Improved Techniques for Phishing Email Detection based on Random Forest and Firefly-Based Support Vector Machine Learning Algorithms

UNIVERSITY OF
KWAZULU-NATAL

AKINYELU AYOBAMI ANDRONICUS

(213574026)

A thesis submitted in fulfillment of the requirements
for the degree of Master of Science in Computer Science

in the

School of Mathematics, Statistics and Computer Science
University of KwaZulu-Natal
Durban, South Africa

September 2014

UNIVERSITY OF KWAZULU-NATAL

COLLEGE OF AGRICULTURE, ENGINEERING AND SCIENCE

DECLARATION

The research described in this thesis was performed at the University of KwaZulu-Natal under the supervision of Dr. A.O. Adewumi. I hereby declare that all materials incorporated in this thesis are my own original work except where acknowledgement is made by name or in the form of a reference. The work contained herein has not been submitted in part or whole for a degree at any other university.

Signed:

____

Akinyelu, Andronicus Ayobami

Date: September, 2014

As the candidate's supervisor, I have approved/disapproved the dissertation for submission

Signed:

____

Date: September, 2014

Declaration 1 - Plagiarism

I, Akinyelu Ayobami Andronicus, declare that

1. The research reported in this thesis, except where otherwise indicated, is my original research.
2. This thesis has not been submitted for any degree or examination at any other university.
3. This thesis does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.
4. This thesis does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:
    a. Their words have been re-written but, the general information attributed to them has been referenced.
    b. Where their exact words have been used, their writing has been placed in italics and in quotation marks, and referenced.
5. This thesis does not contain text, graphics or tables which are copied and pasted from the internet, unless specifically acknowledged, and the source being detailed in the thesis and in References section.

Signed:

____

Date: September, 2014

Dedication

This research work is dedicated to the Almighty God, who saw me through the thick and thin of this research.

Acknowledgements

Firstly, I want to thank the potentate of my life. I do not know how to express my gratitude, all I can simply say is, all glory and honour be attributed to Him. Secondly, I want to thank my beloved and wonderful parents, Pastor and Mrs Daisi Akinyelu, for giving me the opportunity to embark on this postgraduate journey. Words cannot express how grateful I am; from the depth of my heart, I am saying a big THANK YOU. I also want to thank all my siblings, Opeyemi, Jude and Ebenezer for their words of encouragement and for their care and support. Thirdly, I want to specially thank my supervisor, Dr. Aderemi Oluyinka Adewumi. The truth is, he is more than a supervisor to me. In days of my weariness, he counselled, challenged and charged me; thank you very much for being there for me.

I also want to thank Dr. Siaka (of the Statistics department) for assisting me in the statistical analysis aspect of my work. Furthermore, I want to appreciate Mr. Joseph, Mr. Michael and Dr. Martins for their care, concern and contributions; I am very grateful sirs. I also want to salute my wonderful friends: Joseph Aribodo, Jeremiah Ogunniyi, Tosin Agbaje, Sola Bodede, Peter Merisi, Jeremiah Soji, and all my other friends whose names I have not mentioned, I am very grateful for your contributions and support.

Additionally, I want to thank all the members of the Deeper Life Bible Church (KZN Province) and my office buddies, for all the contributions they have made in one way or another. Finally, I want to say thank you to the University of KwaZulu-Natal for giving me the golden opportunity to join the league of world class connoisseurs who are groomed at this university; Thank you!

My humble prayer is that, God will reward all your efforts. You will all reap the fruit of what you have sown in multiple folds.

Table of Contents

List of Abbreviations

ACO	Ant Colony Optimization
AI	Artificial Intelligence
ANN	Artificial Neural Network
APA	Average Prediction Accuracy
APWG	Anti-Phishing Working Group
AUC	Area Under the receiver operating characteristic Curve
BQPSO	Binary Quantum Particle Swarm Optimization
BP	Back Propagation
CA	Classification Accuracy
CERNET	China Education and Research Network
CSOACN	Clustering based on Self-Organized Ant Colony Network
DARPA	Defense Advanced Research Project agency
DGSOT	Dynamically Growing Self-Organizing Tree
DT	Decision trees
EA	Evolutionary Algorithm
E-Commerce	Electronic Commerce
E-Fraud	Electronic Fraud
EFuNN	Evolving Fuzzy Neural Network
EMD	Earth Movers Distance
FDS	Fraud Detection System
FFA	Firefly Algorithm
FFA_SVM:	Firefly based Support Vector Machines
FN	False Negative
FP	False Positive
FRFS	Fuzzy-Rough Feature Selection
GA	Genetic Algorithm
GB	Global Best
GW	Glow Worm
IDS	Intrusion Detection System

IG	Information Gain
IP	Internet Protocol
ISP	Internet Service Providers
LCS	Learning Classifier Systems
LDA	Latent Dirichlet Allocation
MAC	Media Access Control
ML	Machine Learning
MLP	Multilayer Perceptron
NB	Naïve Bayes
NI	Nature-Inspired Technique
NN	Neural Networks
O-SVM	Grid Algorithm Based SVM
PA	Prediction Accuracy
PADPS	Phishing Attack Detection and Prevention System
PENFF	Phishing Evolving Neural Fuzzy Framework
Pr	Precision
PSO	Particle Swarm Optimization
PSOAANN	Particle Swarm Optimization Trained Auto Associative Neural Network
RBF	Radial Basis Function
R	Recall
RF	Random Forest
SI	Swarm Intelligence
SSO	Simplified Swarm Optimization
SVM	Support Vector Machines
TN	True Negative
TP	True Positive
UCI	University of California, Irvine
URL	Uniform Resource Location
USA	United States of America

List of Figures

List of Included Articles

Articles accepted for peer review journal (ISI)

1. A. A. Akinyelu and A. O. Adewumi, "Classification of Phishing Email using Random Forest Machine Learning Technique," Journal of Applied Mathematics, vol. 2014, Article ID 425731, 2014.

Articles under review for peer review journal (ISI)

1. A. A. Akinyelu and A. O. Adewumi, “Application of Nature-Inspired Techniques to Electronic Fraud Detection: A Review”, Submitted to Artificial Intelligence Review.

List of Tables

Abstract

Electronic fraud is one of the major challenges faced by the vast majority of online internet users today. Curbing this menace is not an easy task, primarily because of the rapid rate at which fraudsters changes their mode of attack. Many techniques have been proposed in the academic literature to handle e-fraud. Some of them include: blacklist, whitelist, and machine learning (ML) based techniques. Among all these techniques, ML-based techniques have proven to be the most efficient, because of their ability to detect new fraudulent attacks, as they appear. There are three commonly perpetrated electronic frauds, namely: email spam, phishing and network intrusion. Among these three, more financial loss has been incurred owing to phishing attacks.

This research investigates and reports the use of ML and Nature Inspired technique in the domain of phishing detection, with the foremost objective of developing a dynamic and robust phishing email classifier with improved classification accuracy and reduced processing time**.** Two approaches to phishing email detection are proposed, and two email classifiers are developed based on the proposed approaches. In the first approach, a random forest algorithm is used to construct decision trees, which are, in turn, used for email classification. The second approach introduced a novel ML method that hybridizes firefly algorithm (FFA) and support vector machine (SVM). The hybridized method consists of three major stages: feature extraction phase, hyper-parameter selection phase and email classification phase. In the feature extraction phase, the feature vectors of all the features described in Section 3.6 are extracted and saved in a file for easy access. In the second stage, a novel hyper-parameter search algorithm, developed in this research, is used to generate exponentially growing sequence of paired C and Gamma (γ) values. FFA is then used to optimize the generated SVM hyper-parameters and to also find the best hyper-parameter pair. Finally, in the third phase, SVM is used to carry out the classification. This new approach addresses the problem of hyper-parameter optimization in SVM, and in turn, improves the classification speed and accuracy of SVM.

Using two publicly available email datasets, some experiments are performed to evaluate the performance of the two proposed phishing email detection techniques. During the evaluation of each approach, a set of features (well suited for phishing detection) are extracted from the training dataset and used to construct the classifiers. Thereafter, the trained classifiers are evaluated on the test dataset. The evaluations produced very good results. The RF-based classifier yielded a classification accuracy of 99.70%, a FP rate of 0.06% and a FN rate of 2.50%. Also, the hybridized classifier (known as FFA_SVM) produced a classification accuracy of 99.99%, a FP rate of 0.01% and a FN rate of 0.00%.

Chapter One

Introduction and Background

1.0 Introduction

Fraud can, simply, be defined as an act of deceiving people into revealing their personal information, essentially for the purpose of theft. Electronic fraud (e-fraud) refers to every form of criminal deception carried out using the internet[1]. E-fraud is a cybercrime that is usually facilitated by the use of a computer and a computer network [2]. Online fraud is commonly executed through emails. Most of the emails have an increased note of urgency in them, which usually requires users to either update or verify their account information immediately. Fraudsters pose as friends, relatives or authoritative figures making e-fraud very difficult to detect. Fraudsters, usually try to stay undetected in order to ensure maximum gains. They achieve this by disguising their activities and spontaneously changing their mode of operation. Norton cybercrime noted that e-fraud costs the global economy $388 billion a year [3]. Also, 14 adults, worldwide, fall victim to online fraud every seconds, resulting in over 1 million cybercrime victims yearly [3]. It is evident, there is an urgent need to secure the online environment and make it conducive and safe for e-business. This can be achieved by building an intelligent fraud detection system, capable of automatically adapting to spontaneous changes in fraud patterns**.** Many techniques have been proposed in the literature to handle e-fraud. Some of them include: blacklist [4], whitelist [5] and machine learning (ML) based techniques [6]). Among all these techniques, ML-based techniques have proven to be the most efficient, because of their ability to detect new fraudulent attacks as they appear. Furthermore, nature inspired (NI) techniques can be used to optimize the performance of ML-based techniques, especially, in the feature extraction stage and in the hyper-parameter optimization stage (for Support Vector Machine (SVM)), as used in, [7],[8],[9] and [10]. In this research, two improved fraud detection techniques are developed. The first technique is based on a ML algorithm (random forest (RF)) and the second technique is based on the hybridization of a ML technique (SVM) with a NI technique (firefly algorithm (FFA)).

[1] http://www.pcmag.com/encyclopedia/term/63509/e-fraud

1.1 Problem statement

One of the major threats faced by vast majority of online users and businesses is e-fraud. Since the emergence of electronic commerce (e-commerce) in 1994, e-fraud has evolved at a much faster rate [11]. Figure 1.1 [12] shows the increase in the number of phishing attacks from 2010 to 2013. The number of phishing attacks rose from 279,580 in 2011 to 448,126 (in 2013) leading to an estimated loss of about USD $5.9 million. Also, Figure 1.2 reveals that payment service industries and financial institutions are the two major target industries of fraudsters in recent times [13]. Due to the usefulness, effectiveness and lucrativeness of e-commerce, many organizations and individuals, today, now perform their business transactions online. The high rate of online business patronage is, undoubtedly, one of the major factors that have led to the rapid increase in online fraud. Consequently, many individuals and organizations, who have fallen prey in the past, now seek new ways of making their defence mechanism strong enough to prevent subsequent attacks. Behdad et al. [2] noted that a secured e-fraud defence system should be able to identify and also secure users from both existing and emerging attacks. Several traditional techniques have been used to build fraud detection systems, but many of these techniques are static, as they only have the capability to detect existing attacks. They are not capable of detecting unknown or emerging attacks. ML-based techniques are one of the few techniques suitable for developing fraud detection systems that can adapt to a changing environment. Their ability to detect both existing and emerging fraudulent attacks makes them suitable for this task. This research investigates and reports the use of ML and NI techniques in the domain of phishing detection, with the foremost objective of developing a dynamic and robust phishing email classifier, with improved classification accuracy and reduced processing time**.**

The research questions examined in this research are outlined thus:

- ❖ How can phishing email detection be greatly improved using Firefly based SVM and RF algorithm?
- ❖ What technique can be used to develop a fast and accurate phishing email classifier, capable of effectively detecting both known and emerging phishing attacks, with low FP, FN and high classification accuracy?

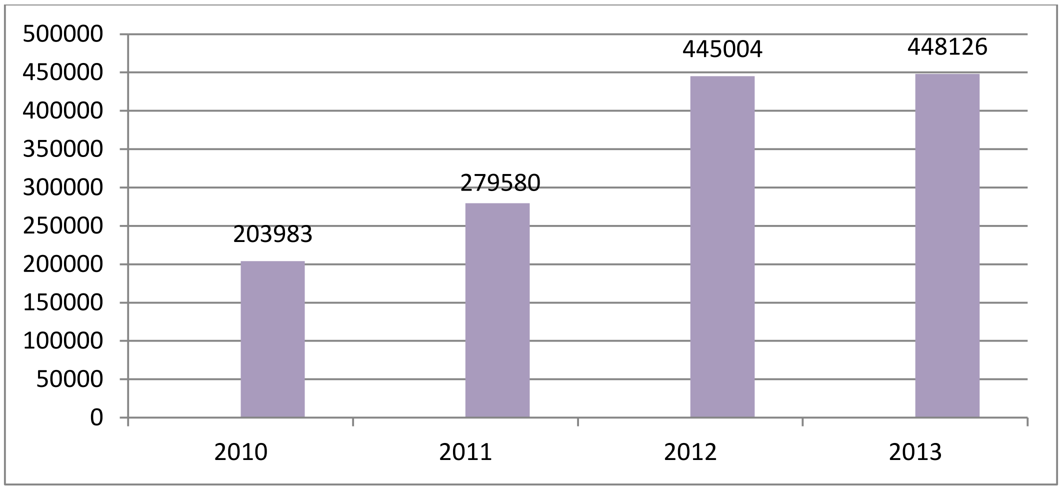

Figure 1.1: Phishing attacks from 2010 to 2013

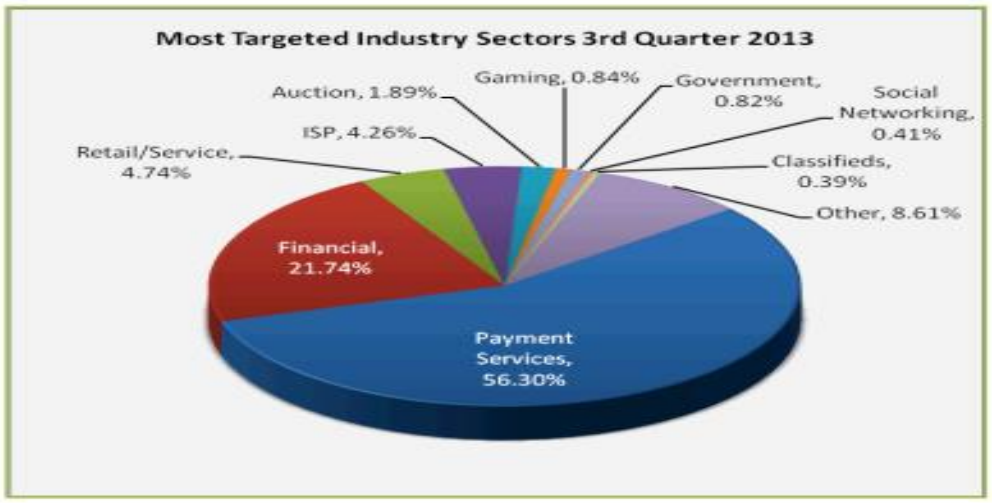

Figure 1.2: Target industries by phishers

1.2 Research objectives

The need for a robust online fraud detection system that can effectively and efficiently adapt to the changing patterns of phishing attacks cannot be overemphasized. Much work has been done by the research community to significantly reduce phishing attacks with great accuracy.

The major objectives of this research work are to:

- Identify the best set of features (or attributes) suitable for accurately classifying phishing emails.

- ❖ Develop an improved phishing email technique, capable of accurately detecting both known and emerging phishing attacks. Emphasis is on classification speed and accuracy.

1.3 Research methodology

This research tackles phishing detection using two different approaches:

- ❖ ML-Based technique (using RF algorithm)
- ❖ A NI-based ML technique. In this approach, a NI technique (that is, FFA) is hybridized with a ML algorithm (that is, SVM).

Two phishing e-mail classifiers are developed using RF algorithm, SVM and FFA. The first classifier is developed using the RF algorithm, and the second classifier is developed using both FFA and SVM. The performances of both classifiers are evaluated using a dataset collected from two online sources. The evaluations yielded better classification accuracy compared to proposed approaches reviewed in literature. More details are provided in Chapter 4.

1.4 Scope and limitation

This research focuses on three key areas:

- ❖ E-Fraud
- ❖ Machine Learning
- ❖ Nature Inspired Techniques

1.4.1 Electronic Fraud

According to Behdad et al. [2], there are three popular types of e-fraud perpetrated by scammers today, namely, phishing, email spam and network intrusion. Some other types of e-fraud include: business fraud, investment fraud, internet auction fraud, identity theft, advance fee fraud, credit card fraud, auction fraud, and overpayment fraud [14]. This research focuses on phishing. This is primarily because, the financial loss incurred by many individuals and organizations as a result of phishing attacks is much higher, compared to the other types of e-fraud. Additionally, much work has not been done on phishing detection compared to email spam and network intrusion.

1.4.2 Machine Learning

There are a number of techniques that can be used to handle phishing detection, including: blacklist approach, whitelist approach, heuristics-based approach, and data mining or ML-based approach [15]. Mahmoud et al. [15] carried out a literature survey of phishing mitigation techniques and discovered that the best performing phishing email classifiers used ML techniques. In the survey, the detection techniques proposed by Fette et al. [1] and Bergholz et al. [6] yielded the best result. Fette et al. [1] used RF and Bergholz et al. [6] used SVM. The excellent performance of the two proposed ML-based techniques made RF and SVM a suitable choice for this research. This research therefore focuses on RF and SVM.

1.4.3 Nature Inspired Techniques

Based on a comparative study in Table III of Behdad et al. [2], literature is very scare on the application of NI techniques to phishing email classification. Most phishing detection in literature have been based on blacklist [4] [16], whitelist [17] [18], visual similarity [5] [19], heuristic [20] [6]. There are several NI techniques that can be used, such as evolutionary algorithm (EA) [21] [22], swarm intelligence (SI) [23], artificial neural network [24]. This research focuses on one of the SI technique, namely, the FFA. FFA is a more recent SI technique that have not been explored in literature for phishing email classification.

The lack of availability of datasets for this study is the primary limitation. Due to privacy regulations, it is very difficult to obtain a dataset that contains emails in their original and consistent format. This work therefore have to depend on two reliable publicly available dataset, SpamAssassin [25] and Monkey.org [26].

1.5 Contribution to knowledge

The core contributions of this research work are as follows:

1. Introduction of a novel NI based ML technique (FFA_SVM). In FFA_SVM, two algorithms are hybridized. FFA and SVM. This new technique consists of three phases. The first stage involves feature extraction. In the second phase, FFA is used to find the optimized hyper-parameters for SVM (Radial Basis Function (RBF) kernel), using 5-fold

cross validation. The optimized parameters are then used to build a SVM model for classification. Finally, SVM is used to perform the classification.

2. Implementation of a search method for SVM hyper-parameters. Hsu et al.[27] suggested that the best way to identify the optimized C and Gamma values is to try using an exponentially growing sequence of C and Gamma. To the best of our knowledge, this method has not been explored in the domain of phishing email filtering. In this research, a novel algorithm is developed, based on the hyper-parameter search technique, proposed by Hsu et al.[27].
3. Development of two phishing email filters based on SVM, FFA and RF algorithm. Experimental results showed that the two phishing email classifiers compare fairly and better with some results in literature.

1.6 Outline

The rest of this dissertation is structured as follows:

Chapter two gives a background knowledge on some selected commonly used NI techniques and Artificial Intelligence (AI) techniques that have been applied successfully to e-fraud detection. Furthermore, the chapter presents a literature review on some AI-based techniques and NI-based techniques that have been applied successfully, as noted in the literature, to tackle three commonly perpetrated e-frauds – email spam, phishing and network intrusion. Also, other ML techniques that been applied to e-fraud are summarized. Finally, some available methods that can be used to improve the performances of NI techniques are outlined in Chapter two.

Chapter three gives a description of the two techniques proposed in this research (that is, the RF-based technique and the firefly based SVM technique). Also, the features used by the two proposed techniques in this research are described in Chapter three. Chapter four provide details about the set up used for all the experiments performed in this research. It also provides details about the dataset and the performance measure used for all the experiments. Additionally, Chapter four presents the experimental results obtained from the tests carried out on the RF-based technique and the firefly based SVM technique respectively. The method used to evaluate the performance of the techniques proposed in this research is also explained in Chapter four. Finally, the summary and conclusion is presented in Chapter five alongside with some recommendations on some techniques that can be explored in the future.

Chapter Two

Literature Review and Background information

2.0 Introduction

This chapter gives an overview of some basic concepts related to our work. It then gives a review of e-fraud as well as some AI-based techniques proposed by researchers in handling various forms of e-fraud. Initially, basic concepts that provide overview of NI and ML techniques are presented alongside with a general introduction to e-fraud.

2.1 Basic Concepts

Some basic concept on NI techniques, ML, ANN and email filtering are discussed next.

2.1.1 Nature Inspired techniques

Nature has an easy and amazing way of solving complex problems. It is the source of inspiration to many techniques (referred to as NI techniques), such as the NI techniques listed in Section 1.4.3. NI techniques are essentially AI techniques that are inspired by the approach nature adopts in solving its problems [2]. For example, Particle Swarm Optimization (PSO) is inspired by the emerging behavior of bird flocking or fish schooling [28]. FFA is inspired by the flashing lights of fireflies [29].

