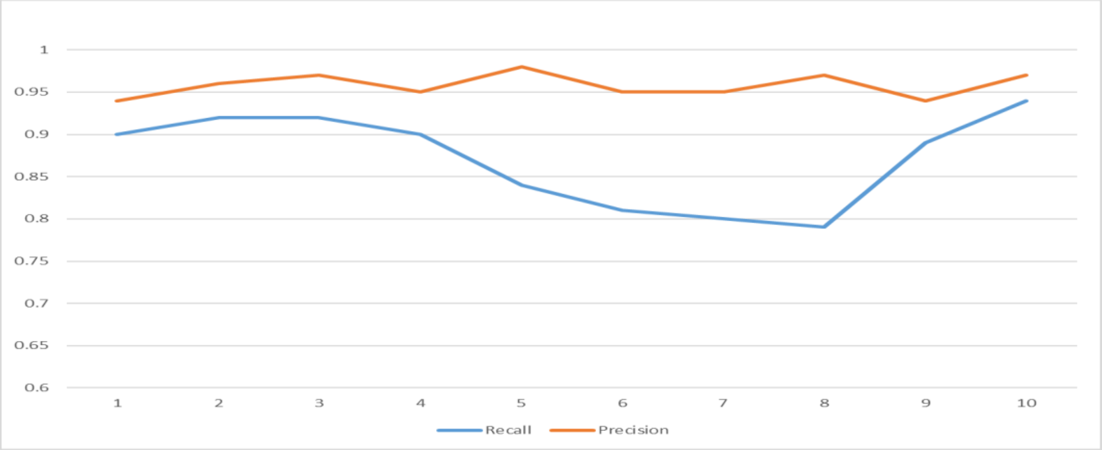


Figure 6.5 The results of swarm size=50 and number of iterations=100

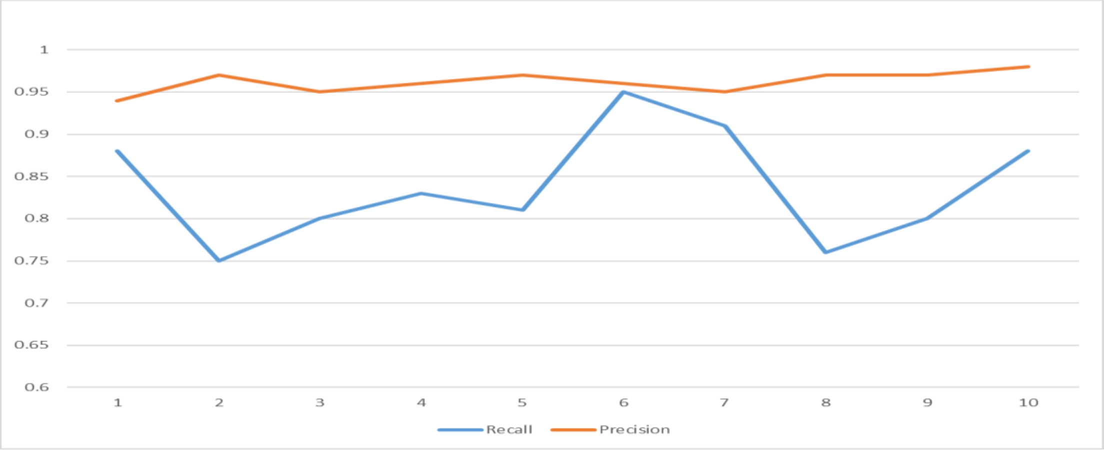


Figure 6.6 The results of swarm size=10 and number of iterations=200

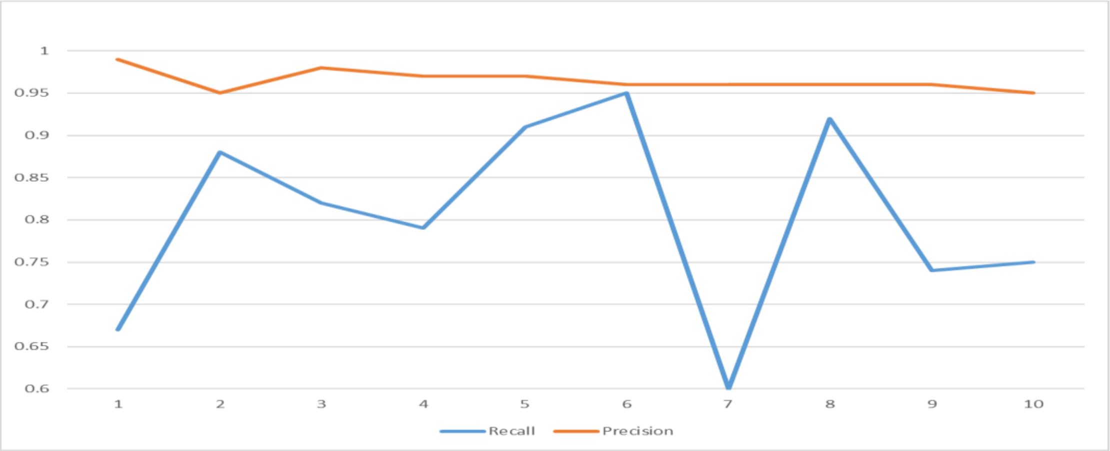


Figure 6.7 The results of swarm size=20 and number of iterations=200

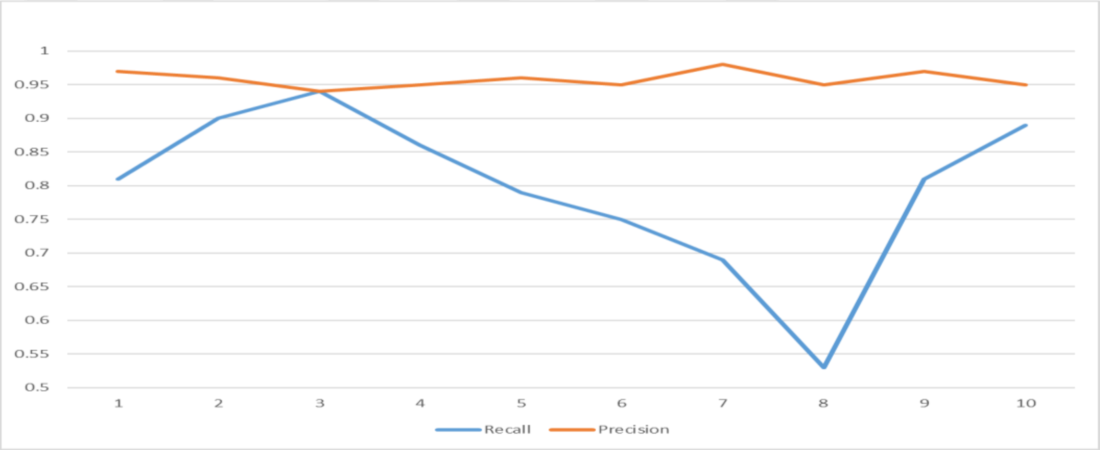


Figure 6.8 The results of swarm size=30 and number of iterations=200

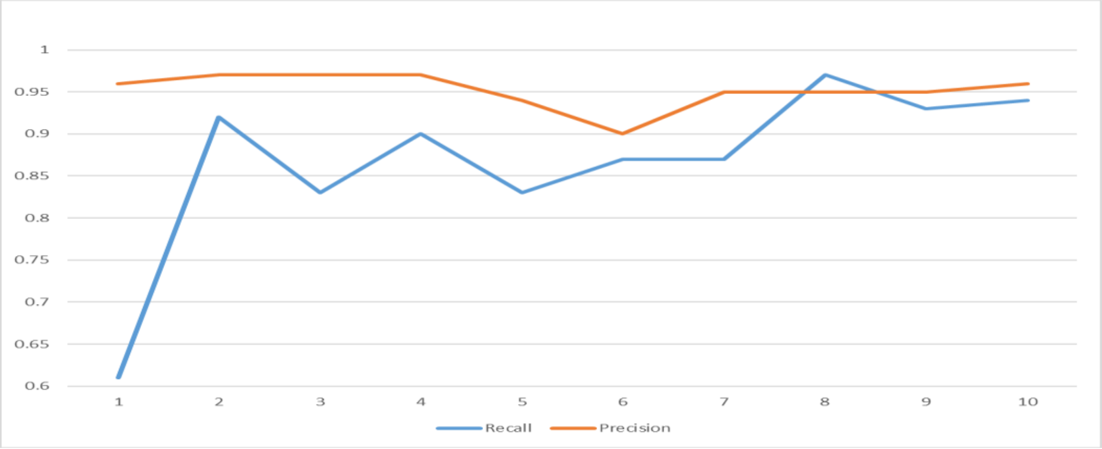


Figure 6.9 The results of swarm size=40 and number of iterations=200

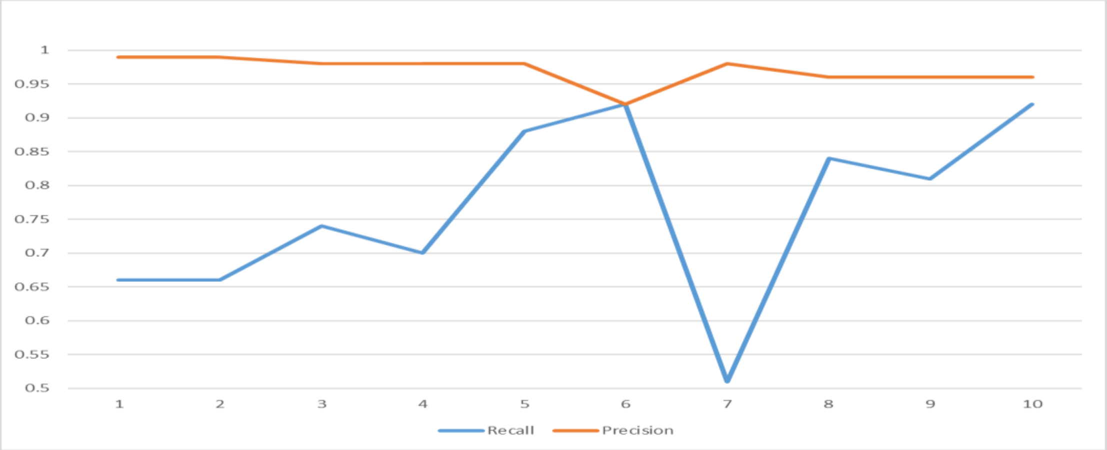


Figure 6.10 The results of swarm size=50 and number of iterations=200

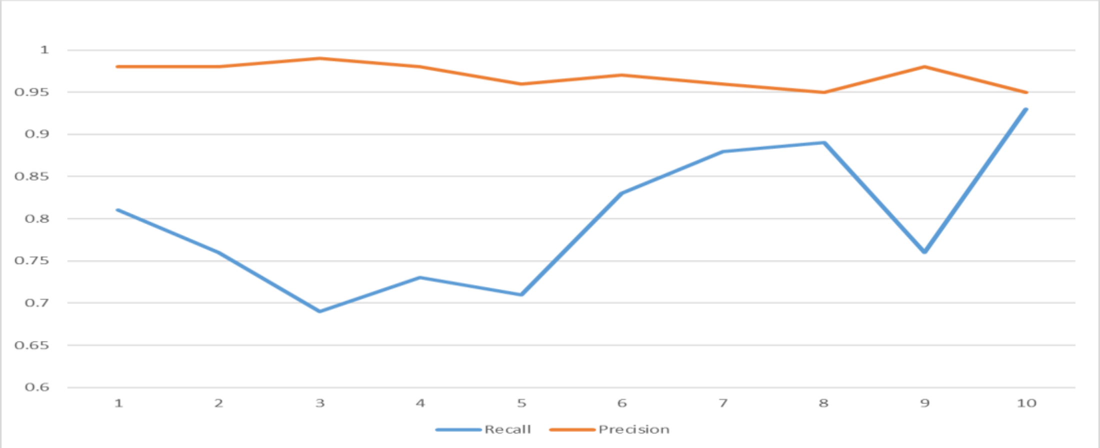


Figure 6.11 The results of swarm size=10 and number of iterations=300

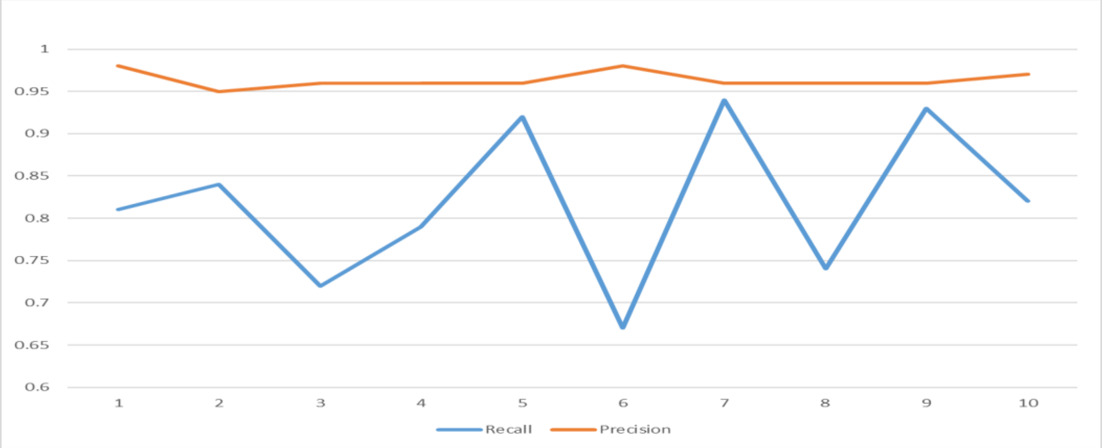


Figure 6.12 The results of swarm size=20 and number of iterations=300

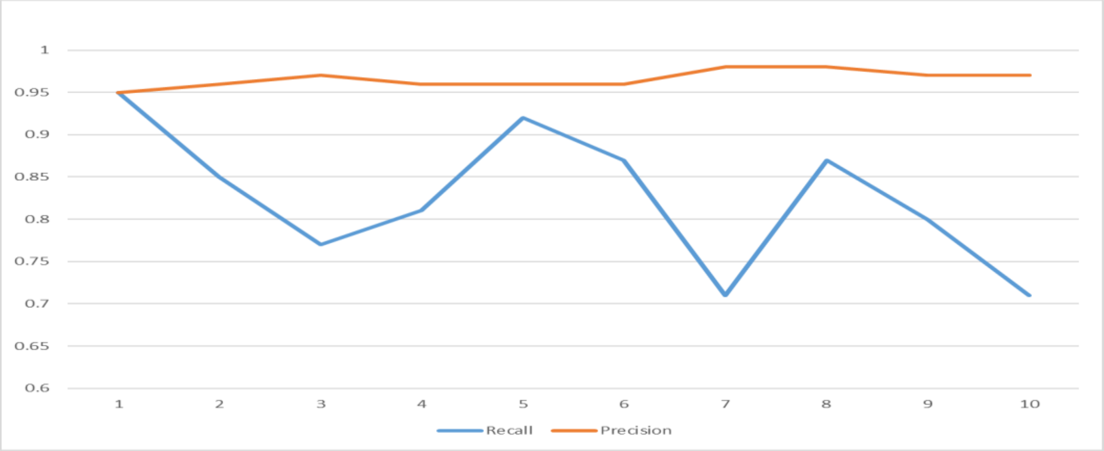


Figure 6.13 The results of swarm size=30 and number of iterations=300

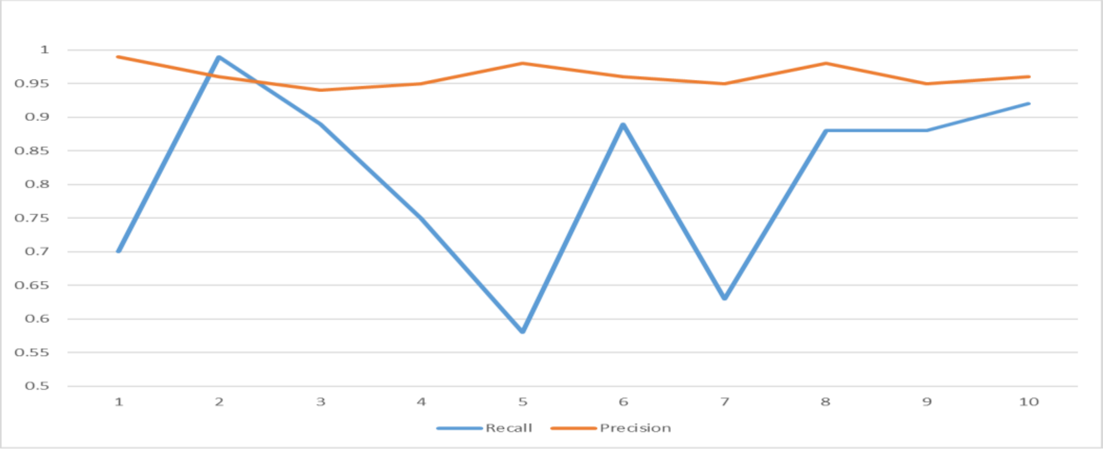


Figure 6.14 The results of swarm size=40 and number of iterations=300