NI computing is an emerging computational model for handling complex real world classification and optimization problems such as: e-fraud [30], traveling salesman problem [31], graph coloring [32], parking space problem [33], hostel allocation problem [34]. NI techniques and their applications are prominent in literature, very likely due to some inherent intriguing features they present including:

1. **Robustness**: Optimization problem can still be resolved even when many uncertainties (such as noise in the environment) are present in the input data [35] [36].
2. **Versatility**: NI techniques have the capability of searching through a large database of solutions thereby making it suitable for solving any problem with less dependence on the domain knowledge [37].

3. **Adaptability**: NI techniques, as opposed to other AI methods, can be used to deal with problems that have changing patterns such as e-fraud. This is because this technique takes advantage of the dynamic problem-solving capability of nature.
4. **Specialization**: In NI techniques, the population is divided into different groups, and various tasks are assigned to each of them alongside with specialized soldiers to secure them. For example, XCS, a type of learning classifier system (LCSs), make use of a niche-based technique when it want to conserve its population of rules [38].

2.1.2 Machine Learning

Arthur Samuel (1959) defines ML as a "field of study that gives computers the ability to learn without being explicitly programmed". Similarly, Tom Mitchell (1998) defines ML as "a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E". Furthermore, Ayodele [39] noted that learning involves finding hidden statistical regularities or patterns within data. ML algorithms are popularly used to solve problems involving automatic classification of data (such as e-fraud detection) [6]. They have the ability of analyzing data contents and extracting hidden patterns from them. The primary aim of ML-based systems is prediction, that is, prediction of known patterns that were learned from the training data [40].

There are several types of ML algorithms which can be grouped into different classes based on their mode of learning as outlined below [39]:

1. **Supervised learning:** Supervised learning involves finding unknown patterns in labelled data. Algorithms that belong to this class include: SVM, Naïve Bayes (NB) Classifier, Logical Regression, Neural Network (NN), Bayesian networks, RF, K-Means clustering, etc. These algorithms can only be used when the data available are well labeled, that is, when the desired output is known.
2. **Unsupervised learning:** Unsupervised learning involves finding concealed patterns in unlabeled data. Some algorithms that can be used to handle unsupervised learning problems include: hidden Markov models, mixture models, K-means, etc.

3. **Semi-supervised learning:** This involves finding patterns in a dataset consisting of both labeled and unlabeled instances; it falls between supervised learning and unsupervised learning.
4. **Reinforcement learning:** Reinforcement learning is concerned with how software agents learn through try-and-error interactions with their environments [41]. Unlike other types of ML, in reinforcement learning, the software agent is not told what to do or what actions to take, the agents try different actions and embark on the actions that yield the best reward [42].
5. **Transduction:** Algorithms in this category are similar to supervised learning. Here, outputs are predicted based on training inputs, training outputs and new inputs [39].
6. **Learning to learn:** In learning to learn, actions are taken based on some set of assumptions (referred to as inductive bias) the algorithm has learnt, from past experience [39].

2.1.3 Artificial Neural Network

Artificial Neural Network (ANN) is a ML method that is inspired by the central nervous system of humans (especially, the brain) [43], in that, ANN tries to electronically simulate the human brain. ANN is made up of many artificial neurons interconnected with each other. The number of neurons in the network depends on the problem that is to be tackled. Inputs (having associated weights) are fed into the network through these neurons. The associated weights of these inputs are generally adjusted to train the network. ANN is divided into two: feed-forward networks and recurrent/feedback network [24]. Furthermore, feed-forward network is divided into three: single-layer perceptron, MLP and RBF nets [24]. Also, feedback network is divided into four, namely: competitive networks, kohonen self-organizing network, Hopfield network and ART models [24]. The most popular type of NN in use today are the MLPs [44]. A typical MLP NN consist of one input layer, one output layer and one or more hidden layers [44]. ANN has been successfully used to solve complex real world problems. Some of the applications of ANN, with respect to network intrusion detection, phishing detection and email spam detection are discussed in Section 2.4.

2.1.4 Email Filtering

Email filtering most often refers to the process of automatically separating unsolicited emails from legitimate emails [45]. This is normally carried out by an email filter. Phishing attacks are typically perpetrated via emails. These emails usually contain social engineering messages (with specific phrases) that demand users to perform specific actions (such as clicking on a URL). Therefore, the content of these emails are useful features for phishing detection. Very few phishing email filters have been developed, in contrast to many existing email filters that have been developed for spam emails. Many of the existing email filters used several phishing detection techniques, ranging from blacklists [4], visual similarity [19], heuristic [20], and ML [46]. Of all these techniques, the ML-based technique achieved the best results.

2.2 Electronic Fraud and its Detection

E-fraud can simply be defined as deceitful actions carried out with the aid of electronic devices [2]. E-fraud is becoming highly lucrative, because of the high rate of patronage by fraudsters. Millions of dollars have been lost by many individuals and organizations. For example, the loss incurred globally from credit and debit card transactions as at August 2013 add up to $11.27 billion [47]. In another example, in [48], the Financial Fraud Action UK (FFA UK) revealed that the loss suffered by UK card holders was summed up to £450 million, which is 16 percent higher than the loss incurred in 2012, which was £388.3 million. This is quite alarming. Behdad et al. [2] pointed out that it is practically impossible to handle e-fraud without automation, because of the complexities involved. In order to build a robust fraud detection system, a lot of important factors, as discussed in Section 2.2.1, must be taken into proper consideration.

2.2.1 General Challenges in e-Fraud Detection

E-fraud detection is not an easy task. There are very few ready-made fraud detection solutions [49], because of some challenging peculiar properties faced by fraud detection [2]. Some of the general challenges faced in the detection of e-Fraud include:

1. **Experience Imbalance:** For NI techniques to work effectively they must be trained with a set of well-balanced data (that is, the samples in the minority class must be equal to the

samples in the majority class). Unfortunately, real world data are usually imbalanced [2]. These imbalances have a negative effect on the performance of a fraud detection system.

2. **Online Learning:** The classification accuracy of a fraud detection system depends on the volume of data fed into it during the training phase. Lack of complete or adequate data from the onset will have an adverse effect on the learning process [2]. The fraud detection system will have to make its decision based on the available data and at the same time, adapt as fast as possible as it encounters incoming data [2].
3. **Adaptive Adversaries:** Fraud detection is, increasingly, becoming more difficult because of the rapidly changing patterns used by fraudsters in perpetrating fraud [50]. This is a major challenge, because the module responsible for detecting fraud may become obsolete over time, implying that the module will not be capable of handling new attacks [2].
4. **Concept Drift:** The statistical properties of a target variable, which a model tries to predict, change over a period of time in unpredicted ways. This is a challenge, because fraudsters operate using different patterns, often implying that a pattern indicating non-fraud today may be used by a fraudster tomorrow, thereby, reducing the predictive accuracy of the system [51]. E-fraud systems should be sensitive enough to detect patterns (new or evolving concepts) accurately, such as the change in the buying pattern of a customer due to inflation [2].
5. **Noise:** Noise refers to errors in the training data used by ML algorithms. Developing a robust e-fraud detection system involves the use of a large volume of data. According to an online resource, the likelihood of error existence in a large dataset is very high. The presence of noise in a dataset will reduce the learning rate of a detection system, because the system will need to compensate for the introduced noise. Markus and Myers pointed out that noise can be generated randomly, distributed uniformly or non-uniformly and can also be created by fraudsters [52]. A fraud detection system should be robust and sensitive enough to handle noise and concept drift accurately [2]. As suggested by Behdad et al. [2], some major challenges faced by e-fraud systems with regards to noise include:
   i. Distinguishing between noise introduced randomly or noise maliciously created by fraudsters, which is sometimes similar to random noise, and

ii. Segregating between noise and concept drift. The smallest amount of noise can be very harmful, because the system's prediction rate should be as accurate as possible.

6. **Unequal Misclassification Costs:** In fraud detection, the method used for classification of errors has its own cost implication on the e-fraud system which, usually, has unequal misclassification costs. Examples of misclassification cost include, False Positive (FP) error cost (e.g. resources wasted investigating non-fraud) and False Negative (FN) error cost (e.g. the expense involved in failure to detect fraud) [53]. These misclassification cost differs from example to example and it can possibly change with time. There is no doubt that this will have an adverse effect on the learning problem in question.
7. **Fast Processing and Large Volume of Data:** Some e-fraud detection systems require real time detection, implying that the system's speed is of the utmost importance [2]. Processing of large volumes of data will lead to a prolonged processing time.

Basically, all the challenging properties of fraud highlighted in Section 2.2.1 need to be taken into proper consideration, if we want to build a robust fraud detection system that is sensitive to fraudulent attacks in a changing environment.

2.3 Prominent Types of Electronic Fraud

There are three prominent types of fraud, namely: email spam, phishing and network intrusion [2]. An introduction to these three types is given in the next three subsections.

2.3.1 Email Spam

Generally, email spam is defined as unsolicited bulk email [2]. It is, usually, addressed to hundreds or thousands of recipients. Gao [54] further explains that spam emails usually meet the following three criteria below:

1. Anonymity: The address and the identity of the email sender are not revealed.
2. Mass Mailing: The email is sent to a mailing list containing the addresses of a vast group of people.
3. Unsolicited: The recipients in the mailing list have not granted confirmable consent for the email to be sent.

There are quite a number of techniques that can be used to handle the effect of email spam, such as user awareness, disciplinary measures, anti-spam software and email filter. Filtering is one of the widely used techniques today. Filtering can be defined as the automatic grouping of emails (both incoming and outgoing) into spam mails and legitimate mails [55]. Duan et al. [55] suggested that a spam detection system can be implemented at various levels. It can be implemented at the router's end of email transmission, at the destination mail server, and also at the destination mailbox. Meanwhile, Duan et al., pointed out that filtering does not solve spam as a problem totally, because unsolicited mails are still delivered. It only separates junk mails from legitimate mails, thereby, saving time for users [55]. Fawcett [50] views spam filtering from the optic of data mining and outlined some of its challenges as follows:

1. **Skewed and drifting class distributions:** The distributions of messages (both spam and legitimate mails) between users are unequal. It varies between individuals.
2. **Unequal and uncertain error costs:** Sometimes, a spam detector might not be very accurate when filtering its messages. Errors committed can be very critical, because, for some users, every individual message is very important; a single missed message can be very costly.
3. **Disjunctive and changing target concept:** "Spam" as a concept which changes over time. It is difficult to determine how much the change will be at a certain point in time, partly, because patterns adopted by spammers change over time.
4. **Intelligent adaptive adversaries:** Spammers are improving in their filtering-evasion skills on a daily basis; they now employ new sophisticated techniques that some filtering system may not detect.

2.3.2 Phishing

Phishing is an act that attempts to electronically obtain delicate or confidential information from users (usually, for the purpose of fraud) by creating a replica website of a legitimate organization. Phishing is, usually, perpetrated by sending deceitful and well composed emails to users. These emails, usually, contain links to cloned websites, and clicking on this links may re-direct users to a phishing website or a malware hosting website. Malware hosting websites are, usually, infected with malicious codes that can gain access to private information of users and also cause damages to users' computers. Due to the vast number of email messages received by various users today, separating legitimate emails from phishing emails is a challenging task, therefore, the need for a quicker, robust and effective filtering technique cannot be overstated. Several approaches have been proposed in the literature, including: network-based approach, blacklist, whitelist and content-based approach. Network-based approaches are costly to implement, difficult to maintain and time consuming [56]. Blacklist (that is, list of reported phishing websites) and whitelist (that is, list of target companies) approaches yield high FP and FN rates; their effectiveness is limited to the information stored in them. This limitation makes blacklist and whitelist incapable of automatically detecting new phishing attacks as they occur. The Anti-Phishing Working Group (APWG) noted that the average uptime for phishing a website is 44.39 hours (that is, less than 2 days). Content-based approach aims at capturing the content and structural properties of a data. According to White et al. [57], the blacklist approach is the widely used phishing detection approach adopted by many today. Nevertheless, Bergholz et al., pointed out that a content-based technique is the most accurate and secure of all the phishing detection techniques mentioned above [46]. This is because, the content-based technique is capable of discovering new fraudulent patterns in large datasets as they evolve. Phishing is a classification problem and Martin et al. [58], outlined five stages involved in phishing attacks:

1. **Planning**: At this stage, plans on who the target organization should be and how to get the email address of the organization's customers are determined.
2. **Setup**: Here, the method for sending the messages (usually mass mailing) and obtaining the revealed users information is devised.
3. **Attack**: At this stage, the fraudulent and deceptive message is sent out to users' addresses.
4. **Collection**: Here, the information of the victimized users' are captured.
5. **Attack**: At this stage, the actual fraud is committed using the captured information revealed by users at the collection stage.

There are several approaches that have been applied to phishing detection. Adida et al. [59] suggested that phishing can be tackled and eliminated at the email level, since many scammers use email as their tool for committing fraud. Dhamija and Tygar [60] also suggested that email can be eliminated at the website level. They proposed that a security toolbar may be incorporated into web browsers. Another approach proposed by Dynamic Security Skins [61] involves the use of visual hash. In this approach, visual hash was generated randomly and used to customize the web and windows form of a browser. Visual hash is responsible for identifying websites that have been authenticated successfully by the browser. Buntine also proposed a method called Cryptographic identity verification [62]. The author pointed out that this method can only work if the entire web infrastructure (both servers and client) is changed. In addition, increasing the awareness of users can increase mitigation against malicious attack; users should be well trained on various ways of identifying phishing website.

Khonji [15] summarizes the approaches that can be applied to tackle phishing attacks into four categories, namely: offensive defence, correction, prevention and detection approach, respectively.

2.3.2.1 Offensive Defence Approaches

The objective of approaches that fall in this category, is to neutralize the effect of the phishing attack. This method is, mostly, applicable to users that have already fallen victim to the attack (that is, users that have already filled out and submitted their private information into the HTML forms of the phishing website). In this approach, whenever a user is misled to a phishing website, a software installed on the user's browser will also submit several fake samples of information to the phishing website, so that it will be difficult for the attackers to locate the actual information submitted by the user.

2.3.2.2 Correction Approaches

Approaches in this category aim at, either, removing the phishing files from the website or making the phishing website inaccessible. Both can be achieved by alarming the internet service provider that hosted the website, in order that they will carry out the appropriate or required action.

2.3.2.3 Prevention Approaches

These approaches aim to both prevent users from falling victims and to stop phishers from defrauding users in the future. The latter can be achieved by involving law enforcement agencies. These agencies can carry out their investigation and penalize these attackers by making them pay dearly for their crimes. This serves as a deterrent and, in turn, minimizes subsequent attacks.

2.3.2.4 Detection Approaches

The primary focus of approaches in this category is in identifying phishing attacks and classifying them into legitimate and illegitimate. This is usually achieved by examining hundreds of possible phishing features in every email and automatically filtering them. The examination of phishing features enables the detection system to be capable of adapting to new phishing attacks as they occur. Detection approach can be further classified into four approaches, namely, blacklist detection approach, whitelist detection approach, network-based detection approach and content-based detection approach. These four approaches are discussed, briefly, below.

Blacklist Detection Approach

Blacklist refers to the list of reported phishing websites. The blacklist approach is one of the approaches adopted by some internet service providers (ISP), web browser providers and some email providers (such as Gmail, yahoo, Microsoft, etc.) in order to identify blacklisted phishing addresses. These providers use the information on the blacklist to secure their system and, in turn, shield their users from falling victim to phishing attacks. For example, if an email is sent from an IP address that is already blacklisted, the email provider can either block this email from been delivered or send the email to the spam folder of the recipient. A blacklist usually contains domain names and IP addresses of previously detected phishing websites. Some blacklists also contain keywords, IP addresses of open proxies and relays, IP addresses of ISPs that host phishing websites and RFC violators (IP addresses that violate the internet and network engineering standards). Almomani et al. [63] reported that there are more than 20 spam blacklists commonly used today and these blacklists are, usually, updated at regular intervals; for example, the blacklist of Firefox browser (stored in the user’s profile) is, usually, updated every 30 minutes [64].

Whitelist Detection Approach

A whitelist refers to the list of targeted companies (such as eBay, Paypal, Visa, etc.). The whitelist approach is very similar to the blacklist approach; both of them are used to secure users from fraudulent attacks. The major difference between a whitelist and a blacklist lies in the information contained in both of them. A whitelist contains email addresses, IP addresses and domain names that are considered to be spam free. Generally, various providers use a whitelist to inform of their filtering decision. For example, the network admin for an organization can decide to set up a whitelist of Media Access Control (MAC) addresses and then use it to control access into their network. Also, some spam filters keep a whitelist of email addresses, IP addresses and domain names and use this list to decide whether an email is legitimate or not.

Network-based Detection Approach

This approach is used by various network administrators to secure their network from intrusion. Generally, when a user sends a message over the network, it is formatted into a smaller unit called packet which contains the message sent by the user and the IP address of the sending network. However, the IP address can be faked in such a way that it will be hidden. Generally, network-based approach aims at blocking any network packet that is deemed to be illegitimate (that is, packets that contain disguised IP addresses).

Content-based Detection Approach

The content-based approach is another method that can be used to detect fraudulent attacks. This approach involves analyzing the content and structural properties of the data. For example, Microsoft Internet Explorer (Version 7) has an inbuilt classifier, that analyzes the contents of web pages and filters them, based on some criteria [6]. Bergholz et al., noted in [46] that the content-based approach is the most effective and secure of all the filtering approaches, even though [57] also noted that the black-list approach is the most widely used approach.

Quite a number of phishing detection systems have been developed, based on the four approaches discussed above. Network-based approaches are, generally, costly to implement and time consuming [56]. Blacklist and whitelist approaches, usually, yield high FP and FN rates, because their effectiveness is only limited to the information stored on the blacklist (or the whitelists), thus, making them incapable of detecting new and emerging phishing attacks. Khonji et al. [15] described the classification performance of some of these detection techniques and outlined their limitations. Some of these techniques and their limitations are outlined in Table 2.1. Based on some case studies conducted by Aburrous et al. [65], some criteria and indicators that can be used for phishing detection were collated. A full list of these indicators and criteria are outlined in Table 2.2. One of the methods for effectively dealing with phishing emails is in identifying and preventing these emails from entering into users' mailboxs. E-mail filters can be used to achieve this. Also, web browsers can be well equipped with mechanisms for detecting and blocking illegitimate addresses (that is, phishing websites) from going through. If all these can be achieved, then phishing activity, as a whole, will be reduced to a negligible minimum.

2.3.3 Network Intrusion

Network intrusion involves several related actions occurring at different parts of the network. The state of the network changes when there is occurrence of several related actions in different parts of the network. In intrusion detection, the events that occur in the network are, usually, monitored and analysed for security issues. An intrusion detection system (IDS) is responsible for preventing any action in the network that attempts to compromise the integrity, confidentiality and availability of a certain resource, and it is also responsible for strictly monitoring any illegal access to the network's resources [2]. Signs of intrusion can be detected by examining the log files and the traffic of the network [2]. Modern IDS adopts both host based and network based approaches [2], that is, the IDS monitor both system logs (host-based approach) and the flow of network packets (network-based approach). Generally, there are two basic approaches that are used when building a network intrusion system, namely: misuse detection and anomaly detection. The misuse detection approach focuses on protecting the network from known network threats and system vulnerability. Anomaly detection aims to search for events in the network that deviate from the normal behaviour of users, hosts or network connection.

2.4 Application of Nature-Inspired Technique to e-Fraud

Generally, one of the major challenges faced by AI techniques (with centralized control) is the ability to adapt to changing environment [2]. NI techniques have been able to overcome this challenge. NI techniques are inspired by the dynamic approach used by natural systems in achieving their various tasks. Behdad et al. [2] clarified that the objective for using these approaches is because natural systems (such as organs in an animal's body or a group of animals or insect) can adapt easily to changing environment and they also have the ability to recover quickly from adverse conditions. Some NI techniques have been proposed in the literature for detection of email spam, network intrusion and phishing as summarized in the following subsections. We present various approaches under the different types of NI techniques used.

2.4.1 Application of ANN for e-Fraud Detection

Some existing ANN-based e-fraud detection techniques that have been proposed in literature to handle email spam, phishing and network intrusion are discussed in this subsection.

2.4.1.1 ANN for Email Spam Detection

Zhan et al. [66] developed a spam filtering system based on the back propagation NN algorithm (one of the most successful and widely applied NN models). The filtering system automatically generates new rules from incoming messages, thereby, capable of automatically detecting emerging spam features or patterns. The system was developed in conjunction with other filtering mechanism (such as blacklisting and Bayesian filtering) in order to reduce the occurrence of the "false negatives" rate and "false positives" rate. Zhan et al., tested the system on an English dataset (acquired from SpamAssasin) and a Chinese dataset (acquired from China Education and Research Network (CERNET) Computer Emergency Response Team). They used 1,000 emails each, from both datasets and compared the performance of the system to that of SpamAssasin, a widely used, robust, heuristic-based, spam filtering system that combines different spam detection techniques. The result of the experiments showed that the filtering system performed better in terms of classification accuracy, FN and FP rate, compared to SpamAssasin. The experiment yielded an accuracy rate of 98.5% and very low FP and FN rates. Similarly, Yang and Elfayoumy [67] performed a number of experiments to evaluate the performance of both feedforward back propagation NN and Bayesian classifiers. They developed two different algorithms, based on NN and Bayesian classifiers, respectively, and compared their performance. The result of the comparison revealed that even though training a NN takes a longer period of time, neural networking still has high tolerance to noisy data and also has the ability to classify new emerging patterns. Elfayoumy carried out some tests using two different groups of network topologies. The result of the tests point to the following:

1. The number of layers in a NN has a negligible impact on the performance of the network.
2. The number of neurons in both hidden and output layers does not reduce the error rate of the network.

3. The number of output neurons in the output layer does not have a reasonable impact on the performance of the network.
4. Large numbers of nodes are difficult to train and they require a large amount of memory and execution time.

The above indications clearly reveal that the number of nodes or hidden layers in a network do not have a huge impact on the performance of the network. It is, therefore, advisable to avoid networks with large numbers of layers, nodes or neurons.

Table 2.1: Phishing detection techniques and their limitations

Detection Techniques	**Limitations**	**References**
PhishNet made use of blacklist and heuristics	High FP rate of 5% and a FN rate of 3%	[4]
SpoofGuard made use of heuristics	Very high FP rate of 38% and FN rate of 9%	[68]
PhishCatch made use of heuristics	It cannot be adopted in a dynamic environment because it was developed using manually hard-coded rules. It also has a very high FN rate of 20%	[69]
PhishWish made use of blacklists and heuristics.	It has a similar limitation to PhishCatch as it was also hard-coded with 11 rules. It also has a high FP rate of 8.3%	[16]
CANTINA made use of heuristics and network communication	High FN rate of 11%.	[70]
Visual similarity-based phishing detection made use of blacklist, whitelist and visual similarities	Very high FP rate of 17.4% and FN rate of 8.3%. There is also a possibility of delay in the browser due to image processing.	[5]

Note: FP refers to False Positive; FN refers to False Negative

Table 2.2: Phishing indicators [65]

Criteria	Phishing indicators	
URL & Domain Identity	1	Using IP address
	2	Abnormal request URL
	3	Abnormal URL of anchor
	4	Abnormal DNS record
	5	Abnormal URL
Security & Encryption	1	SSL certificate
	2	Certificate authority
	3	Abnormal cookie
	4	Distinguished names certificate
Source code & Java script	1	Redirect pages
	2	Straddling attack
	3	Pharming attack
	4	OnMouseOver to hide the link
	5	Server form handler
Page style & Contents	1	Spelling errors
	2	Copying website
	3	Web Forms with Submit button
	4	Pop-up windows
	5	Disabling right click
Web address bar	1	Long url address
	2	Replacing similar char for URL
	3	Adding a prefix or suffix
	4	Using the @ symbol to confuse
	5	Hexadecimal char Codes
Social-human factor	1	Emphasis on security
	2	Public generation salutation
	3	Buying time to access accounts

2.4.1.2 ANN for Phishing Detection

Phishing attacks are difficult to detect, because of the sophisticated techniques used by phishers. Much work has been undertaken by researchers in the domain of ML. Zhang & Yuan [71] proposed a phishing detection algorithm, based on ANN. They applied the most common structure of a multilayer feed forward NN (consisting of one input layer, one hidden layer and one output layer). Zhang & Yuan pointed out that the feedforward NN is better at modelling relationships between inputs and outputs, compared to other regular (or frequently used) NN approaches. Zhang & Yuan implemented their algorithm on a dataset consisting of 4,560 phishing email and 4,202 legitimate e-mails. They compared the performance of the proposed ANN-based algorithm to other NI algorithms, applied in the domain of phishing email filtering, and found it to be more suitable. Zhang & Yuan further noted that, even though their technique requires more time for parameter settings, it performed better compared to other classifiers. In another work, Almomani et al. [63] developed an algorithm called “Evolving Fuzzy Neural Network” (EFuNN). This algorithm, adopts the same technique used by NN. Almomani et al. explained that the major difference between EFuNN and NN is in the nodes creation. The number of nodes in EFuNN is created during the learning phase. Almomani et al., introduced a new framework, based on their proposed algorithm, known as Phishing Evolving Neural Fuzzy Framework (PENFF). The framework consist of five layers, namely: Input layer, Fuzzy output layer, Rule nodes layer, Fuzzy input layer and Output layer. Almomani et al., stated further that the framework is capable of predicting and also detecting “zero-day” phishing e-mail. “Zero-day” phishing emails are emails that contain links to phishing websites that are not yet discovered, that is, websites that are not yet included in the blacklist of known phishing emails. Finally, Almomani et al., tested their algorithm on a dataset consisting of two classes. The first class consists of 2,000 phishing emails (acquired from the monkey website [26]) and the second class contains 2,000 ham emails (from spamassassin [72]). The result of their test showed that PENFF has the ability to detect phishing emails with a reduced error rate. Furthermore, Almomani et al., noted that their proposed framework can also be implemented in various learning systems. Martin et al. [58] proposed a new model that uses ANN to classify and predict phishing websites. The model has an embedded algorithm, responsible for updating the weights in the output and hidden layers of the NN. In their paper, Martin et al., explained how weights can be updated in a multi-layer network. Martin et al., planned to test their algorithm on two publicly available datasets, provided by Jose Nazario [26]. Martin et al., believed that their framework works better and gives a lower error rate, compared to some other data mining classification algorithms mentioned in the literature.

2.4.1.3 ANN for Network intrusion Detection

Shun and Malki [73] proposed a NN-based IDS. The network architecture of their proposed NN consists of, one input layer, one hidden layer and one output layer. The input layer consists of several nodes (determined by the number of input data set), the output layer contains two nodes (one node to indicate normal traffic and the second node to indicate malicious traffic). Shun and Malki trained and tested their network on a dataset obtained from the repository of the Defense Advanced Research Project Agency (DARPA). The dataset was classified into three: normal traffic, known attack and unknown attack. Shun and Malki carried out four different tests on the trained network and claimed that all the results yielded a 100% classification for both normal traffic and known attack and 76% for unknown attack, indicating that a NN is an effective method that can be used to secure a network from both current and future intrusions. Kachurka and Golovko [74] developed a prototype for an IDS, based on the NN algorithm. Kachurka and Golovko noted that the prototype is capable of accurately detecting both known and unknown attacks in real-time with low FP and FN rates. The system combines both anomaly detection techniques and misuse detection techniques (in contrast to most IDS methods that use only one of these techniques). The two detectors (anomaly detector and misuse detector) operate in parallel mode. Kachurka and Golovko performed a number of experiments on a real computer network using KDD data. They trained their network with a dataset containing 500 different connections and, thereafter, launched several attacks on the network to test its effectiveness. The result of their test (involving about three different threshold settings for the network detectors) indicated that their proposed IDS prototype is capable of handling both anomaly and misuse detection in real time.

2.4.2 Application of Evolutionary Algorithm for e-Fraud Detection

The EA, a subset of evolutionary computation, is a generic population-based metaheuristic optimization algorithm that uses a mechanism inspired by natural biological processes such as, recombination, selection, mutation and reproduction. Algorithms in this class apply the principle of survival of the fittest to operate on a population of individuals and, based on this principle, they produce improved results. Different set of solutions are produced in different generations and new individuals are selected based on their fitness value. The new set of evolved solutions (or individuals) after each generation are usually more fit compared to the individuals they evolved from [75]. The EA can be used to solve complex real world optimization problems that evolve overtime (such as fraud detection). Some of its applications are discussed in the next four sub-sections:

2.4.2.1 EA for Email Spam Detection

Dudley et al. [76], developed a multi-objective EA that can be used for spam filtering. Dudley et al. explained that the algorithm is an improvement over SpamAssasin [76]. SpamAssassin detects spam messages by, firstly, carrying out a series of test (of different types) on an email. It then multiplies the result of each test by a pre-assigned weight, automatically derived by a 3-stage process, adds the resultant values together, and then classifies the message appropriately, based on whether the sum is greater or smaller than a user-defined threshold [76]. According to Dudley et al., in SpamAssasin, users only have the privilege to set their preferred threshold (before the entire tests are carried out), but they do not have the power to assign weight to an individual test if they want to. The multi-objective EA improved on this limitation. The algorithm was designed in such a way, that it provides several ranges of setups (with different sets of weights), giving users the liberty to switch between setups at will. This implies that users can now choose between setups that have their own preferred weight at every point in time. Dudley et al., ran a number of experiments to evaluate the algorithm's performance, and the outcome was favourable. In another work, Sanpakdee et al. [77] proposed a mechanism for filtering spam emails using genetic algorithm (GA). In the mechanism, the email content was first extracted, and then divided into seven different groups (Adult, Financial, Commercial, Beauty & Diet, Travelling, Home-Based Business and Gambling) based on the relative meaning of the extracted words. A string of chromosomes comprised of the seven different groups (or genes), is then composed. The weight of each group (represented in the string of chromosomes) is calculated (using a formula explained in [77]) and then encoded into binary values. The binary values are fed into GA for fitness evaluation (by a defined fitness function), and then the fittest chromosomes are selected and used to create varieties of spam mail prototypes (that is, a set of rules). These prototypes are used for spam filtering. Sanpakdee et al., tested the mechanism on 1,097 spam mails and 300 hams (collected from Enron Email Data set [78]). The test yielded an accuracy of 85%. Sanpakdee et al., pointed out that the system can also dynamically generate new spam mail prototypes.

2.4.2.2 EA for Phishing Detection

Shreeram et al. [79] proposed a GA-based Phishing Attack Detection and Prevention System (PADPS). Shreeram et al., tested the system on a set of data that was collected from the APWG. Firstly, the dataset was analyzed and then fed into GA for fitness evaluation. After evaluation, a set of rules was formed and stored in a database used by the phishing detection system. Shreeram et al., noted that the GA-based system is capable of detecting both phishing emails and malicious links on web pages.

2.4.2.3 EA for Network intrusion Detection

Gong, et al. [80] developed a network intrusion detection software based on GA. Gong et al., focused on network misuse detection. They used GA to generate a set of classification rules from network audit data. The classification rules were then used to detect incoming network connections in real time. Gong et al., tested their software on a selected subset of the 1998 DARPA dataset, and the test yielded good results. Gong et al., noted further that the GA-based software is capable of detecting network intrusions and also classifying network misuse detection in real time. In a similar work, Crosbie and Spafford [81] developed a robust IDS based on genetic programming. The IDS consist of multiple functional units (working independently) encapsulated as autonomous agents. These autonomous agents are responsible for detecting anomalous behaviour (e.g. repeated connection attempts, connections to privilege ports, etc.) in the network under continuously changing conditions. The agents were divided into groups and trained. The group with the best known agent was then selected and used by the IDS. Crosbie and Spafford trained and tested the GA-based IDS on a dataset that contained both intrusive and non-intrusive scenario data. The evaluation yielded positive results. This implies that genetic programming is a useful tool for developing IDSs (consisting of autonomous agents), capable of detecting both known and unknown network connections.

2.4.3 Swarm Intelligence for e-Fraud Detection

SI is a discipline that deals with some social living beings (belonging to natural and artificial systems) that organize themselves using two remarkable features: decentralized control and selforganization [82, 83]. SI focuses primarily on the combined behavior of these individuals resulting from their interactions with themselves and with their environment. Examples of some social living beings that falls into the domain of SI include: colonies of ants and termites, school of fish, flocks of birds, herds of land animals [82]. Parpinelli and Lopes [83] noted that from the inception of SI, the two major study areas were PSO [28] and ant colony optimization (ACO) [84]. Although there are many recent SI techniques, this review will concentrate on the application of PSO and ACO to fraud detection. Both methods have been applied successfully in solving several optimization problems including e-fraud including email spam, phishing and network intrusion as we present in this section.

ACO (initially proposed by Marco Dorigo [85] in 1992), is another NI technique that can be used for solving complex real world optimization task (such as e-fraud detection). ACO is inspired by the foraging behaviour of some ant species [84]. In search for food, these ants migrate from their nest (in colony) to different locations until they discover a food source. On their way back home, these ants deposit pheromones on the ground for the sake of the other members of their colony. These deposited pheromones help the other members to decide on the shortest path to take when they are also going back to their nest. Kolias et al. [86] explain that the path with greater pheromone concentration has the higher probability of being selected and ants that follow this path will, likely, return to their nest earlier than the ants that follow a path with a lesser pheromone concentration.

PSO is a meta-heuristics inspired by the cordinate movements of bird flocks and fish schools [83]. It is a search technique which uses a set of agents to locate the global minimum or maximum in a search space [87]. Parpinelli and Lopes in [83], noted that each agent (or particle) can be seen as a potential solution to the problem at hand and the position of each particle is determined by the solution that they currently represent. Changes in the position of each particle are guided by a mathematical formula which controls the particle's velocity and position. The movement of each particle is influenced by the quality of its own local experience (that is, the local best) as well as the quality of the neighbouring particles in the swarm [83]. This is expected to move the swarm towards the overall best and optimized solution. According to the literature, PSO has been applied successfully to solve several optimization problems.

The next subsections presents some application of ACO and PSO to spam email detection, phishing and network intrusion detection.

2.4.3.1 ACO for E-mail Spam Detection

El-Alfy [88] developed an ACO-based spam filter. That researcher proposed an algorithm (called AntSFilter) that uses ACO to mine through the dataset and extract the relevant set of rules. These extracted rules are then used to train the email classifier. El-Alfy evaluated AntSFilter on a public dataset called Spambase (made available by UC Irvine (USA) ML Repository repository [89]) and compared its performance to three other ML algorithms (Multi-Layer Perceptron (MLP), NB and Ripper classifiers). El-Alfy reported that AntSFilter performed better. In another work, Yin et al. [90] proposed a hybridized spam filtering algorithm (called LAD-ACO) that combines both linear discriminant analysis (LDA) and ACO. LDA was used to reduce the dimensionality of the feature space, and ACO was used to classify the email into spam or legitimate emails. Yin et al., evaluated the performance of LAD-ACO. They performed several experiments on a dataset obtained from Ling-Spam (a publicly available dataset), and compared the results obtained with two other ML algorithms (SVM and Naïve Bayes) in terms of precision, recall and TCR, and the result yielded 96.83%, 90.25% and 5.78%, respectively.

2.4.3.2 ACO for Phishing Detection

Radha and Valarmathi [91] proposed an ACO-based phishing website detection system. Furthermore, they proposed an associative classification algorithm that, progressively, finds existing association rules among data items. To avoid performing a comprehensive search for all possible set of rules in the dataset, Radha and Valarmathi used ACO to extract only the set of relevant rules. The extraction was carried out by updating the pheromones values in such a way that a set of better rules will be generated for each generation. Thereafter, only the set of rules that meet a predefined threshold will be selected. The selected rules will then be used to build a model for the phishing website classifier. Radha and Valarmathi tested their algorithm on 512 website URLs, and the test yielded an accuracy of 98%. The same authors, in another work [92], proposed an e-banking phishing website detection system that uses classification a data mining algorithm and PSO to classify a phishing website. ACO was used to improve the classification result (that is, the number of correctly classified phishing website). They worked with 27 features extracted from two publicly available dataset, and their implementation yielded an accuracy of 91%. Jensen and Shen [7] proposed a new fuzzy-rough feature selection (FRFS) technique based on ACO. They improved the data reduction process of the original fuzzy-rough set method by using ACO to extract the optimal subset of features. Jensen and Shen noted that FRFS can be applied to ML problems such as phishing detection. They applied their technique to complex systems monitoring and compared the result obtained with the original version of the fuzzy-rough feature selection method. They also compared their result to some other feature selection methods. Jensen and Shen reported that their test yielded the best data reduction on average and the smallest set of reducts.

2.4.3.3 ACO for Network Intrusion Detection

Abadi and Jalili [93] proposed an ACO-based optimization algorithm, called, AntNAG, for the minimization analysis of a large-scale network. To enhance the performance of their algorithm, they used a local search heuristic. AntNAG was tested on 12 different large-scale network attack graphs. The result showed that the proposed algorithm can be used successfully for minimization analysis of large-scale network. Zhang and Feng [94] proposed a new approach to network intrusion detection - a framework based on two existing ML methods, SVM and CSOACN (Clustering based on Self-Organized Ant Colony Network). Zhang and Feng pointed out that the proposed system can work in three modes: it can work using pure SVM, pure CSOACN or a combination of both. Zhang and Feng evaluated the performance of the three modes and the hybridized classifier performed better in terms of average detection rate, FP and FN rates and in terms of training time. Gao et al. [8] presented a new intrusion detection approach based on SVM and ACO. In their approach, they used ACO for the selection of relevant features and finally parsed the selected features to SVM for classification. The proposed method was tested on the MIT's KDD Cup 99 dataset [89] and yielded a prediction accuracy of 99.4%, 95.2% and 98.7%, respectively, on three different subsets of that dataset.

2.4.3.4 PSO for E-mail Spam Detection

Behjat et al. [95] proposed a new approach to spam e-mail detection. They used Binary Quantum PSO (BQPSO) for feature selection. Behjat et al., reported that BQPSO reduced the number of extraneous and redundant features which, in turn, increased the prediction accuracy of their MLP classifier. In another work [9], the same authors used a PSO-based algorithm to solve the high dimensionality problem associated with feature selection in spam email detection. The study found that the proposed PSO-based algorithm yielded excellent feature selection results and a high classification accuracy. Xue et al. [96] proposed two novel PSO-based multi-objective feature selection algorithms to provide solutions to a multi-objective spam detection problem involving the maximization of classification performance and minimization of the number of selected features. Xue et al., investigated the two novel approaches and their findings revealed that the approaches performed better, compared to some conventional methods.

2.4.3.5 PSO for Phishing Detection

Sumathi and Prakash [97] proposed a novel PSO-based method for phishing websites detection. The method used data mining and association rule approaches to search for relevant phishing patterns from a dataset consisting of phishing websites. Sumathi and Prakash used PSO to search for the best pair of hyper parameters for SVM classification. The study showed that PSO can be used to improve the prediction accuracy and efficiency of classifiers. Pandey and Ravi [98] proposed another new method, known as PSO Trained Auto Associative Neural Network (PSOAANN) to provide solutions to problems faced by phishing email and phishing website detection. PSOAANN consist of three layers: input layer, hidden layer and output layer. The input and output layers consist of an equal number of nodes, while the number of nodes in the hidden layer is user defined. PSO was used to train the network. Pandey and Ravi evaluated PSOAANN on two different dataset containing phishing websites and phishing emails, respectively. The result showed that PSOAANN can classify phishing emails adequately without feature selection and also classify phishing websites when used with a new feature selection algorithm, proposed by Pandey and Ravi.

2.4.3.6 PSO for Network Intrusion Detection

Chen and Qian [99] used PSO in combination with RBF NN to determine solutions concerning network intrusion detections. PSO was used to optimize the RBF parameters. The study found that the proposed technique (PSO-RBFNN) performed better than the existing RBF NN. Srinoy [10] proposed a technique that used PSO to solve the feature selection problem in network intrusion detection. They evaluated the performance of the approach on five multi-class problems and their study revealed that their method performed better in terms of accuracy, compared to some other classification methods. SVM was used as the PSO fitness function evaluators. Yi et al. [100] proposed a model (known as PSO-BP) for intrusion detection. Their model combined PSO and back propagation (BP) NN together. In the model, the PSO algorithm was used for parameter optimization and the optimized parameters were then used to train the BP NN. PSO-BP was evaluated and the results showed that the algorithm can effectively detect intrusive activities. Saxena [101] introduced a novel method for network intrusion detection. That researcher used PSO in combination with SVM and K-Means clustering algorithm in his approach. PSO was used for parameter optimization, K-Means was used to generate different training subsets and SVM was used to classify intrusive attacks from legitimate actions in the network.

A summary of all the e-fraud applications discussed above and some other applications outlined by [2] are given in Table 2.3.

Table 2.3: NI techniques and its applications

Nature-Inspired Techniques	**Applications**	**References**
ANNs	Email Spam Classification Phishing Detection Network Intrusion Detection	[66, 67] [58, 102] [103-105]
EAs	Email Spam Classification Network Intrusion Detection	[76, 77] [80, 81, 106-110]
ACO	Network intrusion detection Email Spam Classification Phishing Detection	[8, 93, 94] [88, 90] [7, 93]
PSO	Network intrusion detection Email Spam Classification Phishing Detection	[10, 99, 111] [9, 95, 112, 113] [92]
Artificial Immune Systems	Email Spam Classification Network Intrusion Detection	[114] [115]
Learning classifier systems	Network Intrusion Detection	[51, 116, 117]

2.4.4 Application of ML-based Techniques to e-Fraud

Some of the ML-based techniques proposed in literature to handle email spam, phishing and network intrusion are discussed in this section. Specifically, we highlight the application for SVM, decision trees and RF (a random combination of the two) and NB techniques for classification and detection of e-fraud. Brief overviews of these techniques are provided before presenting a review of literature on their application to e-fraud.

SVM is a content-based supervised ML model that has embedded learning algorithms, responsible for analysing data and also recognizing patterns in data. It is very useful in the domain of data classification and regression analysis. In a typical classification problem, the dataset is, usually, divided into two – training dataset and test dataset. As it is a supervised ML method, each instance in the dataset are, usually, labelled (that is, each instance belongs to a given class, e.g. spam and non-spam) [27]. Furthermore, each instance contains some set of features or attributes which is, usually, extracted from them. The primary aim of SVM is to perform predictions, based on the extracted features [27]. SVM performs predictions in the following way:

1. Some set of features (in the form of vectors) are extracted from both the training data and the test data.
2. The features extracted from the training dataset is used to train SVM
3. After training, SVM produces a model
4. The features extracted from the test dataset is then used to test the trained SVM model
5. SVM classifies the test dataset into the given classes based on the knowledge learnt from the training data.