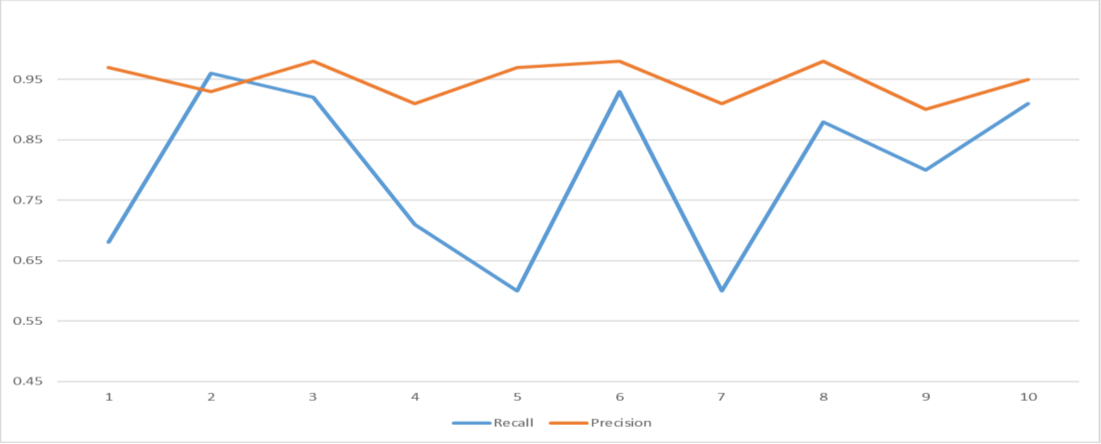


Figure 6.15 The results of swarm size=50 and number of iterations=300

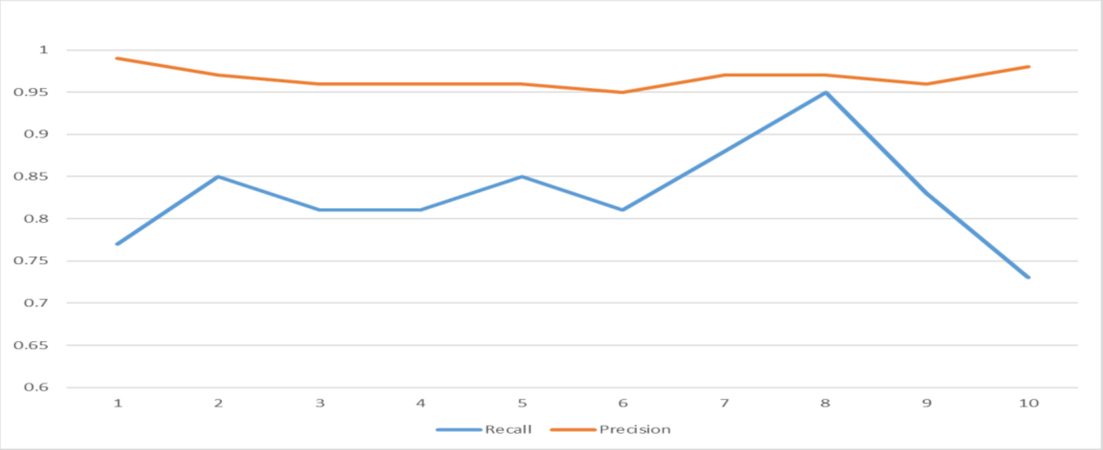


Figure 6.16 The results of swarm size=10 and number of iterations=400

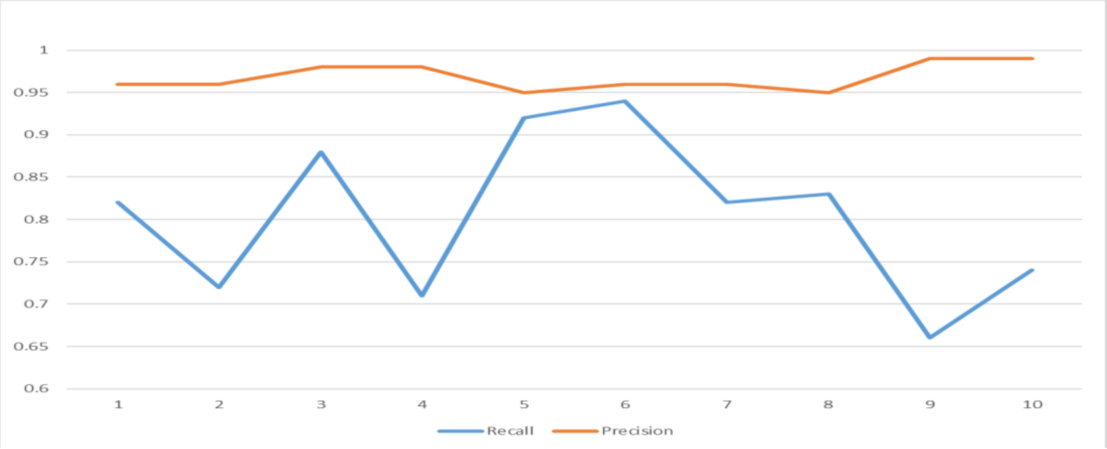


Figure 6.17 The results of swarm size=20 and number of iterations=400

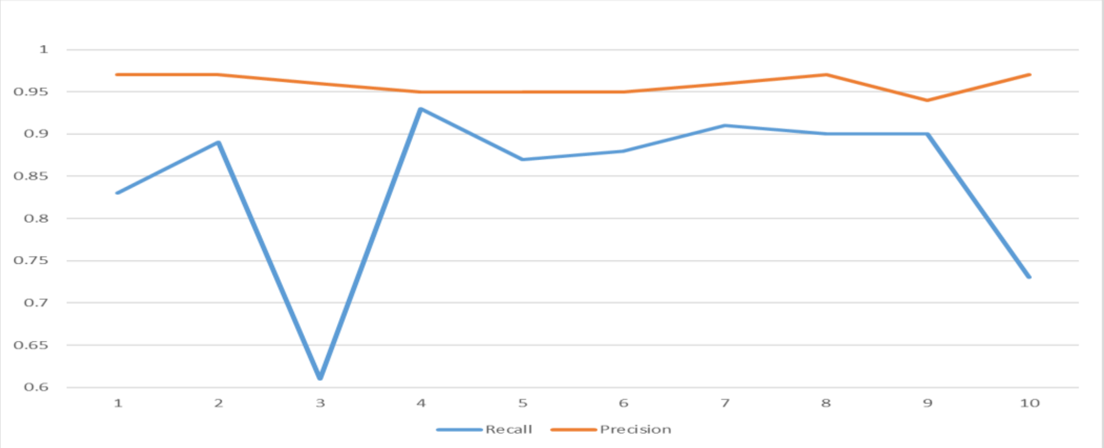


Figure 6.18 The results of swarm size=30 and number of iterations=400

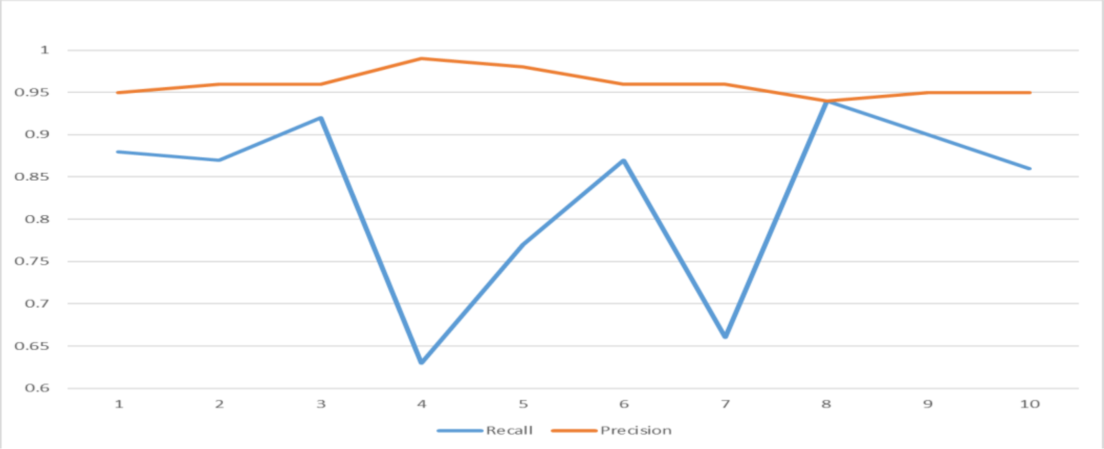


Figure 6.19 The results of swarm size=40 and number of iterations=400

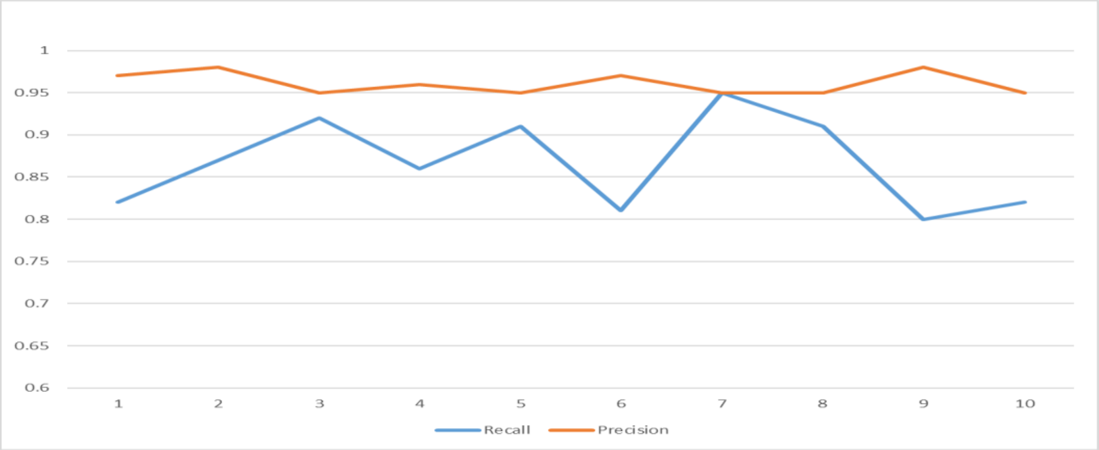


Figure 6.20 The results of swarm size=50 and number of iterations=400

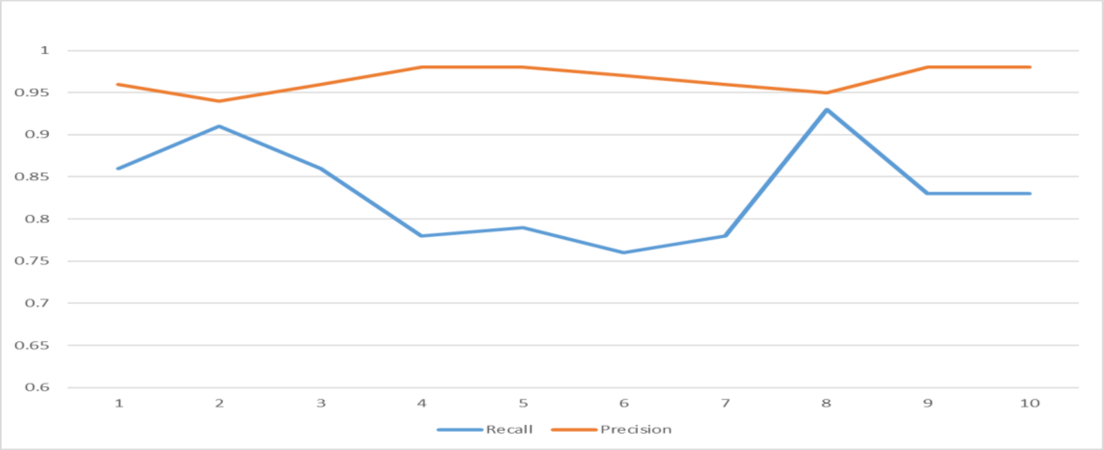


Figure 6.21 The results of swarm size=10 and number of iterations=500

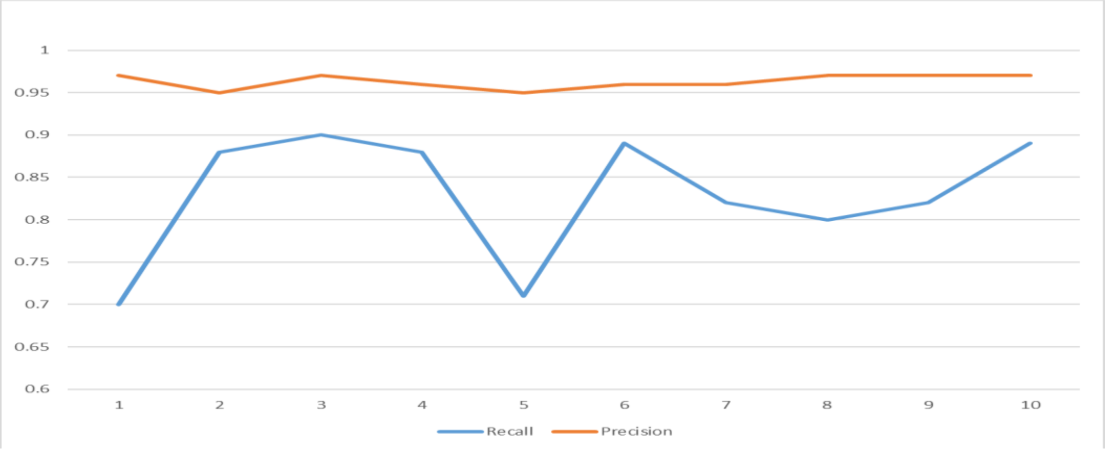


Figure 6.22 The results of swarm size=20 and number of iterations=500

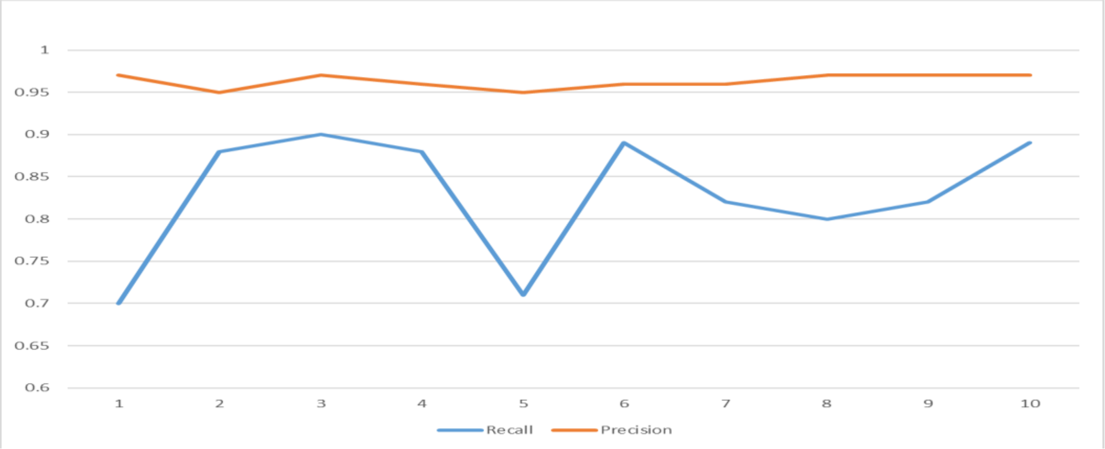


Figure 6.23 The results of swarm size=30 and number of iterations=500

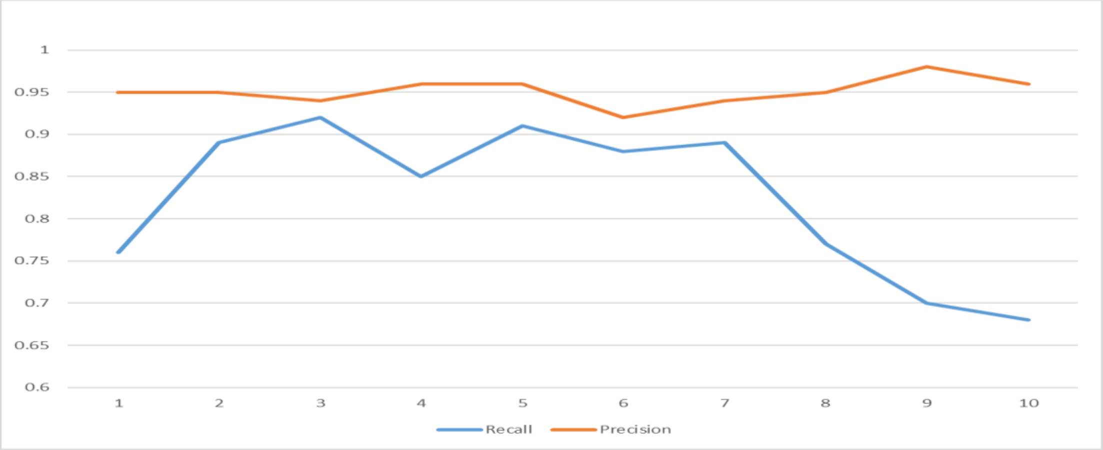


Figure 6.24 The results of swarm size=40 and number of iterations=500

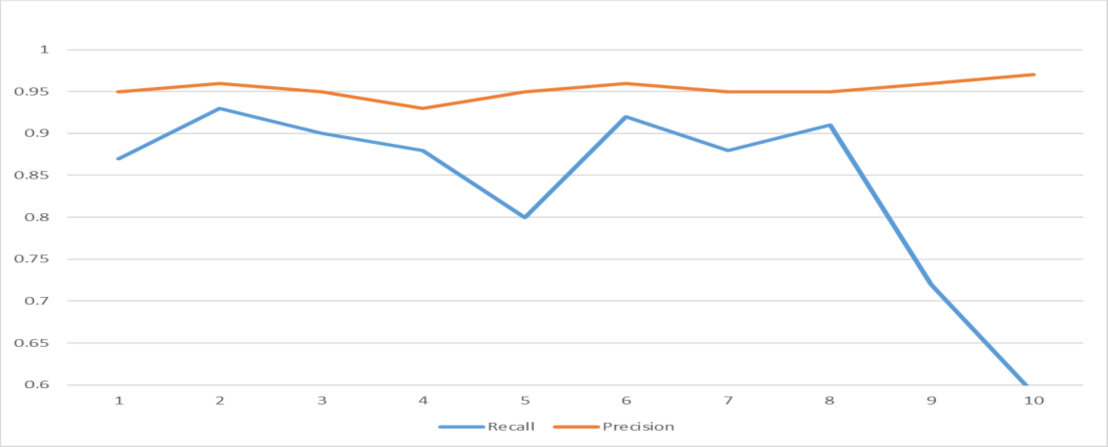