SVM has proved to be very efficient in handling ML-related tasks. Several SVM-based classification techniques have been proposed in literature as discussed in the subsections below.

Decision trees (DT) are predictive models used for tasks involving prediction of classes. They carry out predictions by using a set of binary rules to calculate specific target value. Generally, decision trees are constructed with the primary aim of identifying a strategy to reach a goal. DTs are useful for tasks involving classification and tasks involving prediction. Classification decision trees are used to handle problems involving categorical values, while regression decision trees are useful for tasks involving continuous value [118]. DTs, usually, have some embedded algorithms used for splitting a node into two daughter nodes. Sometimes, some DTs can have hundreds and even thousands of nodes which could be very expensive in terms of processing time and memory consumption. To avoid this, a method known as pruning is, usually, used to reduce the number of nodes which in turn simplifies the tree [118]. DTs are very useful and efficient in handling classification and regression problems. The decision rules these trees generate are usually easy to interpret and also performs classification task very fast [118].

RF is an ensemble data mining technique introduced by Leo Breman [119] at the University of California, USA. RF performs classification and regression using a combination of decision tree models. RF was introduced to provide solutions to problems involving [120]:

1. Data mining
2. Analysis of data
3. Building a predictive model

Ensemble models (such as RF) usually perform better compared to individual models because they combine the results from different models [118]. This thesis employed RF, hence, more details on RF are provided in Chapter three.

Finally, NB is a commonly used method in the domain of text categorization [121]. Classifiers here are usually constructed based on Bayes theorem. In NB classification, feature values are independent; they contribute independently to the classification output [121]. The presence or absence of each feature has an impact on the prediction accuracy of the constructed model. To train a Naïve Bayesian classifier, just a small amount of data is needed to estimate the means and variance of the variables [121]. NB classification has been applied successfully to complex real world classification tasks and it yielded promising results.

2.4.4.1 SVM for Email Spam Detection

Wang et al. [122] proposed a classification model for spam filtering using SVM for classification. They noted that the model is capable of performing active online learning. The model was evaluated on a publicly available spam dataset, known as TREC2006. The results revealed that the model is proficient in spam email filtering. Similarly, Jongsub et al. [123] proposed another classification method for email spam filtering. The approach used SVM algorithm. Jongsub et al., explained that the proposed approach follows the learning procedure used by the n-gram model. The study showed that their proposed approach performed better than some other methods. Chi-Yao and Ming-Syan [124] proposed a SVM-based model for spam detection and also proposed an incremental update scheme for re-training the SVM model. They developed a spam detection system (known as MailNet) based on the proposed model, which they claimed to be capable of detecting spam emails in evolving social network. MailNet was evaluated on a live dataset, obtained from the email server of a university and the findings revealed that the SVM-based model can used for spam detection in a real world scenario.

2.4.4.2 SVM for Phishing Detection

Bergholz et al. [6] proposed a novel model-based approach suitable for detecting phishing email. In their study, two model-based phishing email features, Dynamic Markov chain and Latent Topic Model, were introduced. Bergholz et al., performed a number of experiments using SVM classifier (and some other classifiers) and their results showed that the proposed model is well suitable for detecting phishing emails. Chandrasekaran et al. [125] proposed a content-based approach to phishing email detection using SVM model. In the approach, some structural phishing features contained in emails were extracted from emails and used as the criteria for classifying emails into legitimate and phishing emails. In a similar work, Huang et al. [126] proposed another method for detecting phishing URLs. The study also used 23 features, 4 of which are structural phishing URL features, 9 are lexical features and 10 are brand name features. Huang et al. evaluated their approach using SVM classifier, and it produced a prediction accuracy of 99.0%.

2.4.4.3 SVM for Network Intrusion Detection

Mukkamala et al. [127] used SVM and NN to solve the problem posed by network intruders. The study revealed that SVM and NN are highly efficient in performing either attack or normal classifications. The study also showed that SVM and NN would decline very slightly in performance if they are trained with small amount of data. Khan et al. [53] proposed a method for network-based anomaly detection using SVM for classification. Khan et al., also used a Dynamically Growing Self-Organizing Tree (DGSOT) algorithm to improve the training time spent by SVM. The study revealed that the proposed method performed better, compared to the Rocchio Bundling technique, in terms of training time and accuracy.

2.4.4.4 Random Forest for Email Spam Detection

Irena et al. [128] studied two different learning methods for classification of spam emails – supervised and semi-supervised learning methods. In the study of the supervised learning approach, Irena et al., investigated the use of RF classifier for both filtering of spam emails and also filing of emails into folders automatically. Irena et al., compared the performance of RF with some other classification algorithms and the study showed that RF outperformed the other algorithms in terms of classification accuracy, implying that RF is a suitable approach that can be used for email filtering. In another work, Tang et al. [129] developed two different models for spam email detection based on RF and SVM with the goal of building a classifier capable of detecting spam emails. Tang et al., extracted some global sending behaviour features from a database containing a mixture of spam and non-spam IP addresses. These features were then used as input to the two models. The model returns the reputation scores of all the IP addresses as output. The study reveals that RF and SVM are reliable spam classification techniques when combined with global sending behaviour features of IP addresses. McCord et al. [130] proposed a detection scheme for discovering spam messages on Twitter (a social networking website). McCord et al., extracted some spam, non-spam, user-based and content-based features from a dataset containing Tweets (messages) and followers/following information of active Twitter users. Furthermore, McCord et al., evaluated the proposed detection system, based on the extracted features using four different data mining techniques and the results revealed that RF classifier outperforms the other three, in terms of precision and F-measure.

2.4.4.5 Random Forest for Phishing Detection

DeBarr et al. [131] introduced a new method for analysis of phishing messages, based on spectral clustering using an RF classifier. In the method, spectral clustering was used to analyze URL links for websites, usually contained in messages sent by phishers. DeBarr et al., evaluated the performance of their methods on two publicly available corpuses and compared the result with another filtering technique, that is, LDA. The results revealed that the RF classifier, combined with spectral clustering, performed better in terms of classification accuracy, precision, F-measure and Area Under the receiver operating characteristic Curve (AUC). Aggarwal et al. [132] developed a phishing detection technique known as PhishAri, for detecting phishing URL on twitter network, real-time. The detection system was developed, based on URL features and Twitter specific features, which DeBarr et al., claimed to be a strong detection mechanism for detecting phishing Tweets. DeBarr et al., evaluated their proposed technique using three different classifiers – RF, DT and NB classifiers. The study found that RF performed better, compared to the other two, in terms of prediction accuracy. The study also revealed that PhishAri is capable of detecting new or zero-hour phishing Tweets in real time. Sarju et al. [133] proposed a spam email filtering technique based on 46 structural features extracted from a dataset containing 8,000 emails. In their study, three classification algorithms were used, namely, NB classifier, RF and AdaBoost. Some experiments were conducted to test the accuracy of the proposed method and the results revealed that RF classifier had a better classification accuracy compared to the Bayes classifier and AdaBoost.

2.4.4.6 Random Forest for Network Intrusion Detection

Jiong et al. [134] proposed three RF-based approaches to provide solution to the problems posed by anomaly, misuse and hybrid-network based intrusions. In the approach, RF algorithm was used to automatically build patterns of network intrusions and patterns of network services for misuse intrusions and anomaly intrusions, respectively. Jiong et al., developed three different IDS (misuse IDS, anomaly IDS and misuse + anomaly IDS), based on their proposed approach. Furthermore, some experiments were performed to evaluate the performance of the proposed approaches and the results showed that the three approaches yielded high detection rate and low FP rates. Similarly, Elbasiony et al. [135] introduced three intrusion detection approach that used a RF classification algorithm and a k-means clustering algorithm. The first and second techniques were proposed to provide a solution for misuse detection using RF and anomaly detection using k-means, respectively. The third approach was proposed to provide a solution for both misuse and anomaly detection using a combination of RF and K-means. RF was used to construct network intrusion patterns from a dataset containing samples of network connections. K-means was used to cluster different network connection data with the aim of grouping most of the network intrusions into one cluster. Elbasiony et al., evaluated the performance of their three approaches on a dataset containing network connections and the test yielded promising results. Revathi and Malathi [136] proposed a novel approach for detection intrusion attacks using a combination of RF and Simplified Swarm Optimization (SSO – a simplified version of PSO) referred to as SSO-RF. SSO was used for feature selection and RF is used for classification of network intrusions. The method was tested on a dataset provided by DRAPA 98 and KDD cup 99 and the result obtained outperformed some other techniques. Also, the results revealed that SSO-RF is capable of detecting network intrusions of different kinds.

2.4.4.7 Naïve Bayes for Email Spam Detection

Hovold [137] developed a spam filter using NB classifier. The classifier performs classification based on a sequence of tokens referred to as word position-based features. Hovold also proposed a novel weighting classification system. Some experiments were carried out to test the efficiency of the system using uni-gram, bi-gram and tri-gram attributes and the experiment showed that NB classifier performs better in terms of precision, accuracy and recall, compared to some other ML algorithms. Sahami et al. [138] constructed a spam email filter using the Bayesian approach. Sahami et al., trained the email filter using domain-specific features extracted from a dataset containing both legitimate and spam emails. Also, they tested the performance of the trained classifier using four different Naïve Bayesian filtering techniques and the results revealed that NB yielded high FPs, implying that it was not efficient enough to build a content-based spam filter. Metsis et al. [139] carried out a study on five different versions of Naïve Bayes. In the study, Metsis et al., constructed six new corpus known as Enron corpus, and evaluated the performance of the NB variants on the constructed datasets, all containing ham and spam messages of varying numbers. The ham messages were generated by a group of about 150 employees working in Enron Corporation, USA. The performances of the five NB variants were compared with each other and the result showed that two of the five NB versions, (that is, Flexible Bayes and multinomial NB with Boolean attributes) yielded better results.

2.4.4.8 Naïve Bayes for Phishing Detection

Zhang et al. [17] proposed a new model for phishing web pages detection using NB classification approach. In the study, Zhang et al., developed a text classifier and an image classifier, based on the proposed model. Furthermore, Zhang et al., introduced an algorithm that hybridizes the classification results of the text and image classifiers. The text classifier uses Bayesian rules to classify the textual contents extracted from web pages. The image classifier uses the Earth Movers Distance (EMD) to quantity the visual similarity of suspicious web pages and then uses a Bayesian model to estimate the matching threshold of the measured web page similarity. Zhang et al., performed some series of experiments to evaluate the performance of the two classifiers. The classifiers performed better, compared to some other approaches. The study also showed that the hybridized algorithm performed better than both the text classifier and the image classifier. Santos[140] proposed a data mining technique based on semantic ontology to provide a solution to the phishing email detection problem. The approach used a NB classification algorithm. Santos noted that the classifier is capable of filtering phishing emails, based on some extracted features. Furthermore, Santos evaluated the performance of the data mining technique, and it yielded promising results.

2.4.4.9 Naïve Bayes for Network Intrusion Detection

Farid et al. [141] introduced a new approach for automatic detection of network intrusions using the NB classifier and boosting to provide a solution to classification problems, usually encountered in the classification of large datasets containing intrusion detections. The proposed technique uses the NB classifier to generate the probability set to be used in the training process. Farid et al., evaluated the performance of the proposed approach on a variety of network intrusions stored in the KDD99 dataset and compared it with some existing classification algorithms. The results yielded high prediction rates and low FP. Amor et al. [142] carried out an experimental study on the use of NB classifier in network intrusion detection. The study considered intrusions at three different levels. Firstly, all the classes of attacks presented in the KDD99 datasets were considered. Furthermore, Amor et al., considered four classes of attacks: DOS, R2L, U2R and Probing. Thirdly, they considered two levels of attacks by grouping the samples in the dataset into normal and abnormal attacks. Amor et al., performed a number of experiments using the samples in KDD99 and compared the performance of their proposed approach to DTs. The results revealed that NB classifier is faster (from a computational point of view) and also yields relatively similar results.

Several other AI-based techniques have been proposed to provide solution to the e-fraud detection problem faced by the vast majority of online users. Techniques include SVM, NB classifier, RF, Hidden Markov model. Some of this proposed techniques and their applications, including the ones discussed in Section 2.3, are summarized in Table 2.4.

2.5 Improving the Performance of e-fraud Detection Systems

Even though it has been established that NI techniques are very effective in solving complex real-world difficulties, it is of importance to note that these techniques will only work effectively if they are applied in excellent conditions. Behdad et al. [2] proposed some methods that can be employed in resolving the challenges of applying NI techniques discussed in Section 2.2.1, which will in turn enhance the performance of these techniques. A summary of these methods are presented next.

2.5.1 Experience Imbalance

Many of the AI techniques are developed based on the assumption that the learning environment are well balanced [2]. This is not the case in the world of fraud detection; datasets are usually not balanced [2]. Kotsiantis et al. [143] pointed out that learning from an imbalanced data is usually a difficult task for many classification algorithms. Also, the learned model is generally biased towards the majority class in the data. Some techniques that can be used to improve the performance of e-fraud detection systems are discussed next.

1. **Sampling Techniques:** This technique improves performance by artificially balancing the dataset. This is achieved by either oversampling (that is, increasing the samples in the minority class to match that of majority class) or under sampling (that is, decreasing the samples in the majority class to match that of the minority class).
2. **Using Appropriate Evaluation Metrics:** when measuring performance, to avoid inaccurate prediction by the algorithm being used, it is very important to apply the correct evaluation metrics.
3. **One-class learning:** Here, instead of training classifiers with two classes at the same time, classifiers are trained with one class at a time. This is done so that classifiers will be able to clearly differentiate between data belonging to the majority class and minority class after the training.
4. **Adaptive Parameters Used Internally by Algorithms:** The performance of an algorithm will be greatly improved if the correct values for the parameters setting are always used (and not random values). An ideal method can be developed to dynamically find the correct and optimized value to be used at every point in time.

2.5.2 Adaptive Adversaries

The performance of an e-fraud detection system depreciate progressively in an adversarial environment [144]. This is because, fraudsters tries to defeat the system's defense mechanism by changing their mode of attack [144]. Game theory can be used as a solution to spam and intrusion detection [114] [145]. Here, the classification problem is considered as a game between adaptive adversaries and intelligent classifiers [2].

2.5.3 Concept Drift

In a dynamic environment, concepts (targeted by the learning algorithm) changes frequently. To avoid inaccuracy in classification (due to changes in concept), Behdad et al. [2] suggests that parameters or rules should be evaluated continually. Some techniques that can be used to handle concept drift are outline and discussed next [146].

1. **Instance Selection:** Here, instances are selected based on their relevance to the current concept
2. **Instance Weighting:** Here, weights are assigned to various instances based on their relevance and age, and the instance with the highest weight is selected.
3. **Collective Learning:** Here, a set of concept description is maintained and selection is done collaboratively by choosing the concept with the most relevant description.

2.5.4 Noise

The effect of noise can be greatly reduced by using resampling method [2]. In resampling, the arithmetic mean of several collected samples is calculated and used. Pietro et al. [147] advised that it is better to have dynamic resampling (that is, using different resampling rates) if the level of noise in each data is unequal.

2.5.5 Unequal Misclassification Cost

A solution to this problem is to develop a system that is more sensitive to FP errors than FN errors. This is because FP errors are costlier than FN errors. To achieve this, some approaches are discussed in [148] and [149].

2.5.6 Fast Processing and Large Volume of Data

A solution to this problem is by increasing the processing speed of NI techniques when dealing with large volume of data. Some methods that can be employed to enhance the processing speed are discussed in [150] [151] [152].

2.6 Summary

Many techniques have been proposed in the literature to handle e-fraud. Some of them include: blacklist, whitelist, and ML based techniques. Among all of these techniques, ML-based techniques have proven to be the most efficient, because of their ability to detect new fraudulent attacks as they appear. NI algorithms can also be used to improve the classification speed and accuracy of ML-based techniques. In this chapter, some of the proposed NI-based and ML-based e-fraud detection techniques are reviewed. From the review, it is discovered that many of the proposed techniques did not focus on phishing email detection, which is one of the fraud types that has affected the global economy greatly. This is the primary reason why this research focused on phishing email detection. Furthermore, it was also discovered that, of the few techniques that focused on phishing email detection, many did not yield very good and accurate results. This research, therefore, focused on developing improved phishing email detection techniques with high classification accuracy and speed.

Table 2.4: AI techniques and their applications

AI Techniques	Applications	References
Naıve Bayes classifiers	Email spam detection Network intrusion detection Phishing	[66, 67, 153-156] [141, 142, 157, 158] [17, 140]
Support Vector Machines (SVMs)	Email spam detection Phishing detection Network intrusion detection Credit card fraud detection	[122-124, 159] [6, 125, 126] [53, 127] [125, 160]
Visualization	Email spam detection Phishing detection Network intrusion detection	[73, 161] [162, 163] [164, 165]
Random Forest	Email Spam Detection Phishing Network Intrusion detection	[128-130] [131-133] [134-136]
Learning vector quantization	Network intrusion detection	[166, 167]
Fuzzy association rules	Network intrusion detection	[168]
Agent-based IDS	Network intrusion detection	[74]
AdaBoost algorithm	Network intrusion detection	[71]
Hidden Markov model	Credit card fraud detection	[110]
Association Rules	Credit card fraud detection	[77]
Dempster–Shafer theory	Credit card fraud detection	[169]

Chapter Three

Proposed Phishing Detection Techniques

3.0 Introduction

Phishing attacks are prominently perpetrated via sending of e-mails. These e-mails usually contain social engineering messages (with specific phrases) that demand users to perform specific actions (such as clicking a URL). Therefore, the content of these emails are useful features for phishing detection [170]. Many detection techniques have been proposed to build e-mail filters, but many of them are only suitable for handling spam e-mails. For example, a popular method (known as “bag-of-words”) extracts all the words present in an e-mail, identifies the highest occurring words and use each of these words as the features for classification. This technique (also referred to as text classification method) works very well for the filtering of spam e-mails, but not for phishing e-mails, because, phishing e-mails contain some unique features that are only specific to phishing attacks. Features such as: presence of IP-based URLs, presence of non-matching URLs. This indicates that spam filtering techniques cannot be used to effectively handle phishing attacks. Therefore, a list of phishing-attack-specific features has to be defined and used to build an effective e-mail filtering system. Some of the existing solutions that can handle phishing attacks used different techniques ranging from blacklists [16], visual similarity [17], heuristic [18], and ML [8]. Of all these techniques, ML-based technique achieved the best result. It is noteworthy that some existing spam filtering techniques (such as SpamAssasin [171] and Spamato [172]) went beyond just “bag-of-word” methods, because they designed a set of spam emails heuristics that could also successfully detect some existing phishing emails features (such as, presence of IP-based URLs). These methods can be combined with the two techniques proposed in this research, to build a hybrid (phishing and spam) email filtering system with high classification accuracy and very low FPs and FNs. This chapter describes the two ML-based techniques proposed in this research. A brief introduction to RF, SVM and FFA is also given in this chapter.

3.1 Random Forest

RF is an ensemble learning classification and regression method suitable for handling problems involving grouping of data into classes. The algorithm was developed by Leo Breiman and Adele Cutler [119]. RF algorithm is essentially taking randomly constructed decision trees, summing them and then dividing the sum by the total number of trees. Each tree in the forest have high variance, because, each of the tree are trained on different randomly selected data. Also, each tree is built from a number of randomly selected attributes [173]. To split a tree node, a small set of attributes are randomly selected from a larger set of attributes. The attribute that gives the highest level of learning is then selected and used as the split point [173]. Each of the chosen attributes forms the nodes of the tree [173]. RF has two sources of randomness [173], namely:

- ❖ Randomness in the data (bootstrap) and
- ❖ Randomness in the attribute selection (that is, the split points).

Randomness in the data comes from the random selection of the data instances used in training each tree in the forest. And the randomness in the attributes comes from the random selection of attributes used in splitting each node of a tree. The number of combined decision trees determines the strength of the overall classifier. The more the trees, the more powerful and efficient the classifier becomes. The next subsection provide more details on how a random tree is constructed.

3.1.1 Building a Random Tree

Here, an example is considered involving building a spam e-mail classifier. Assuming there is a well labelled dataset containing both spam and non-spam e-mails and there is a database of possible spam terms. A random tree can be built using the following steps:

1. Divide the dataset into two: training and test datasets, and extract the words from all the emails in both datasets (each word is used as a feature).
2. Start constructing decision trees using the steps below.
3. Randomly select n group of words or n features (e.g. 50 words) from the training dataset
4. For each of the randomly selected 50 words, check for the presence or absence of each of these words in all of the e-mails contained in the training dataset (perform the comparison sequentially).

5. Compute the information gain (IG) for all the 50 words. (Please take note that, to select split point for each node in the tree, only compute the IG of the randomly selected points – that is, 50 words).
6. Split the dataset into two using the split point that has the highest IG. If there is a tie, just choose any of the split points (see Figure 3.1).

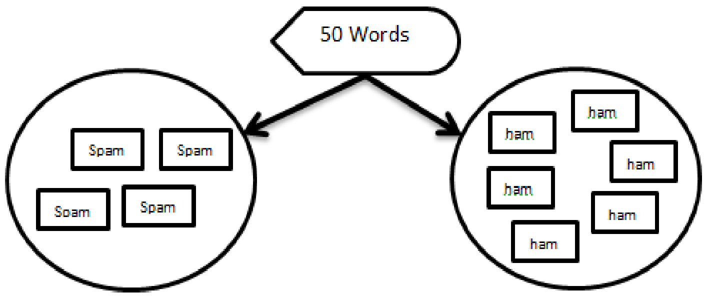

Figure 3.1: Example of daughter nodes split

7. Repeat step 3 – 7 to construct more trees.
8. Stop constructing trees when a user defined terminating value is reached.
9. Evaluate each constructed decision tree on the test dataset and sum the prediction accuracy of all the trees.
10. Output the average accuracy (that is, divide the sum by the total number of trees in the forest).