Figure 6.25 The results of swarm size=50 and number of iterations=500

APPENDIX C – Selected Features by Proposed Approach

WORD_FREQ_ADDRESS:	CONTINUOUS.
WORD_FREQ_OVER:	CONTINUOUS.
WORD_FREQ_INTERNET:	CONTINUOUS.
WORD_FREQ_ORDER:	CONTINUOUS.
WORD_FREQ_MAIL:	CONTINUOUS.
WORD_FREQ_RECEIVE:	CONTINUOUS.
WORD_FREQ_REPORT:	CONTINUOUS.
WORD_FREQ_ADDRESSES:	CONTINUOUS.
WORD_FREQ_BUSINESS:	CONTINUOUS.
WORD_FREQ_YOU:	CONTINUOUS.
WORD_FREQ_CREDIT:	CONTINUOUS.
WORD_FREQ_MONEY:	CONTINUOUS.
WORD_FREQ_HPL:	CONTINUOUS.
WORD_FREQ_LAB:	CONTINUOUS.
WORD_FREQ_DATA:	CONTINUOUS.
WORD_FREQ_DIRECT:	CONTINUOUS.
WORD_FREQ_CS:	CONTINUOUS.
WORD_FREQ_PROJECT:	CONTINUOUS.
WORD_FREQ_EDU:	CONTINUOUS.
WORD_FREQ_CONFERENCE:	CONTINUOUS.
CAPITAL_RUN_LENGTH_TOTAL:	CONTINUOUS.

In each experiment, the algorithm is executed according to the number of iterations and the Swarm size. This experiment is repeated 10 times to make sure that it is stable. For each iteration, the selected features will be chosen randomly and the accuracy will be given with the selected features. At the end of each experiment, 10 different features and 10 different accuracy will be obtained. The results of each experiment are shown in chapter 4. Also, Table 4.2 shows the best accuracy, worst accuracy, average accuracy and the standard deviation.