Each constructed tree in the forest is considered to be a classifier. Each of these trees have extremely high variance, because each of them is trained on different randomly selected data samples. The RF method has also been used to solve similar problem in literature, such as in [1], [128] and [18]. The RF algorithm is given in Figure 3.2

Algorithm 1: Random Forest

```
Input:   N: number of nodes
         M: number of features
         D: number of trees to be constructed
Output:  V: the class with the highest vote
1.  Begin RF Algorithm
2.  While stopping criteria is false do
3.  Randomly draw a bootstrap sample A from the training data D
4.  Use the steps below to construct tree $T_i$ from the drawn bootstrapped sample A:
    4.1. Randomly select m features from M; where m << M
    4.2. For node d, calculate the best split point among the m features
    4.3. Split the node into two daughter nodes using the best split
    4.4. Repeat I, II and III until n number of nodes has been reached
5.  Build your forest by repeating steps I – IV for D number of times
6.  End While
7.  Output all the constructed trees $[T_i]_1^D$
8.  Apply a new sample to each of the constructed trees starting from the root node
9.  Assign the sample to the class corresponding to the leaf node.
10. Combine the decisions (or votes) of all the trees
11. Output V, that is, the class with the highest vote.
12. End RF Algorithm
```

Figure 3.2: Algorithm for RF [29]

More details on RF can be found in [174] and [119].

3.2 Support Vector Machines

SVMs are arguably among the most successful classification algorithms [175]. It was invented by Vapnik [176] in 1995 and it is suitable for classification problems involving two classes. Utilizing a set of labelled input data (converted to vectors), SVM builds a model that predicts the likely classes each of the input data belongs to. SVM model maps the input vectors as points in a feature space in such a way that the two classes are separated by a hyper-plane with the widest possible margin. SVM then uses this feature space to classify new data mapped into the space based on which side of the hyper-plane they fall into. An SVM with the largest margin (that is, distance between the hyper-plane and the closest data point) will yield a better result. The wider the margin, the better the classifier performance. Support vectors (as shown in Figure 3.3) are the closest points to the hyper-plane. They are the critical points that give the maximum margin for all the N-points in the data; these points support the hyper-plane, hence, they are termed support vectors.

Given a set of training dataset, SVM finds the best (or optimal) hyper-plain with the maximum margin, which split the dataset into two different classes (e.g. positive and negative). To split the dataset into two different classes (with a minimal number of training errors), an optimal hyper-plane that best separates the two classes need to be constructed. This can be achieved by solving the optimization problem in equation (1) [176]:

$$\min_{w,b,\in} \quad \frac{1}{2}W^TW + C\left(\sum_{n=1}^{N} \in_n^{\sigma}\right)$$

$$\text{Subject to:} \quad y_n(W^T.x_n + b) \geq 1 - \epsilon_i, \qquad n = 1,2,\dots,N \qquad (1)$$

$$\epsilon_n \geq 0, \qquad n = 1,2,\dots,N$$

$$W \in \mathbb{R}^d,\ b \in \mathbb{R}$$

Where C is a constant (also called penalty parameter), x is the input vector, y is the data label (+1 or -1) and W is a vector belonging to the Euclidian space d. If C is satisfactorily large and σ is reasonably small, the vector V_0 and constant b_0 that best minimizes the function (1) subject to the given constraints determines the optimal hyper-plane, which is then used to classify the dataset(s) with the best margin. Equation (1) describes an optimization problem that finds the best hyper-plane (that is, hyper-plane with the optimal margin) which minimizes the number of training errors (given by the sum of deviation) and also maximizes the margin with the correctly classified data points [176]. Correctly classified data points (or vectors) are the points that agree with the label y_n. Hsu et al. [14] pointed out that $K(x_i, x_j) \equiv \phi(x_i)^T\phi(x_j)$ is called the kernel function, which provides simple bridge from linearity to non-linearity for various algorithms that can be expressed in terms of dot products [177].

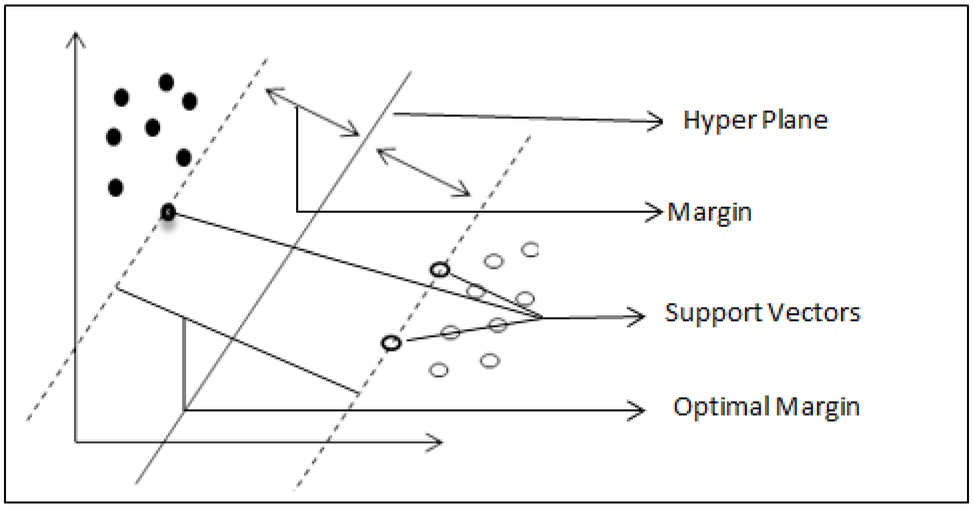

Figure 3.3: Diagram showing support vectors, margin and hyper plane

Hsu et al. [27] outlined four commonly used kernel functions namely:

1. Linear Kernel: $K(x_i, x_j) = x_i^T x_j$
2. Polynomial Kernel: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$
3. Radial Basis Function (RBF) Kernel: $K(x_i, x_j) = \exp(-\gamma \parallel x_i - x_j \parallel^2),\ \gamma > 0$
4. Sigmoid Function kernel: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$

Where $\gamma, r\ and\ d$ are the kernel parameters.

For this work, the RBF kernel is used, because it has the ability to handle cases that involve the nonlinear relation between class labels and attributes [27]. The RBF kernel has two parameters: C (also termed, soft margin constant) and Gamma. These two parameters have a significant effect on the decision boundary of SVM and, in turn, have an effect on the classifier accuracy [178]. The C and Gamma pair that will yield the best classification accuracy varies from problem to problem, it is dependent on the problem at hand (that is, it is problem-dependent). Therefore, identifying the best C and Gamma pair for a given problem is an important aspect of SVM classification. The traditional method of achieving this is via grid search using cross validation. Another reliable method is the use of the optimization technique. For grid search, the training data are divided into n-folds with the goal of using each part for both training and testing. For example, if the training data are divided into 5 parts, 4 parts will be for training and the 5th part will be used for validation. This process will be repeated 5 times, such that at each run, different subsets will be chosen for validation and the remaining subsets will be used for training. Each of the 5 runs gives an estimate of 1/5th of the training data, then averaging the estimate will give a general estimate of what the out of sample error would be on 4/5th of the data. A range of parameter values (that is, C and γ) are chosen (usually on a logarithmic scale) and the classifier accuracy for each of the (C, γ) pair is estimated using cross validation. The parameter pair that gives the best cross validation accuracy is then selected and used to train the SVM model. Hsu et al. [27] suggested that trying an exponentially growing sequence of C and Gamma is a good way of identifying good parameters; for example, $C = 2^{-5}, 2^{-3}, 2^{-1}, \dots\dots, 2^{-15}, \gamma = 2^{-15}, 2^{-13}, 2^{-11}, \dots\dots, 2^{3}$. SI techniques such as, FFA, PSO, ACO, glow worm (GW) can also be used to search for the optimized (C, γ) pair**.** These techniques are capable of finding the best available value from a large search space by maximization or minimization method. In this research, FFA is used to search for the best (C, γ) pair. A brief introduction to FFA is presented in the next section.

3.3 Firefly Algorithm

FFA (developed by Yang [29]) is a metaheuristic that is inspired by the flashing lights of fireflies - a unique and amazing feature of fireflies. There are about 2,000 species of firefly, and most of them produce short flashes (by a process called bioluminescence) at regular intervals [29]. The firefly produces these flashlights to attract potential mating partners and preys and also to send warning signals to predators that might want to attack them. FFA is stochastic in nature and can be used to solve difficult real world optimization problems and NP-hard problems [179].

The light intensity of the flashlight produced by firefly decreases with increase in distance, that is, light intensity is inversely proportional to the square of distance (that is, $I \propto {}^{1}/_{r^2}$). Also, as the distance increases, light is absorbed into the atmosphere which in turn reduces the light intensity. Yang [29] explained that flashlight can be formulated in such a way that it will be associated (or proportional) with the value of the fitness function to be optimized. The author also noted that some of the flashing attributes of fireflies can be idealized, so as to develop FFAs. FFA has a number of variants, but this paper focuses on the original version developed by Yang [29]. This algorithm was formulated using three idealized rules as follows:

1. All fireflies are unisex.
2. The attractiveness of fireflies is proportional to their light intensity, implying that fireflies with less brighter intensity will move towards fireflies with brighter intensity.
3. The light intensity of fireflies is affected or determined by the landscape of the objective function to be optimized.

In FFA, there are two important issues that need to be defined namely, the variation of light intensity and the formulation of attractiveness. Typically, for maximization problems, the light intensity (I) produced by a firefly at location, y, is directly proportional to the fitness value of the objective function, that is , $I(y) \propto F(y)$. Light intensity varies with distance between fireflies, and with the degree of absorption by the atmosphere, given by equation (2):

$$I(r) = I_0 e^{-\gamma r^2} \tag{2}$$

Where I_0 is the original light intensity, γ is the light absorption coefficient (constant), and r is the distance. Take note that in equation 1, the singularity at r = 0 is avoided in the expressionI/r^2, by combining the effect of both the inverse square law and absorption and approximating them in Gaussian form as shown in equation 1. Also, the attractiveness, β of various fireflies is proportional to the light intensities emitted by them. It is defined in equation (3):

$$\beta = \beta_0 e^{-\gamma r^2} \tag{3}$$

Where β_0 is the attractiveness at r = 0.

The distance between two fireflies $x_i \; and \; x_j$ is expressed by the Euclidian distance given in equation (4):

$$r_{ij} = \| x_i - x_j \| = \sqrt{\sum_{k=1}^{d} \left(x_{i,k} - x_{j,k}\right)^2} \tag{4}$$

Where d represent the dimensionality of the problem at hand. The movement of firefly i to a more attractive firefly j is controlled by equation (5):

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} \left(x_j - x_i\right) + \alpha \epsilon_i \tag{5}$$

$\alpha \in [0,1], \gamma \in [0, \infty)$. ϵ_i are random numbers drawn from Gaussian distribution, ϵ_i can simply be replaced by $rand - {}^{1}/_{2}$ where *rand* belongs to the set of uniformly generated real numbers between 0 and 1. The second term in the equation reflects firefly movement as a result of attraction to fireflies with brighter intensities. When $\beta_0 = 0$, movement will depend on random walk only.

Algorithm 2: Firefly Algorithm

```
1. Define initial values of firefly parameters: NF, NG, $\beta_o$, $\alpha$, and $\gamma$
2. Define Fitness function G(x), x = (x1.,…xd)t
3. Initialize n positions of firefly (i=1,2,3,….n)
4. Evalute G(x) to determine light intensity $L_i$ of firely $x_i$
5. while (j < NG)
   5.1 for k = 1 to NF
       5.1.1 for m = 1 to NF
           5.1.1.1 If ($L_k < L_m$)
               5.1.1.1.1 Move firefly k towards firefly m
           5.1.1.2 end if
           5.1.1.3 Calculate attrativeness variance with distance r using exp(-$\gamma$r)
           5.1.1.4 Calculate new fitness values for all fireflies
           5.1.1.5 Update firefly light intensity $L_i$
       5.1.2 end for
   5.2 end for
6. end while
7. Output optimized firefly light intensity
```

Figure 3.4: Algorithm for firefly [29]

3.4 Proposed RF-Based Technique for Phishing Email Detection

This technique is based on the RF algorithm. In this RF-based technique, firstly, the email dataset is divided into 10 groups. Furthermore, the feature vectors of all the features described in Section 3.6 are extracted from a dataset consisting of phishing and legitimate emails. After, the extraction, the IG for all the extracted vectors is calculated and the features with the best 8 IG are then selected and used to train the classifier via 10-fold cross validation. At the end of each fold, a tree is constructed and the CA for the tree is calculated. At the end of the 10-fold cross validation, the average CA (that is, the average CA of all the trees in the forest) is calculated and displayed on the screen. The pseudocode and flowchart for the RF-based technique used in this work is given in Figures 3.4 and 3.5.

Algorithm 3: RF-Based Classifier

```
Input:    NFold: Number of folds for cross validation
          NT: Number of Trees
Output:  Average - Average Classification Accuracy
1.  Process dataset
2.  Extract features
3.  Calculate information Gain and select best 8 features
4.  For I = 1 : NFold
    4.1. For J = 1 : NFold
        4.1.1.  If (J == I)
            4.1.1.1.    Process test dataset
        4.1.2.  Else
            4.1.2.1.    Process training dataset
    4.2. End J
    4.3. For K = 1 : NT
        //Start training the classifier by building different trees
        4.3.1.  Build tree K using the current training dataset
    4.4. End K
    4.5. For L = 1 : NT
        //Test all the trees on the current test dataset (that is, calculate the CA for each tree)
        4.5.1.  Sum  = Sum + CA
    4.6. End L
    4.7. Average  = Average + Sum
    4.8. Sum  = 0
5.  End I
6.  Average = Average / NFold
7.  Output Average
```

Figure 3.5: Pseudocode for RF-based classifier

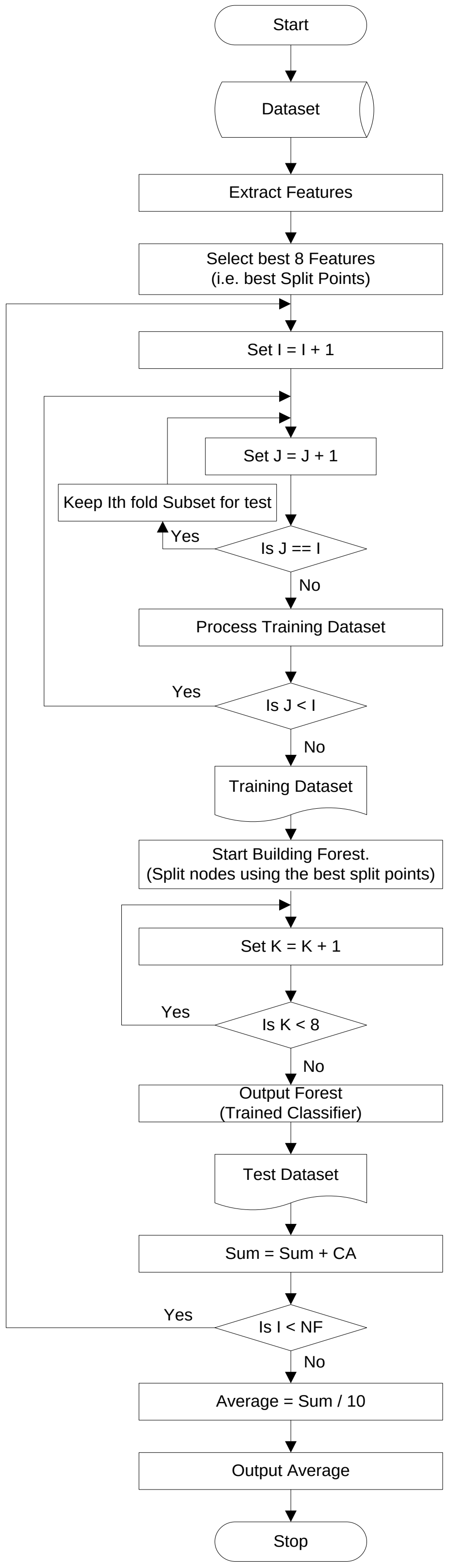

Figure 3.6: Flowchart for RF-based classifier

3.5 Proposed FFA_SVM Techniques for Phishing Email Detection

This novel technique is developed based on the hybridization of FFA and SVM (known as FFA_SVM). This technique consists of three major phases: the feature extraction phase, hyper-parameter selection phase and the email classification phase. In the feature extraction stage, the feature vectors of all the features described in Section 3.6 is extracted and saved in a file for easy reference. The extracted feature vectors is then divided into two sets, one set for training and the other set for testing. In the second phase, FFA is used to find the optimized SVM hyper-parameter pair (that is, C and γ). This is achieved through cross validation. The hyper-parameter pair with the highest cross validation accuracy is, usually, the optimized hyper-parameter pair. To further improve the classification accuracy, a hyper-parameter search algorithm (shown in Figure 3.7) is developed, based on the search technique proposed by Hsu et al. [27]. Hsu et al., suggested that the best way to identify the optimized C and Gamma values is by trying exponentially growing sequence of C and Gamma [27]. To the best of our knowledge, this method has not been explored in the domain of phishing email filtering. In the third phase, the selected hyper-parameter pair is used to construct a SVM model (based on the extracted feature vectors in the training dataset) which is, in turn, used to classify new emails. Finally, the performance of the constructed SVM classifier is then evaluated on the test dataset. The flowchart and pseudocode for FFA_SVM is shown in Figure 3.8 and Figure 3.9 respectively.

3.5.1 Objective Function

The objective function for FFA_SVM is defined by Equation (6). Classification accuracy is the major criteria used in designing the objective function. This implies that a firefly with high classification accuracy will yield a high fitness value. The objective function has one predefined weight,W_A, for the classification accuracy. The weight can be adjusted from 80 to 100%, depending on how high users want the accuracy to be. In this research, 95% was used. A similar approach was used in [180] and [181].

$$fitness_i = W_A * SVMAcc_i \qquad (6)$$

Where W_A = Predefined weight for SVM accuracy,

$SVMAcc_i$ = SVM accuracy for each firefly generation

Algorithm 4: ExponentialSequenceGen

```
Input:   CMinPower: the minimum exponent of C
         CMaxPower: the maximum exponent of C
         GMinPower: the minimum exponent of Gamma
         GMaxPower: the maximum exponent of Gamma
         N: number of exponential sequence of C and γ pair of values to generated
         Rand: randomly generated integer value
         CList: List containing all the C values
         GList: List containing all the G values
Output: PList: list containing the paired C and Gamma values
1.  Begin ExponentialSequenceGen
2.  Enter the value for N
3.  Generate random number and save it in variable 'rand'
4.  CMinPower = -rand
5.  CMaxPower = CMinPower + ((N - 1) * 2)
6.  GMinPower = -CMaxPower
7.  GMaxPower = GMinPower + ((N - 1) * 2)

8.  While (CMinPower < CMaxPower)
    8.1   B = 2 ** CMinPower
    8.2   Add B to CList
    8.3   CMinPower = CMinPower + 2
9.  End While
10. While (GMinPower < GMaxPower)
    10.1   C = 2 ** GMinPower
    10.2   Add C to GList
    10.3   GMinPower = GMinPower + 2
11. End While
12. For I = 1 : N
    12.1   Pair I^th value in CList with the I^th value in GList
    12.2   Add pair to PList
13. End for
14. Output PList
15. End ExponentialSequenceGen
```

Figure 3.7: Algorithm for generating paired exponential growing sequence of C and γ values

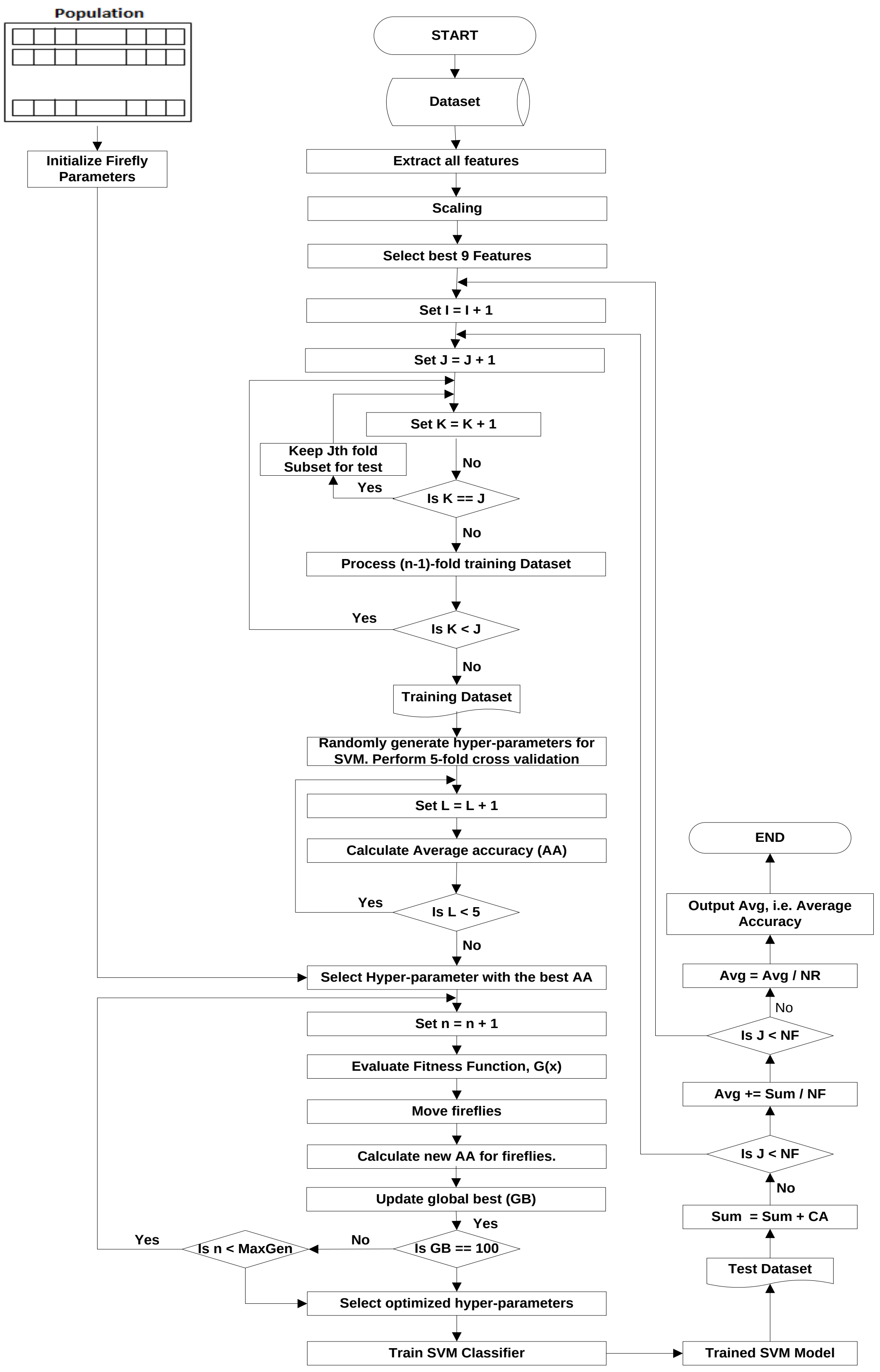

Figure 3.8: Flowchart for FFA_SVM

Algorithm 5: FFA_SVM

```
Input:   NR: Number of runs
             NF: Number of folds for FFA_SVM cross validation
             NFF: Number of fireflies
             NFold: Number of folds for SVM cross validation
             MaxGen: Maximum Generation
Output: ACA: Average Classification Accuracy
1.    Process Dataset
2.    Calculate information Gain and select best 8 features
3.    Define initial values of firefly parameters: NF, NG, $\beta_o$, $\alpha$, and $\gamma$
4.    Evalute G(x) to determine light intensity $L_i$ of firely $x_i$
5.    For i = 1 : NR
      5.1.   For J = 1 : NF
            5.1.1.   Generate initial populations of fireflies $x_i$ (i = 1,2,…NFF)
            5.1.2.   For k = 1 : NFF
                  5.1.2.1.  For m = 1 : NFold
                        5.1.2.1.1.  Calculate average classification accuracy (ACA) using the firefly
                                    positions (i.e. C & $\gamma$)
                  5.1.2.2.  End m
            5.1.3.  End k
            5.1.4.  While (n < MaxGen)
                  5.1.4.1.  Evaluate fitness value
                  5.1.4.2.  Rank firefly and save global best
                  5.1.4.3.  For p = 1 : NFF
                        5.1.4.3.1.  For q = 1 : NFF
                              5.1.4.3.1.1.  If ($I_p < I_q$)
                                    5.1.4.3.1.1.1.  Move firefly p towards firefly q
                              5.1.4.3.1.2.  End if
                        5.1.4.3.2.   End q
                  5.1.4.4.  End p
                  5.1.4.5.  For s = 1 : NFF
                        5.1.4.5.1.   For t = 1 : NFold
                              5.1.4.5.1.1.  Calculate new average classification accuracy (NACA) of the
                                            new firefly positions
                              5.1.4.5.1.2.  If (NACA > ACA)
                                    5.1.4.5.1.2.1.  Update firefly positions and their corresponding
                                                    accuracies
                              5.1.4.5.1.3.  Else
                                    5.1.4.5.1.3.1.  Don't update firefly positions
                              5.1.4.5.1.4.  End if
                        5.1.4.5.2.  End t
                  5.1.4.6.   End s
            5.1.5.   End while
            5.1.6.   Select global best (i.e. the firefly with the highest fitness)
            5.1.7.  Train SVM model with the optimized parameters (obtained from the global best)
            5.1.8.  Test model on current test data (i.e. 1/10th of dataset)
            5.1.9.  Sum SVM classification Accuracy (CA)
      5.2.   End j
      5.3.   Calculate the Average SVM classification accuracy (ASCA) over the number of folds
      5.4.   Sum ASCA
6.    End i
7.    Calculate overall average SVM classification accuracy over the number of runs – divide ASCA by
      NR
```

Figure 3.9: Pseudocode for FFA_SVM

3.6 Features for e-mail classification

“Bag-of-words” representation is the approach commonly used in text categorization problems (such as spam and phishing detection) [128]. In this approach, all unique words in an entire dataset are extracted and used as single features. This, usually, results in tens, hundreds and even thousands of features (depending on the size of the dataset). Nevertheless, Bergholz et al. [46] noted that the approach used in spam filtering is quite different from the approach used in identifying phishing attacks. They explained that the techniques used by spammers to penetrate spam filters (such as typographical errors) do not, usually, appear in phishing emails. Therefore, to adequately filter phishing emails, features specific to phishing attacks will suffice. In this thesis, a group of 17 features that frequently appear in phishing emails is identified from the literature and used. Although, the number of features is low, a high accuracy was still achieved. These features are described in the remaining part of this section.

3.6.1 URLs Containing IP address

The website URL used by many legitimate organizations usually contains the name of the website (for example, www.yahoo.com, tells us that the URL is owned by yahoo). On the contrary, phishers usually mask their website name so that their identity can be hidden from users. They achieve this by using IP-based URLs (such as http://167.88.12.1/signin.ebay.com), therefore the presence of IP-based URLs in an email is an indication that the email is a potential phishing email. This feature was used in [6] and [170].

3.6.2 Disparities between ‘href’ attribute and LINK text

The HTML <a> tag defines an anchor that may be used to establish a link to another website. Linking to another website can be accomplished by defining a “href” attribute; this attribute describes the location of the website that is to be linked to. The links are usually rendered to the browser after the “Link text” has been clicked (e.g <a href=”URL Address”>Link Text</a>). The link text could be a plain text (for example, “Click here”), a URL (yahoo.com), an image or any other HTML element. If the link text is a URL (and it is a legitimate link), it should tally with the website location pointed to by the “href” attribute (e.g. <a href=”http://www.yahoo.com”> yahoo.com </a>); if there is a disparity between the href attribute and the link text (e.g. <a href=http://www.yahoo.com> boguus.com </a>), then the link is likely pointing to a phishing website. All the links (containing a URL-based link text) in an email are checked and if there is a disparity between the link text and the href attribute, then a positive Boolean feature is recorded. This feature was used in [6] and [170].

3.6.3 Presence of ‘Link’, ‘Click’, ‘Here’ in Link Text of a Link

Link text present in most phishing emails usually contain words like “Click Here”, “Login”, “Update”. Hence, the text of all the links present in an email is checked and a binary value of 0 or 1 is recorded based on the presence or absence of the words: *Click Here, Login, Update and Link* in the Link text. Similar feature was used in [8], [6] and [170].

3.6.4 Number of Dots in Domain Name

As suggested by Almomani et al. [63], the total number of dots that should be present in the domain name of a legitimate organization should not exceed three (e.g. www.google.com). A binary value of 1 is recorded if an email contains a URL whose number of dots exceeds three. This feature was used in [170].

3.6.5 Html Email

The e-mail format for each e-mail is defined by MIME standards. The MIME standard defines the type of content contained in each e-mail. The content type (defined by the content-type attribute) could be plain text (indicated by “text/plain”), HTML (indicated by “text/html”). Fette et al. [1] proposed that an e-mail is a potential phish email if it contains a content-type with attribute “text/html”, they based their argument on the fact that it is almost impossible for phishing attacks to be launched without the use of HTML links. This feature was used in [1] and [170].

3.6.6 Presence of JavaScript

JavaScript can either be embedded in the body of an email (using the script (<script>) tag) or in a link (using the anchor (<a>) tag). Some phishers use JavaScript to hide information from users. Fette et al. [1] proposed that an email is a potential phish e-mail if the “javascript” string is contained in either the body of the email or in a link. This feature was used in [1] and [170].

3.6.7 Number of Links

The total number of links embedded in an email is recorded and used as a feature for classification. Zhang et al. [71] explained that phishing emails usually contain multiple numbers of links to illegitimate websites. This feature was used in [71] and [170].

3.6.8 Number of Linked To Domain

For this feature, the URLs contained in an email are extracted, the domain names of each of these URLs are counted, and the counted value is used as a feature. Take note that each domain name is counted only once; subsequent occurrence (of an already counted domain name) is discarded and not counted. This feature simply refers to the total number of distinct domain names referenced by all the URLs in the email. This feature is also used in [170] and [1].\

3.6.9 From_Body_MatchDomain check

To extract this feature, all the domain names in an email is extracted and each of these domain names is matched with the sender's domain (that is, the domain name referred to by the "From" field of the same email); If there is a disparity between any of the comparison, Almomani et al. [63] noted that the email is likely a phish email. This feature was used in [170] and [63].

3.6.10 Spam Filter Feature

SpamAssassin is an effective open source spam filter that is already in use by some organizations today, therefore, taking advantage of its accurate filtering capability will go a long way in reducing phishing to a reasonable extent [1]. In this work, an offline and untrained version of SpamAssassin was used to generate one binary feature. An e-mail is assigned a vector value of 0 if it is labeled as ham by SpamAssassin and it is assigned a value of 1 if it is labeled as spam. This feature was used by Fette et al. [1], Bergholz et al. [6] and Akinyelu et al. [170].

3.6.11 Word List Features

Some group of words that frequently appears in phishing emails were used as features. These words were divided into six different groups, and each of these groups was used as a single feature (making a total of six different features). For each group, the presence of each word is counted and normalized. The groups of words include:

- Update, Confirm
- User, Customer, Client
- Suspend, Restrict, Hold
- Verify, Account
- Login, Username, Password
- SSN, Social Security

This feature is similar to the one proposed by Basnet et al. [182]. [27]

3.7 Summary

This chapter has given a description of the two phishing email detection techniques proposed in this research, that is, the firefly based ML technique (FFA_SVM) and the RF-based technique. Furthermore, the phishing features extracted from the emails is also discussed in this chapter.

Chapter Four

Simulations, Results and Discussion

4.0 Introduction

This chapter gives the details on the setup used for all the experiments performed in this research. It also provides details about the dataset and the performance measure used for all the experiments carried out in this research work.

4.1 Experimental Setup

The experimental setup for the RF-based classifier and the firefly based SVM is described next.

4.1.1 Performance Measure

Generally, four possibilities exist in classification problems involving two classes [15], namely: True Positive (TP, phishing emails correctly classified as phishing), FP (legitimate emails wrongly classified as phishing), True Negative (TN, legitimate email correctly classified as legitimate) and FN, phishing email wrongly classified as legitimate. In all the experiments carried out in this research, the following evaluation metrics were used:

$$\text{FP Rate} = \frac{TP_n}{FP_n + TN_n} \quad (1)$$

$$\text{FN Rate} = \frac{FN_n}{TP_n + FN_n} \quad (2)$$

$$\text{Precision (Pr)} = \frac{TP_n}{TP_n + FP_n} \quad (3)$$

$$\text{Recall (R)} = \frac{TP_n}{TP_n + FN_n} \quad (4)$$

$$\text{F-Measure} = \frac{2 * \text{Pr} * R}{\text{Pr} + R} \quad (5)$$

Where, TP_n, TN_n, FP_n, FN_n refers to the total number of true positives, true negatives, FPs and FNs in that order.

4.1.2 Cross Validation

Due to lack of sufficient data, for all the experiments performed in this research, the 10-fold cross validation method is used. Performing this will correct the statistical dependency of all the individual instances in the dataset [6], and it will also lead to a good and accurate estimate of the evaluation. The dataset used for evaluation is divided into 10 parts. The classifier is trained on 9 of these parts and then validated on the 10^{th} part. This training and validation process is carried out 10 times, such that at the end of the 10^{th} iteration, there would have been different $9/10^{th}$ and $1/10^{th}$ training and validation subset, respectively, for all the 10 runs. Also, at the end of each iteration, the prediction accuracy is calculated and saved. The average classification accuracy is then evaluated at the end of the 10^{th} run. The goal of cross validation is to define a validation subset that can be used to test the prediction accuracy of the model constructed in the training phase. Also, cross validation is known to provide a good estimate of the classifier's generalization or out-of-sample error.

4.1.3 Evaluation for RF-Based Technique

The term 'forest' in 'random forest' simply refers to a collection of decision trees. Each decision tree in the forest is usually constructed by randomly selecting n-number of features. In the RF-based approach, different decisions trees is constructed at the training phase. The constructed trees was then evaluated on the test data.

4.1.3.1 Methodology

In the evaluation performed on the RF-based classifier, firstly, 15 features (outlined in Section 3.6) are programmatically extracted from the dataset using C#. Furthermore, the vector values of all the features are normalized to one. This is to ensure that all the feature vectors fall within the same range (that is, between 0 and 1). Also, IG for all the extracted features is calculated (using the IG method explained by Mitchell [174]) and the best 8 features are selected. Using the selected features as split points, different set of trees are constructed and then used to evaluate the data samples in the dataset.

4.1.3.2 System Configuration

The system properties of the desktop computer used for all the experiments performed on the RF-based method are listed below:

- ❖ Operating System: Windows 7, 32 Bits
- ❖ Processor Speed: 2.20GHz
- ❖ Memory: 2GB RAM

4.1.3.3 Parameters settings for RF-Based Technique

As explained earlier, in the RF-based classifier, 10-fold cross validation is used. Furthermore, for the classifier training, 15 features are extracted from the dataset, and after initial sensitivity experimental runs, it was observed that 8 features and below yielded the best classification accuracy and speed. The number of features is therefore set to 8.

4.1.4 Evaluation of FFA-SVM

In this novel method, FFA is integrated with SVM. The hybridization was performed with the primary aim of developing an improved classifier capable of automatically detecting both known and emerging phishing emails.

4.1.4.1 Evaluation Method

Firstly, the vector values of 16 features are programmatically extracted from all the emails in the dataset using C# programming language. The extracted features are then normalized. IG for all the extracted features are calculated and the best nine features are selected. Furthermore, the selected features are converted to an input format required by the SVM library used for the implementation, that is, libSVM [183]. The formatted input vectors are then saved in a database for easy access. During classification, the feature vectors are scaled down to ensure that they have a mean of zero and a unit variance. The Gaussian transformation method is used for scaling. For FFA, the firefly parameters suggested by Yang [29] are used. Also, the hyper-parameter selection technique suggested by Hsu [27] is used. See Tables 4.1 and 4.2 for more details.

Many experiments are performed to evaluate the robustness of FFA_SVM. In the experiments, different pairs of C and γ values are used. Each of the (C, γ) pairs are randomly generated using ExponentialSequenceGen algorithm. The experiments are divided into two different groups. The first group is based on the hybridized SVM (that is, FFA_SVM) and the second group is based on the existing SVM method (O-SVM). That is, in the first group, FFA is used in the hyper-parameter selection phase and in the second group, the existing grid algorithm is used. For each group, firstly, 8 features are extracted from different datasets containing 150, 300, 500, 1,000, 2,000, 3,700 and 4,000 emails, respectively. Thereafter, another set of 9 features are extracted from the same group of datasets. Furthermore, 10 sets of (C, γ) pairs are randomly generated using ExponentailSequnceGen. Using each pair, a grid search is carried out on the training dataset via 5-fold cross validation. The best C and γ pair is then selected and used to generate the final classifier by re-training the entire training dataset. Also, another set of C and γ values (20 pairs) are generated. A grid search on each of these pairs is carried out. The two groups of experiments are performed separately with the sole aim of comparing the performance of both FFA_SVM and O-SVM.

4.1.4.2 System Configuration

The system properties of the desktop computer used for all the experiments that were performed on the firefly based SVM method is as follows:

- Operating System: Windows 7, 64 Bits
- Processor Speed: Intel core (TM) i7-4770S CPU @ 3.10GHz
- Memory: 8GB RAM

In all the experiments, 10 times, 10-fold cross validation is performed. Also, to further validate the obtained results and to confirm that the results are statistically significant, some statistical analysis is carried out using z-statistics. In the statistical analysis, the results obtained from FFA_SVM is compared with some other results in the literature. Z-statistics is used for the statistical analysis, because, it is suitable for the analysis of sample spaces greater than 30 samples, according to the central limit theorem. Since 10-fold cross validation, which was repeated 10 times, was carried out for each experiment, the total number of samples (that is, CA) at the end of each experiment is 100. The statistical analysis (using the result shown in Table 4.1) is discussed in the next subsection.

4.1.4.3 Statistical Analysis

The goal of this statistical analysis is to know if it can be concluded with 99% confidence level (i.e. α = 0.01) that FFA_SVM is better than the phishing email technique proposed by Fette et al. [1] (known as PILFER). As earlier stated, Z-test is used for the analysis, because, the number of samples to be analyzed is above 30. Furthermore, since 10 times 10 fold cross validation is used, the number of runs (n_1) for FFA_SVM is 100. As reported by Fette et al. [1], the number of runs (n_2) for PILFER is 10. Additionally, the average FP rate ($\overline{X_1}$) obtained from

FFA_SVM is 0.01 and the average FP rate ($\overline{X_2}$) for PILFER is 0.13. Finally, as shown in Table 4.1, the variance (S_1) for the FP rate obtained from FFA_SVM is 0.0031660, and the variance (S_2) obtained from PILFER, as reported by Fette et al. [1], is 0.0013. The parameters are formally defined next.

Parameters

FFA_SVM: $\overline{X_1} = 0.01,\ n_1 = 100,\ S_1 = 0.0031660$

PILFER [1]: $\overline{X_2} = 0.13,\ n_2 = 10,\ \ S_2 = 0.0013$

$\alpha = 0.01$

Where: $\overline{X_1}$ is the average FP rate obtained from our experiments,

$\overline{X_2}$ represents the average FP rate as recorded by PILFER [1]

Hypothesis

Let the null hypothesis be H_0, and the alternative hypothesis be H_1, where H_0 is a measure of the effect of changing from PILFER to FFA_SVM, and H_1 is what is expected to be true if the null hypothesis does not hold.

H_0: $\overline{X_1} \leq \overline{X_2}$,

H_1: $\overline{X_1} > \overline{X_2}$

H_1 is true, if and only if, $\overline{X_1} - \overline{X_2} > 0$

Calculation of Test Statistics

The Z-test is now used to test the hypothesis. The formula is given by:

$$Z = \frac{(\overline{X_1} - \overline{X_2})}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}} \qquad (7)$$

Where $S_1^2 and\ S_2^2$, $n_1 and\ n_2$, refers to the standard deviation and the number of runs for FFA_SVM and PILFER respectively.

Decision Rule

The null hypothesis is taken to be true if Z (as defined in equation (7)) is such that $Z \leq Z_\alpha$, otherwise, reject H_0. Here Z_α denotes the critical value and its magnitude is $Z_\alpha = Z_{0.01} = 2.3267$.

The critical value is the minimum value of which the alternative hypothesis is true. From our calculation, Z = 9.3257. Since Z = 9.3257 > Z_α, the null hypothesis is rejected, that is, $\overline{X_1} > \overline{X_2}$.

Table 4.1: Classification results for FFA_SVM over 100 runs

Classification Accuracy	False Positive	False Negative	Recall	Precision	F-Measure
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
99.75	0.2857	0	100	98.0392	99.0099
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100

100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
99.75	0.2857	0	100	98.0392	99.0099
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100

100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
99.75	0.2857	0	100	98.0392	99.0099
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
99.75	0.2857	0	100	98.0392	99.0099
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
100	0	0	100	100	100
Variance					
0.002424242	0.003166041	0	0	0.149129179	0.03802368

4.1.4.4 Parameters settings for FFA_SVM

The FFA and SVM parameters used for the evaluation of FFA_SVM are given in Table 4.2 and 4.3, respectively.

Table 4.2: SVM Parameters used for Evaluations

SVM Parameters [27]	C =	2^{-11}	2^{-9}	**…………………………**	2^{1}	2^{3}	2^{5}
	γ =	2^{-5}	2^{-3}	**…………………………**	2^{7}	2^{9}	2^{11}

C = regularization constant, γ = Gamma

Table 4.3: FFA parameters used for Evaluations

Firefly Parameters [29]	$\boldsymbol{\alpha}$	$\boldsymbol{\gamma}$	$\boldsymbol{\beta_0}$	$\boldsymbol{N_f}$	$\boldsymbol{N_g}$
	0.2	1	1	20	10

Key: α = alpha, γ = Gamma, β_0 = Beta, N_f = Number of firefly, N_g = Number of generations

4.1.4.5 SVM Library for FFA_SVM

As aforementioned, this work make use of an open source library for SVM, known as LIBSVM. This library contains classes capable of handling binary classification tasks, regression tasks and distributed estimation. It is also capable of handling problems involving multi-class classification. LIBSVM was originally developed in C programming language but it has been converted to many other programming language, including Java, C#, MATLAB. In this work, the .Net conversion of LIBSVM 2.89 (adapted by Matthew Johnson [184]) is used. This .Net version contains several libraries that can be used by programming languages on the .Net platform (including C# and VB.Net). This library is also extended (using C#) to suite this work. Before LIBSVM can perform a classification task, the feature vectors must first be extracted from the dataset. The extracted vectors is then expected to be formatted in such a way that each line will contain three values (a label, an index and the extracted vector). The three values are expected to be formatted in the following order: Label Index: Value (that is, a label, followed by an index value, followed by a colon, and followed by the extracted vector value). The three values are explained next.