REFERENCES

[1] Lee, S. M., Kim, D. S., Kim, J. H., & Park, J. S. (2010, February). Spam detection using feature selection and parameters optimization. In *Complex, Intelligent and Software Intensive Systems (CISIS), 2010 International Conference on* (pp. 883-888). IEEE.

[2] DeBarr, D., & Wechsler, H. (2012). Spam detection using random boost. Pattern Recognition Letters, 33(10), 1237-1244.

[3] Adewumi, O. A., & Akinyelu, A. A. (2016). A hybrid firefly and support vector machine classifier for phishing email detection. Kybernetes, 45(6), 977-994.

[4] Shi, L., Wang, Q., Ma, X., Weng, M., & Qiao, H. (2012). Spam email classification using decision tree ensemble. Journal of Computational Information Systems, 8(3), 949-956.

[5] Tuteja, S. K., & Bogiri, N. (2016, September). Email Spam filtering using BPNN classification algorithm. In *Automatic Control and Dynamic Optimization Techniques (ICACDOT), International Conference on* (pp. 915-919). IEEE.

[6] Rathi, M., & Pareek, V. (2013). Spam mail detection through data mining-A comparative performance analysis. *International* Journal of Modern Education and Computer Science, 5(12), 31.

[7] Sharaff, A., Nagwani, N. K., & Dhadse, A. (2016). Comparative study of classification algorithms for spam email detection. In *Emerging Research in Computing, Information, Communication and Applications* (pp. 237-244). Springer, New Delhi.

[8] Sharma, A. K., Prajapat, S. K., & Aslam, M. (2014, April). A Comparative Study Between Naive Bayes and Neural Network (MLP) Classifier for Spam Email Detection. In *IJCA Proceedings on National Seminar on Recent Advances in Wireless Networks and Communications. Foundation of Computer Science (FCS)* (Vol. 2, pp. 12-16).

[9] Kaur, H., & Verma, E. P. (2017). E-Mail Spam Detection Using Refined MLP with Feature Selection. International Journal of Modern Education and Computer Science, 9(9), 42.

[10] Kumar, N. S., Rana, D. P., & Mehta, R. G. (2012). Detecting e-mail spam using spam word associations. International Journal of Emerging Technology and Advanced Engineering, 2(4), 222-226.