1. **Label**

For classification task, label is an integer value referring to the class of an instance. They are used to calculate the classification accuracy.

2. **Index**

These are ordered integer values usually starting from 1. They must be in ascending order.

3. **Value**

Value refers to the extracted vector value for each feature. This can a real number or an integer number (for binary features).

For example, assuming there is a binary classification problem at hand, and the task is to process a dataset containing 6 samples and 5 attributes. The processed file (without normalizing or scaling) will typically look like:

```
-1 1:3 2:0 3:0 4:0 5:0
-1 1:0 2:0 3:0 4:0 5:0
+1 1:4 2:0 3:0 4:0 5:3
-1 1:1 2:3 3:5 4:1 5:1
+1 1:1 2:2 3:3 4:1 5:1
+1 1:8 2:1 3:6 4:1 5:4
```

In this work, the features were extracted from each instance in the dataset. Another dataset was then constructed using the format described above.

4.1.4.6 SVM Kernel and Hyper-parameter Optimization

Hyper-parameter optimization is a very important phase in SVM classification. This involves choosing the best pair of hyper-parameters, which will be used to train the entire training data. The standard method of performing hyper-parameter optimization is via grid search. The primary goal of performing a grid search is to find the parameters that gives the best cross-validation rate.

This is usually very time consuming, especially, if it involves classification of large problems. SVM consist of four basic kernel function [27], namely:

- ❖ Linear: $K(x_i, x_j) = x_i^T x_j$
- ❖ Polynomial: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$
- ❖ Radial basis function (RBF): $K(x_i, x_j) = (-\gamma ||x_i - x_j||)^2, \gamma > 0$
- ❖ Sigmoid: $K(x_i, x_j) = tanh(\gamma x_i^T x_j + r)$

In this work, SVM with RBF kernel is considered. This kernel function requires the optimization of two parameters: C and γ. These two parameters are problem-specific and they contribute immensely to the classification accuracy of SVM. Therefore, a diligent search for the best (C, γ) pair is required. This will involve performing a grid search on a sequence of (C, γ) pairs. To achieve this, each (C, γ) pair in the search space is usually evaluated using a cross valuation method. The pair that gives the highest cross-validation rate is then selected and used to train the SVM classifier. In this research, a hyper-parameter selection algorithm is developed (known as ExponentialSequenceGen). This algorithm is based on a hyper-parameter search technique proposed by Hsu et al. [27]. Hsu et al., suggested that trying an exponentially growing sequence of C and γ values is a very good way of searching for the best pair of hyper-parameter [27]. See Figure 3.7 for the algorithm and Table 4.2 for a typical example of exponentially growing sequence of C and γ values.

There are two major factors that contributes to the performance of SVM classification:

- ❖ The number of hyper-parameters generated for cross-validation and
- ❖ The sequence of the hyper-parameters generated.

The number of parameters generated have a great effect on the time consumed in the internal cross validation of the training dataset. This is primarily because, during the training phase, each pair of parameter is usually evaluated against all the samples in the validation dataset. For example, if there are 2000 samples in a validation dataset and there are 30 pair of C and γ values to be evaluated, each of the 30 pair will be evaluated one by one against all the 2000 samples. This is usually very time consuming. Also, the sequence of parameters generated have an effect on the classification accuracy of a given problem. The sequence of parameters used for ProblemA (for example) might not yield the same classification accuracy if used for another problem (say ProblemB). Therefore, proper consideration has to be taken when deciding on the sequence and number of parameters to use for the classification of a given problem.

4.2 Dataset Used

Information about the dataset used for the experiments carried out on the RF-based classifier and the firefly based SVM is given in this section.

4.2.1 Dataset used for RF based Technique

For the implementation, two publicly available and reliable datasets of emails are used: ham corpora and phish corpora. The ham corpora is obtained from SpamAssassin [25] and the phish corpora is collected from Monkey.org [26]. Each of the legitimate emails obtained from SpamAssassin are labelled as Ham and the emails acquired from Monkey.org are labelled as Phishing emails. Datasets of various sizes and varied phish-ham ratios are used. The majority of the datasets contained a higher number of ham emails, compared to phishing emails, because in a real life scenario, people get far more legitimate emails, compared to unwanted emails. A good classifier should be able to accurately detect and filter out phishing emails in the midst of many legitimate emails. All the emails are labelled and distributed evenly into 10 different folders. The summary of the dataset details used for evaluation is given in Table 4.4.

Table 4.4: Datasets used for the evaluation of RF-based Classifier.

Total Sample	Phish: Ham Ratio	Email Per Folder	Total PM per Folder	Total HM per Folder	Total PM	Total HM
150	7:8	15	7	8	70	80
300	1:2	30	10	20	100	200
500	1:4	50	10	40	100	400
1000	1:9	100	10	90	100	900
2000	1:9	200	20	180	200	1800

Key: HM: Ham Email, PM: Phishing Email

Table 4.5: Datasets used for the evaluation of FFA_SVM.

Total Sample	Phish: Ham Ratio	Email Per Folder	Total PM per Folder	Total HM per Folder	Total PM	Total HM
150	7:8	15	7	8	70	80
300	1:2	30	10	20	100	200
500	1:4	50	10	40	100	400
1000	1:9	100	10	90	100	900
2000	1:9	200	20	180	200	1800
3700	1:18	370	20	320	200	3500
4000	1:7	400	50	350	500	3500

Key: HM: Ham Email, PM: Phishing Email

4.2.2 Dataset Used for FFA-SVM

FFA_SVM is evaluated using dataset of ham emails provided by SpamAssassin and dataset of phishing emails provided by Jose Nazario (that is, Monkey.org). Datasets of various sizes and varied phish-ham ratios are used. The majority of the datasets contained a higher number of ham emails, compared to phishing emails, because, as explained earlier, in a real life scenario, people get far more legitimate emails compared to unsolicited emails. A classifier should be able to accurately detect and filter unsolicited emails in the midst of many legitimate emails. All the emails were well labelled and distributed evenly into 10 different folders. The summary of the dataset details used for the evaluation of FFA_SVM is given in Table 4.5.

4.3 Results and Discussion

Results obtained from all the experiments performed on the RF-based technique and FFA_SVM is outlined and discussed in this chapter. Results for the RF-Based technique are first discussed followed by the results for the firefly based SVM.

4.3.1 Result and Discussion for RF-based Technique

For the purpose of evaluation, five different datasets are constructed. Each of the datasets consists of 150, 300, 500, 1,000 and 2,000 emails, respectively. After the dataset construction, the RF-Based classifier is then evaluated on each of the datasets. The evaluation is performed with the primary aim of checking the performance of the RF-Based classifier on both large- and small-sized datasets and to also check its performance on datasets containing varied phish-ham ratio. As shown in Table 4.6, the dataset with the largest number of emails (2,000 emails) yielded the best prediction accuracy, FN rate, F-measure and precision. This indicates that the classifier is capable of accurately classifying datasets with large number of emails. Also, as shown in Table 4.6, there is a slight increase in the FP rate for the dataset containing 2,000 samples. This is because of the presence of noise in some of the data. Some of the emails in the dataset is not well formatted. The increase in time in each of the experiments, is due to the increase in the amount of data. Table 4.7 compares the results obtained in the RF-based technique with the result obtained by PILFER [1]. Table 4.7 reveals that the results obtained by the RF-based classifier outperformed the results obtained by PILFER in terms prediction accuracy, FP and FN rate, Precision, Recall and F-Measure. The improvement in the result is because of the difference in the types of features used by both PILFER and the RF-Based technique. Also, it is because of the difference in the feature selection methods applied by the RF-based technique and PILFER. In PILFER, 10 features was extracted and used for the email classification. Whereas, in the RF-based technique, 15 features are extracted from the email dataset. After the extraction, IG for the 15 features are calculated and the best 8 features are selected. This method is used because of the variance in the number of features contained in various emails. For example, 5 features could be found in an email, whereas, 7 features could be found in another email. Therefore, checking an email for an increased number of features increases the probability of detecting whether the email is phishing or not. Furthermore, to reduce the classification speed, the number of selected features was moderated. To know the minimum number of features required to yield the best classification accuracy, several experiments are performed. For each experiment, different number of features are used, and it was discovered that the experiments with 8 features and below yielded the best classification accuracy. Therefore, the threshold for the number of features is set to 8. Figure 4.2 shows the Receiver Operating Characteristic curve (ROC curve), comparing the performance of PILFER with the RF-Based technique. The figure reveals the improved performance of the RF-Based technique.

Table 4.6: Results obtained from RF-based classifier

S/N	Dataset Information			Performance Evaluation							
	Email Per Folder	Total Email	P: H Ratio (%)	PA (%)	SR	FP (%)	FN (%)	R (%)	Pr (%)	F-M (%)	T(s)
1	15	150	7:8	98.00	0.98	0.00	4.11	95.80	100	97.79	11.82
2	30	300	1:2	98.33	0.99	0.00	4.00	96.00	100	97.75	21.03
3	50	500	1:4	99.20	0.99	0.00	4.00	96.00	100	97.78	33.47
4	100	1000	1:10	99.60	0.99	0.00	4.00	96.00	100	97.78	65.46
5	200	2000	1:10	99.70	0.99	0.06	2.50	97.50	99.47	98.45	141.25

Key: PA: Prediction Accuracy, SR: Success Rate, FP: False Positive, FN: False Negative, R: Recall, Pr: Precision, T: Time, F-M: F-Measure, P: Phish, H: Ham

Table 4.7: Classification Result for RF-Based Classifier Vs PILFER [154]

Technique	PA (%)	FP-Rate	FN-Rate	Precision	Recall	F-Measure
PILFER [1]	99.49	0.13%	3.62%	98.92%	96.38%	97.64%
RF-Based Classifier	*99.70*	*0.06%*	*2.50%*	*99.47%*	*97.50%*	*98.45%*

Key: PA: Prediction Accuracy, FP: False Positive, FN: False Negative, R: Recall, Pr: Precision, F-M: F-Measure

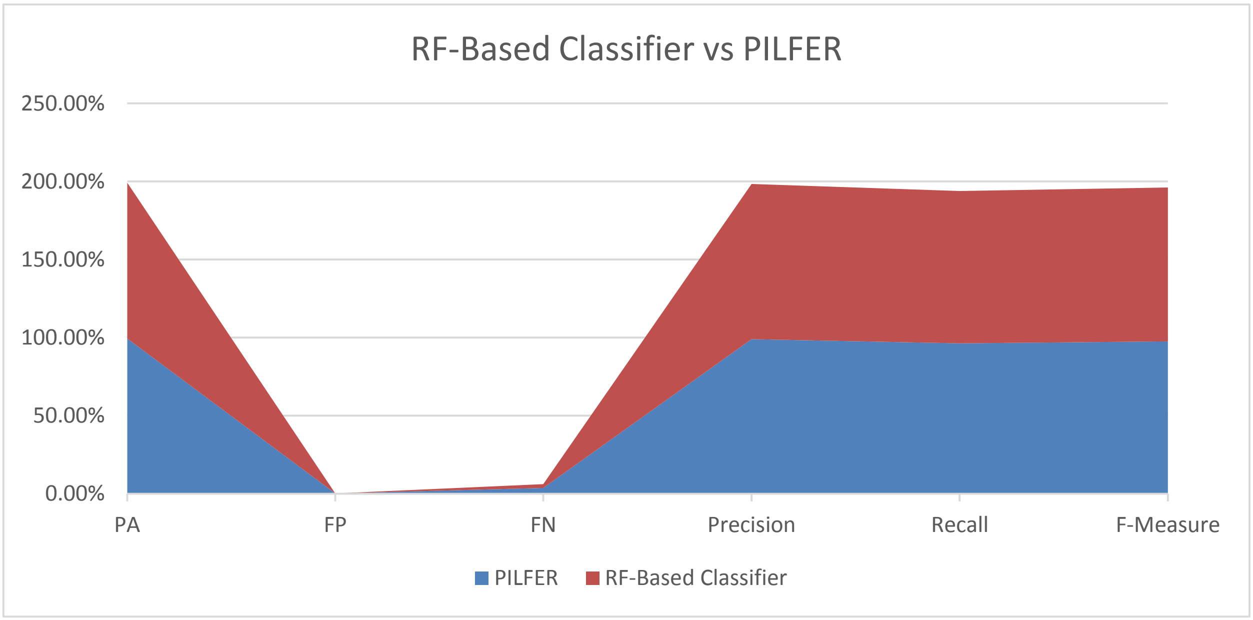

Figure 4.1: ROC Curve comparing the RF-Based Classifier and PILFER [1]

4.3.2 Result and Discussion on FFA_SVM Technique

A number of experiments are performed in order to evaluate the performance of the FFA_SVM. All the experiments yielded very good results. The classification accuracy derived from all the experiments (as shown in Table 4.9 – 4.16) falls between the range of 99.40% and 100%. This indicates that the classifier is capable of correctly classifying datasets with both a few and a large number of emails. Also, the FP rates and FN rates are between the range of 0.16% and 0.00%. This indicates that a very small number of emails is misclassified. As explained in Section 3.1.4, two group of experiments are performed. The experiments are performed with the aim of comparing the performance of FFA_SVM and O-SVM. The experiments are also performed with the aim of knowing the maximum number of (C, γ) pairs that is required to yield the optimized (C, γ) pair for a given problem. Figures 4.2 – 4.19, show the Average Prediction Accuracy (APA), Global Best (GB), Recall (R), Precision (PR), F-Measure, FP Rate and the FN Rate for each of the experiments that are performed on both FFA_SVM and O-SVM. Note that the figures are in pairs. Figures 4.2 and 4.3, 4.4 and 4.5, 4.6 and 4.7, 4.8 and 4.9, 4.10 and 4.11, 4.12 and 4.13, 4.14 and 4.15, 4.16 and 4.17, 4.18 and 4.19, show the result for the first to the ninth experiments, respectively. As shown in Figures 4.2 - 4.19, the optimized hyperparamter pair was found in a search space consisting of a maximum of 20 hyper-parameter pairs. Also, as shown in Figures 4.20 and 4.21, the time consumed by each experiment is relatively low. All this indicates that, a hyper-parameter space consisting of 10-20 exponentially growing sequences of C and γ pairs, randomly generated using the ExponentialSequenceGen algorithm, is enough to construct a fast and accurate phishing email classifier. Furthermore, as shown in Figure 4.14 – 4.19, the performance of FFA_SVM is slightly better than O-SVM in many of the experiments. This is an indication that FFA_SVM is an improved method. It also indicates that FFA can be used to improve the classification result of ML based techniques. Finally, it indicates that FFA_SVM can be used as an alternative method to O-SVM.

As shown in Table 4.8, Figures 4.22 and 4.23, the performance of FFA_SVM is compared to the three best results discovered in the literature. An improved classification accuracy of 99.99% is obtained by FFA_SVM, in contrast to 99.49% [1], 99.31% [185] and 99.85% [6] as obtained by its competitors. Furthermore, in terms of FP rate and FN rate, FFA_SVM outperformed its competitors. The improvement is because of the variance in the types of features used in the experiments and the difference in the feature selection techniques as explained in Section 4.3.1. FFA also played a very important role in the improved result. The high convergence rate of FFA and the amazing speed at which FFA precisely searches for its global optimum underscores the reason why FFA contributed immensely to the improved performance of FFA_SVM.

To further validate the result, some statistical analyses (one tail, two sample z-test) are carried out. FFA_SVM yielded a FP rate of 0.011428% and a variance of 0.003166 using 10-times, 10-fold cross validation. This leads to a z-statistic of 9.3257 when compared to PILFER [1]. On this basis, it can be concluded with 99% level of certainty, that the results produced by FFA_SVM are statistically significant. Statistical analysis could not be carried out on [185] and [6], because the FP variance (or standard deviation) obtained by [185] and [6] was not reported.

Table 4.8: Results obtained for FFA_SVM vs. Three best Results in literature

Technique	**Average Accuracy**	**FP-Rate**	**FN-Rate**	**Precision**	**Recall**	**F-Measure**
Fette et al. [1]	99.49%	0.13%	3.62%	98.92%	96.38%	97.64%
R-Boost [185]	99.31%	1.4%	0%	98.63%	100%	99.31%
Bergholz et al. [6]	99.85%	0.01%	1.30%	99.88%	98.70%	99.29%
FFA_SVM	*99.99%*	*0.01%*	*0.00%*	*99.92%*	*100.00%*	*99.96%*

Table 4.9: Results obtained from the evaluation of FFA_SVM using 9 Features - 10 Runs

S/N	**Dataset Information**			**Performance Evaluation**							
	Total Email	Email Per Folder	P: H Ratio (%)	APA(%)	GB	FP(%)	FN(%)	R(%)	Pr(%)	FM(%)	T(s)
1	150	15	7:8	99.40	100	0.75	0.43	99.57	99.25	99.37	15.52
2	300	300	1:2	99.53	99.67	0.50	0.40	99.60	99.05	99.30	43.06
3	500	500	1:4	99.76	99.80	0.30	0.00	100	98.94	99.44	77.33
4	1000	1000	1:9	99.86	100	0.09	0.60	99.40	99.21	99.29	162.21
5	2000	2000	1:9	99.94	99.95	0.06	0.10	99.90	99.47	99.68	670.38
6	3700	3700	1:18	99.96	100	0.03	0.30	99.70	99.56	99.63	1022.77
7	4000	4000	1:7	99.98	100	0.02	0.00	100	99.86	99.93	2544.72

Key: APA: Average Prediction Accuracy, FP: False Positive, FN: False Negative, R: Recall, Pr: Precision, GB: Global best, T: Time, FM: F-Measure, P: H: Phish: Ham

Table 4.10: Results obtained from the evaluation of FFA_SVM using 9 Features - 20 Runs

S/N	**Dataset Information**			**Performance Evaluation**							
	Total Email	Email Per Folder	P: H Ratio (%)	APA(%)	GB	FP(%)	FN(%)	R(%)	Pr(%)	FM(%)	T(s)
1	150	15	7:8	99.40	100	1.00	0.14	99.86	99.00	99.39	15.71
2	300	300	1:2	99.60	99.67	0.50	0.20	99.80	99.07	99.41	47.82
3	500	500	1:4	99.78	99.80	0.28	0.00	100	99.02	99.48	84.01
4	1000	1000	1:9	99.90	100	0.06	0.40	99.60	99.41	99.49	158.71
5	2000	2000	1:9	99.96	100	0.05	0.00	100	99.57	99.78	631.80
6	3700	3700	1:18	100	100	0.00	0.00	100	99.95	99.98	1066.87
7	4000	4000	1:7	99.99	100	0.01	0.00	100	99.92	99.96	2642.99

Key: APA: Average Prediction Accuracy, FP: False Positive, FN: False Negative, R: Recall, Pr: Precision, GB: Global best, T: Time, FM: F-Measure, P: H: Phish: Ham

Table 4.11: Results obtained from the evaluation of O-SVM using 9 Features - 10 Runs

S/N	Dataset Information			Performance Evaluation							
	Total Email	Email Per Folder	P: H Ratio (%)	APA(%)	GB	FP(%)	FN(%)	R(%)	Pr(%)	FM(%)	T(s)
1	150	15	7:8	99.40	100	0.88	0.29	99.71	99.13	99.38	15.29
2	300	300	1:2	99.60	99.68	0.50	0.20	99.80	99.07	99.41	43.61
3	500	500	1:4	99.68	99.80	0.25	0.60	99.06	99.07	99.20	79.43
4	1000	1000	1:9	99.78	99.90	0.16	0.80	99.20	98.65	98.89	175.02
5	2000	2000	1:9	99.95	99.95	0.06	0.00	100	99.52	99.76	640.50
6	3700	3700	1:18	99.98	100	0.02	0.10	99.90	99.71	99.80	1165.32
7	4000	4000	1:7	99.995	100	0.006	0.00	100	99.96	99.98	2583.44

Key: APA: Average Prediction Accuracy, FP: False Positive, FN: False Negative, R: Recall, Pr: Precision, GB: Global best, T: Time, FM: F-Measure, P:H: Phish:Ham

Table 4.12: Results obtained from the evaluation of O-SVM using 9 Features - 20 Runs

S/N	Dataset Information			Performance Evaluation							
	Total Email	Email Per Folder	P: H Ratio (%)	APA(%)	GB	FP(%)	FN(%)	R(%)	Pr(%)	FM(%)	T(s)
1	150	15	7:8	99.47	100	1.00	0.00	100	99.00	99.47	15.46
2	300	300	1:2	99.60	99.67	0.50	0.20	99.80	99.07	99.41	43.51
3	500	500	1:4	99.70	0.33	0.20	0.20	99.80	98.84	99.28	81.92
4	1000	1000	1:9	99.85	100	0.12	0.40	99.60	98.96	99.26	162.08
5	2000	2000	1:9	99.95	99.95	0.06	0.00	100	99.52	99.76	669.59
6	3700	3700	1:18	99.97	100	0.02	0.20	99.80	99.66	99.72	1115.92
7	4000	4000	1:7	99.98	100	0.03	0.00	100	99.84	99.92	2473.47