[11] Günal, S., Ergin, S., Gülmezoğlu, M. B., & Gerek, Ö. N. (2006, September). On feature extraction for spam e-mail detection. In *International Workshop on Multimedia Content Representation, Classification and Security* (pp. 635-642). Springer, Berlin, Heidelberg.

[12] Chen, J., Huang, H., Tian, S., & Qu, Y. (2009). Feature selection for text classification with Naïve Bayes. Expert Systems with Applications, 36(3), 5432-5435.

[13] Yang, Xin-She. "Firefly algorithm, Levy flights and global optimization." Research and development in intelligent systems XXVI. Springer, London, 2010. 209-218.

[14] Sanz, E. P., Hidalgo, J. M. G., & Pérez, J. C. C. (2008). Email spam filtering. Advances in computers, 74, 45-114.

[15] Cormack, G. V. (2008). Email spam filtering: A systematic review. Foundations and Trends® in Information Retrieval, 1(4), 335-455.

[16] Yang, X. S. (2009, October). Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms* (pp. 169-178). Springer, Berlin, Heidelberg.bos

[17] Yang, X. S. (2010). Firefly algorithm, Levy flights and global optimization. In *Research and development in intelligent systems XXVI* (pp. 209-218). Springer, London.

[18] Yang, X. S., & He, X. (2013). Firefly algorithm: recent advances and applications. International Journal of Swarm Intelligence, 1(1), 36-50.

[19] Visalakshi D, K.R., (2015). P.: A Hybrid ACO Based Feature Selection Method for Email Spam Classification.

[20] Dagher, I., & Antoun, R. (2016). Different PCA scenarios for email filtering. International Journal of Computers and Applications, 38(1), 41-54.

[21] Varghese, R., & Dhanya, K. A. (2017, January). Efficient Feature Set for Spam Email Filtering. In *Advance Computing Conference (IACC), 2017 IEEE 7th International* (pp. 732-737). IEEE.

[22] Jindal, N., & Liu, B. (2007, October). Analyzing and detecting review spam. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on* (pp. 547-552). IEEE.

[23] Youn, S., & McLeod, D. (2007, April). Efficient spam email filtering using adaptive ontology. In *Information Technology, 2007. ITNG'07. Fourth International Conference on* (pp. 249-254). IEEE.

[24] Lieven, P., Scheuermann, B., Stini, M., & Mauve, M. (2007, June). Filtering spam email based on retry patterns. In *Communications, 2007. ICC'07. IEEE International Conference on* (pp. 1515-1520). IEEE.

[25] Kumar, S., Gao, X., Welch, I., & Mansoori, M. (2016, March). A machine learning based web spam filtering approach. In *Advanced Information Networking and Applications (AINA), 2016 IEEE 30th International Conference on* (pp. 973-980). IEEE.

[26] Chang, M., & Poon, C. K. (2009). Using phrases as features in email classification. Journal of Systems and Software, 82(6), 1036-1045.

[27] Kumar, S., Gao, X., Welch, I., & Mansoori, M. (2016, March). A machine learning based web spam filtering approach. In *Advanced Information Networking and Applications (AINA), 2016 IEEE 30th International Conference on* (pp. 973-980). IEEE.

[28] Chang, M., & Poon, C. K. (2009). Using phrases as features in email classification. Journal of Systems and Software, 82(6), 1036-1045.

[29] Liu, W., & Wang, T. (2012). Online active multi-field learning for efficient email spam filtering. Knowledge and Information Systems, 33(1), 117-136.

[30] Chen, B., Dong, S., & Fang, W. (2008). Email header feature study for improving Bayesian anti-spam filter. Journal of Computational Information Systems, 4(3), 1205-1212.

[31] Tang, J., Alelyani, S., & Liu, H. (2014). Feature selection for classification: A review. Data Classification: Algorithms and Applications, 37.

[32] Zhang, L., Jiang, L., & Li, C. (2016). A new feature selection approach to naive Bayes text classifiers. International Journal of Pattern Recognition and Artificial Intelligence, 30(02).

[33] Lin, K., Kang, K., Huang, Y., Zhou, C., & Wang, B. (2007). Naive bayes text categorization using improved feature selection. Journal of Computational Information Systems, 3(3), 1159-1164.

[34] Zagorecki, A. (2014, September). Feature selection for naive Bayesian network ensemble using evolutionary algorithms. In *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on* (pp. 381-385). IEEE.

[35] Yao, Y., & Zhou, B. (2010, October). Naive Bayesian rough sets. In *International Conference on Rough Sets and Knowledge Technology* (pp. 719-726). Springer, Berlin, Heidelberg.

[36] Vikramkumar, Vijaykumar, B., & Trilochan. (2014). Bayes and Naive Bayes Classifier. *Computer Science & Engineering. Rajiv Gandhi University of Knowledge Technologies Andhra Pradesh, India*.

[37] Visalakshi, S., & Radha, V. (2014, December). A literature review of feature selection techniques and applications: Review of feature selection in data mining. In *Computational Intelligence and Computing Research (ICCIC), 2014 IEEE International Conference on* (pp. 1-6). IEEE.

[38] Zhao, Z., Morstatter, F., Sharma, S., Alelyani, S., Anand, A., & Liu, H. (2010). Advancing feature selection research. *ASU feature selection repository*, 1-28.

[39] Saeys, Y., Inza, I., & Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. Bioinformatics, 23(19), 2507-2517.

[40] Kumar, V., Minz, S.: Feature Selection : A literature Review. Smart Comput. Rev. 4, 211–229 (2014). doi:10.6029/smartcr.2014.03.007

[41] Mitra, P., Murthy, C. A., & Pal, S. K. (2002). Unsupervised feature selection using feature similarity. IEEE transactions on pattern analysis and machine intelligence, 24(3), 301-312.

[42] Tang, J., Alelyani, S., & Liu, H. (2014). Feature selection for classification: A review. *Data Classification:* Algorithms and Applications, 37.

[43] Xue, B., Zhang, M., & Browne, W. N. (2014). Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. Applied Soft Computing, 18, 261-276.

[44] Kardan, A. A., Kavian, A., & Esmaeili, A. (2013, May). Simultaneous feature selection and feature weighting with K selection for KNN classification using BBO algorithm. In *Information and Knowledge Technology (IKT), 2013 5th Conference on* (pp. 349-354). IEEE.

[45] Kardan, A. A., Kavian, A., & Esmaeili, A. (2013, May). Simultaneous feature selection and feature weighting with K selection for KNN classification using BBO algorithm. In *Information and Knowledge Technology (IKT), 2013 5th Conference on* (pp. 349-354). IEEE.

[46] Zhao, Z., Morstatter, F., Sharma, S., Alelyani, S., Anand, A., & Liu, H. (2010). Advancing feature selection research. *ASU feature selection repository*, 1-28.

[47] Chantar, H. K., & Corne, D. W. (2011, October). Feature subset selection for Arabic document categorization using BPSO-KNN. In *Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on* (pp. 546-551). IEEE.

[48] Pang, X., & Liao, Y. (2010, March). A text classification model based on training sample selection and feature weight adjustement. In *Advanced Computer Control (ICACC), 2010 2nd International Conference on* (Vol. 3, pp. 294-297). IEEE.

[49] Zhang, Z., Hancock, E. (2011). Feature Selection for Gender Classification. Pattern Recognit. Image Anal. 70–75. doi:10.1109/CIBIM.2011.5949221

[50] Tang, J., Alelyani, S., & Liu, H. (2014). Feature selection for classification: A review. Data Classification: Algorithms and Applications, 37-64.

[51] Yang, X. S. (2010). Firefly algorithm, Levy flights and global optimization. In Research and development in intelligent systems XXVI (pp. 209-218). Springer, London.

[52] UCI Machine Learning Repository: Spambase Dataset, http://archive.ics.uci. edu/ml/datasets

[53] DeBarr, D., & Wechsler, H. (2012). Spam detection using random boost. Pattern Recognition Letters, 33(10), 1237-1244.

[54] Visalakshi D, K.R., (2015). P.: A Hybrid ACO Based Feature Selection Method for Email Spam Classification.

[55] Geetika, Aggarwal, H., Singh, G. (2016). A Hybrid Artificial Bee Colony and Support Vector Machine Classifier for Spam Email Detection. Int. J. Control Theory Appl. 10.

[56] Renuka, D. K., Visalakshi, P., & Sankar, T. Improving E-Mail Spam Classification using Ant Colony Optimization Algorithm.

[57] Zagorecki, A. (2014, September). Feature selection for naive Bayesian network ensemble using evolutionary algorithms. In *Computer Science and*

Information Systems (FedCSIS), 2014 Federated Conference on (pp. 381-385). IEEE.