Key: APA: Average Prediction Accuracy, FP: False Positive, FN: False Negative, R: Recall, Pr: Precision, GB: Global best, T: Time, FM: F-Measure, P: H: Phish: Ham

Table 4.13: Results obtained from the evaluation of FFA_SVM using 8 Features - 10 Runs

S/N	Dataset Information			Performance Evaluation							
	Total Email	Email Per Folder	P: H Ratio (%)	APA(%)	GB	FP(%)	FN(%)	R(%)	Pr(%)	FM(%)	T(s)
1	150	15	7:8	99.47	100	0.63	0.43	99.57	99.38	99.44	14.72
2	300	300	1:2	99.67	99.67	0.50	0.40	99.60	99.05	99.30	46.13
3	500	500	1:4	99.76	99.80	0.30	0.00	100	98.94	99.44	72.07
4	1000	1000	1:9	99.92	100	0.09	0.00	100	99.27	99.62	139.25
5	2000	2000	1:9	99.95	99.95	0.06	0.00	100	99.52	99.76	555.46
6	3700	3700	1:18	99.97	100	0.02	0.20	99.80	99.61	99.70	1034.14
7	4000	4000	1:7	99.99	100	0.01	0.00	100	99.90	99.95	2376.33

Key: APA: Average Prediction Accuracy, FP: False Positive, FN: False Negative, R: Recall, Pr: Precision, GB: Global best, T: Time, FM: F-Measure, P: H: Phish: Ham

Table 4.14: Results obtained from the evaluation of FFA_SVM using 8 Features - 20 Runs

S/N	Dataset Information			Performance Evaluation							
	Total Email	Email Per Folder	P: H Ratio (%)	APA(%)	GB	FP(%)	FN(%)	R(%)	Pr(%)	FM(%)	T(s)
1	150	15	7:8	99.47	100	0.63	0.43	99.57	99.38	99.44	14.54
2	300	300	1:2	99.63	99.67	0.45	0.20	99.80	99.18	99.46	44.68
3	500	500	1:4	99.64	99.80	0.30	0.60	99.40	98.88	99.11	73.96
4	1000	1000	1:9	99.84	100	0.13	0.40	99.60	98.87	99.21	144.38
5	2000	2000	1:9	99.96	100	0.05	0.00	100	99.57	99.78	594.91
6	3700	3700	1:18	99.98	100	0.02	0.10	99.90	99.71	99.80	1070.96
7	4000	4000	1:7	99.99	100	0.01	0.00	100	99.90	99.95	2280.33

Key: APA: Average Prediction Accuracy, FP: False Positive, FN: False Negative, R: Recall, Pr: Precision, GB: Global best, T: Time, FM: F-Measure, P: H: Phish: Ham

Table 4.15: Results obtained from the evaluation of O-SVM using 8 Features - 10 Runs

S/N	Dataset Information			Performance Evaluation							
	Total Email	Email Per Folder	P: H Ratio (%)	APA(%)	GB	FP(%)	FN(%)	R(%)	Pr(%)	FM(%)	T(s)
1	150	15	7:8	99.60	100	0.75	0.00	100	99.25	99.60	14.01
2	300	300	1:2	99.67	99.67	0.50	0.00	100	99.09	99.52	42.38
3	500	500	1:4	99.72	99.80	0.25	0.40	99.60	99.05	99.30	73.95
4	1000	1000	1:9	99.90	100	0.10	0.40	99.60	99.14	99.35	155.58
5	2000	2000	1:9	99.95	99.95	0.06	0.00	100	99.52	99.76	593.56
6	3700	3700	1:18	99.99	100	0.01	0.00	100	99.76	99.88	1045.51
7	4000	4000	1:7	99.98	100	0.02	0.00	100	99.84	99.92	2268.31

Key: APA: Average Prediction Accuracy, FP: False Positive, FN: False Negative, R: Recall, Pr: Precision, GB: Global best, T: Time, FM: F-Measure, P: H: Phish: Ham

Table 4.16: Results obtained from the evaluation of O-SVM using 8 Features - 20 Runs

S/N	Dataset Information			Performance Evaluation							
	Total Email	Email Per Folder	P: H Ratio (%)	APA(%)	GB	FP(%)	FN(%)	R(%)	Pr(%)	FM(%)	T(s)
1	150	15	7:8	99.53	100	0.75	0.14	99.86	99.25	99.52	14.84
2	300	30	1:2	99.53	99.67	0.50	0.40	99.60	99.05	99.30	42.88
3	500	50	1:4	99.74	99.80	0.33	0.00	100	98.86	99.39	72.42
4	1000	100	1:9	99.91	100	0.08	0.20	99.80	99.34	99.56	147.70
5	2000	200	1:9	99.96	100	0.04	0.00	100	99.62	99.80	581.31
6	3700	370	1:18	99.96	100	0.03	0.30	99.70	99.56	99.63	1079.13
7	4000	400	1:7	99.998	100	0.002	0.00	100	99.98	99.99	2181.82

Key: APA: Average Prediction Accuracy, FP: False Positive, FN: False Negative, R: Recall, Pr: Precision, GB: Global best, T: Time, FM: F-Measure, P: H: Phish: Ham

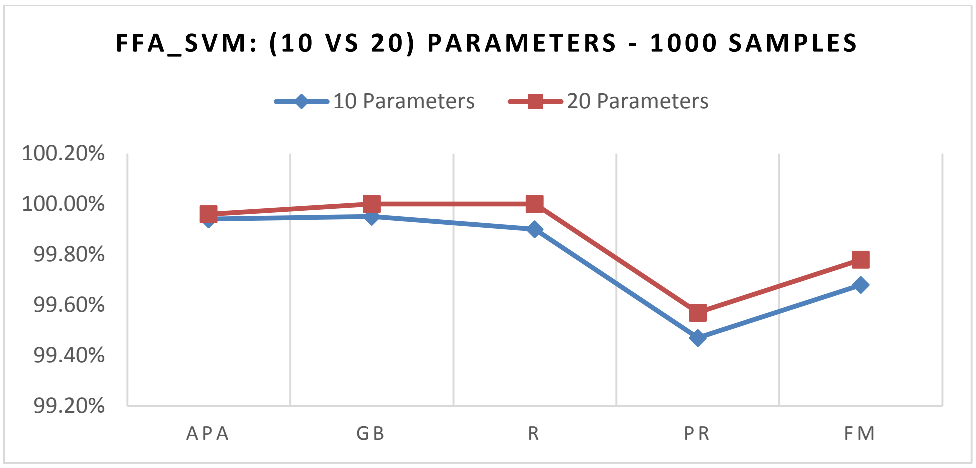

Figure 4.2: Effect of 10 and 20 SVM Parameters on FFA_SVM - 1000 Samples

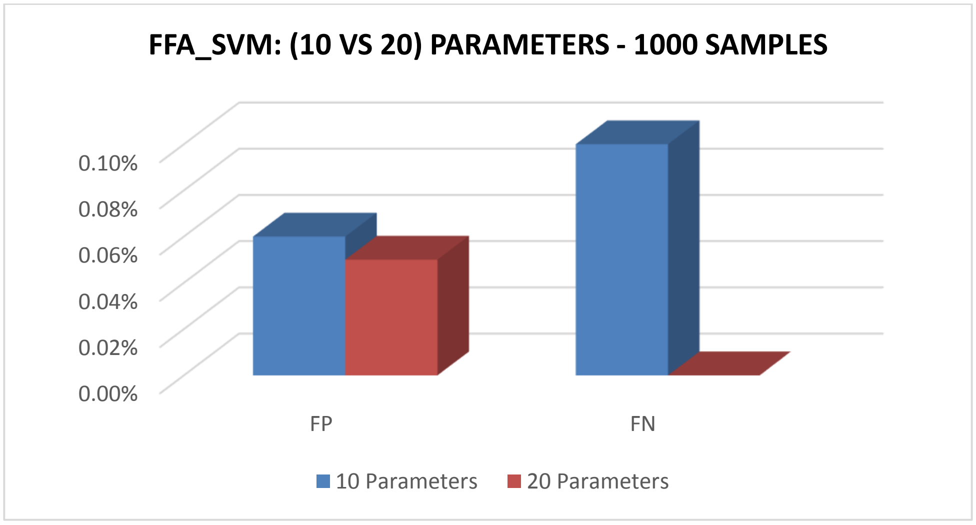

Figure 4.3: Effect of 10 and 20 SVM Parameters on FFA_SVM - 1000 Samples (FP and FN)

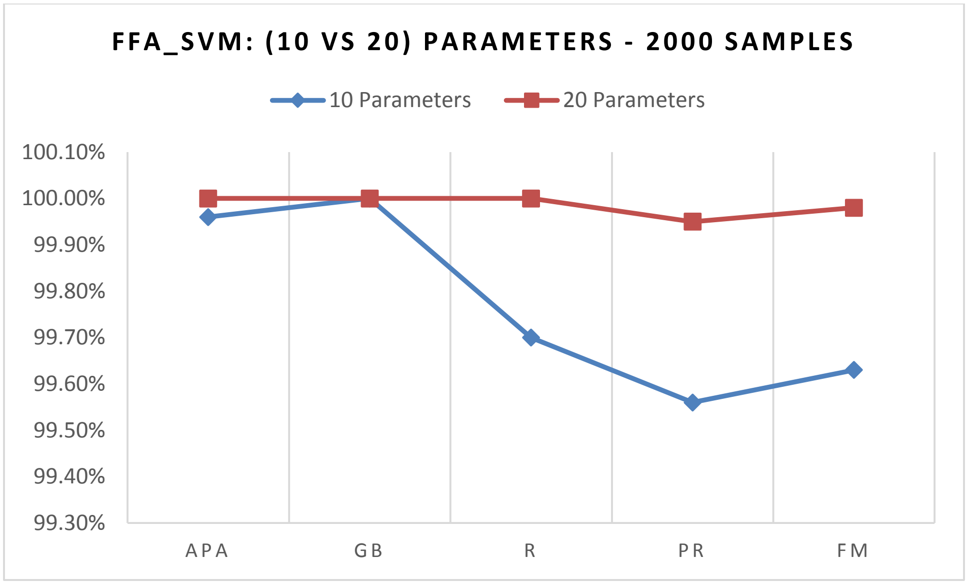

Figure 4.4: Effect of 10 and 20 SVM Parameters on FFA_SVM - 2000 Samples

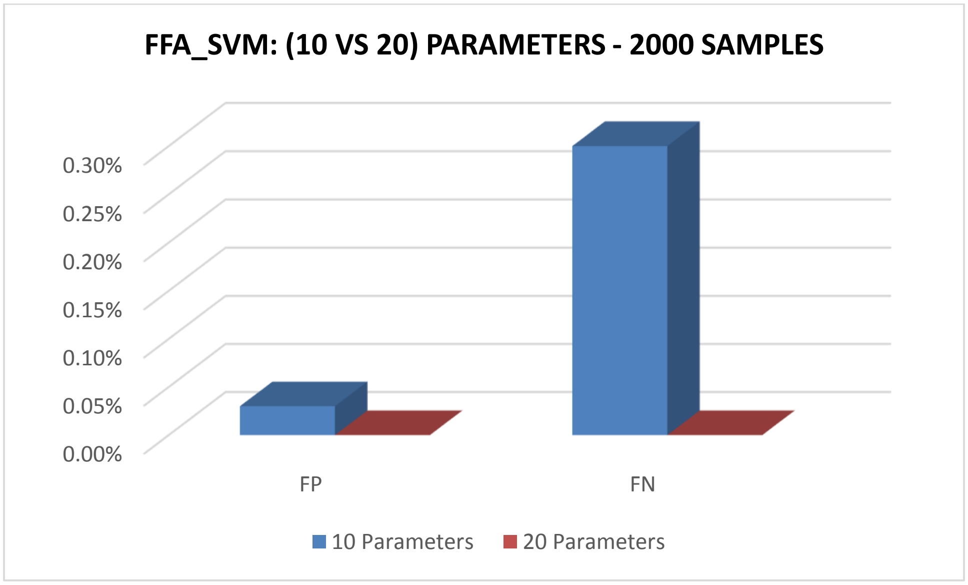

Figure 4.5: Effect of 10 and 20 SVM Parameters on FFA_SVM - 2000 Samples (FP and FN)

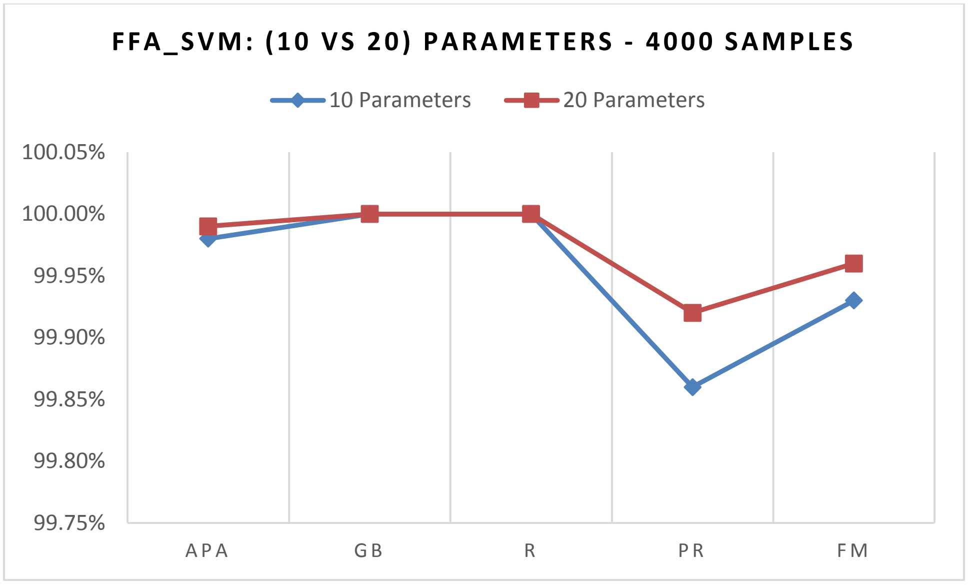

Figure 4.6: Effect of 10 and 20 SVM Parameters on FFA_SVM - 4000 Samples

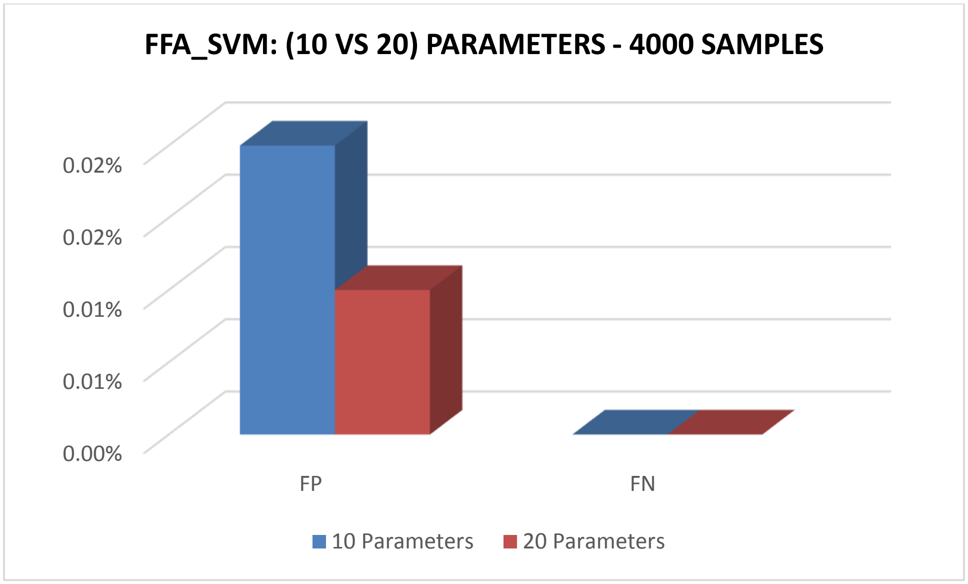

Figure 4.7: Effect of 10 and 20 SVM Parameters on FFA_SVM - 4000 Samples (FP and FN)

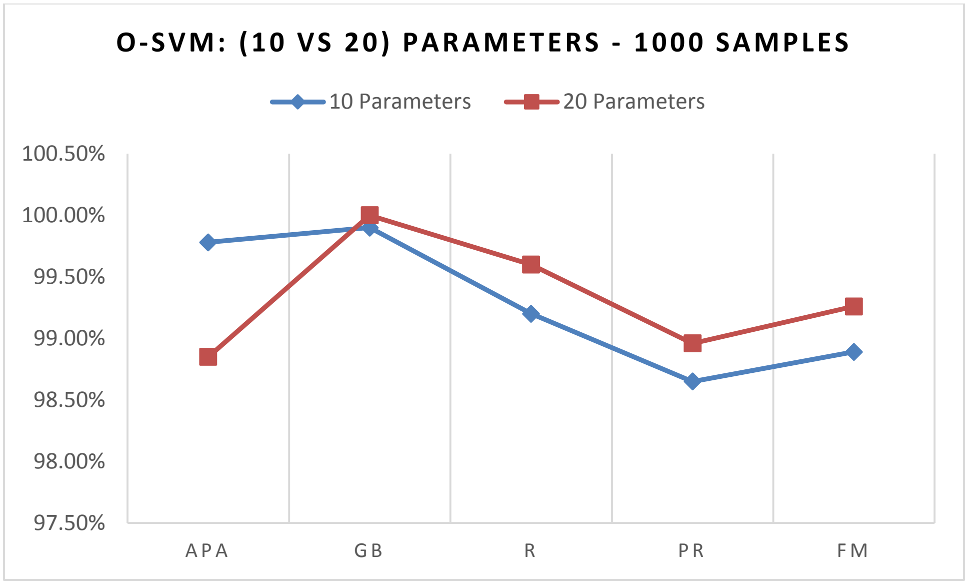

Figure 4.8: Effect of 10 and 20 SVM Parameters on O-SVM – 1000 Samples

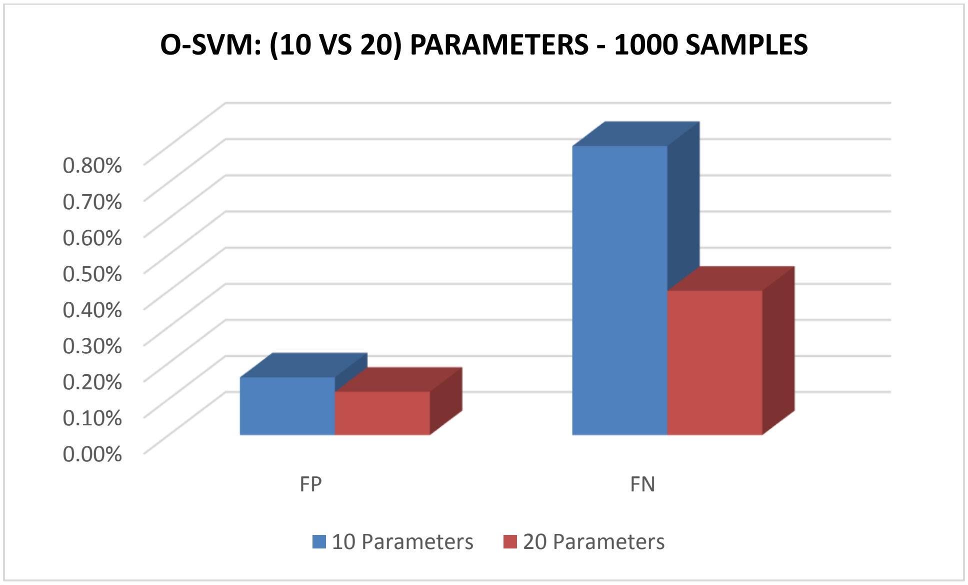

Figure 4.9: Effect of 10 and 20 SVM Parameters on O-SVM - 1000 Samples (FP and FN)

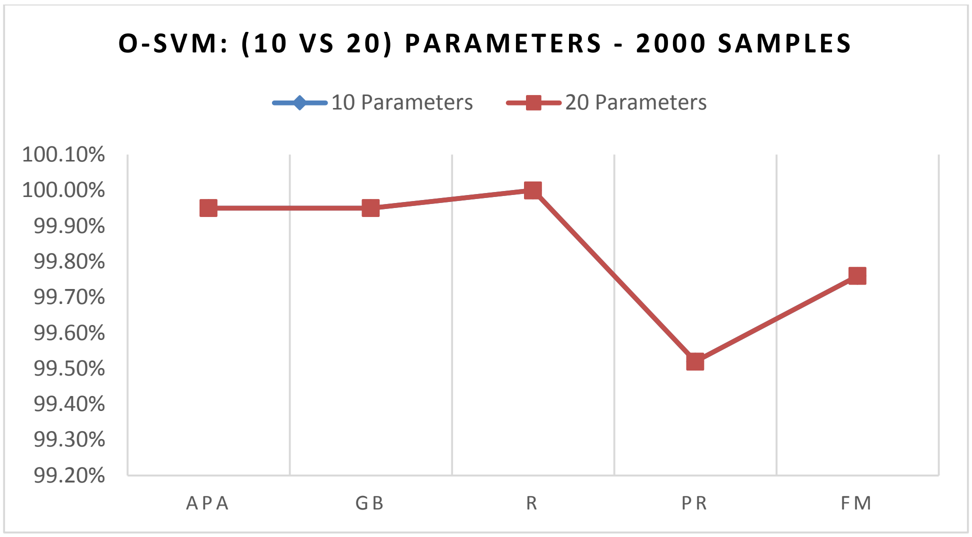

Figure 4.10: Effect of 10 and 20 SVM Parameters on O-SVM - 2000 Samples

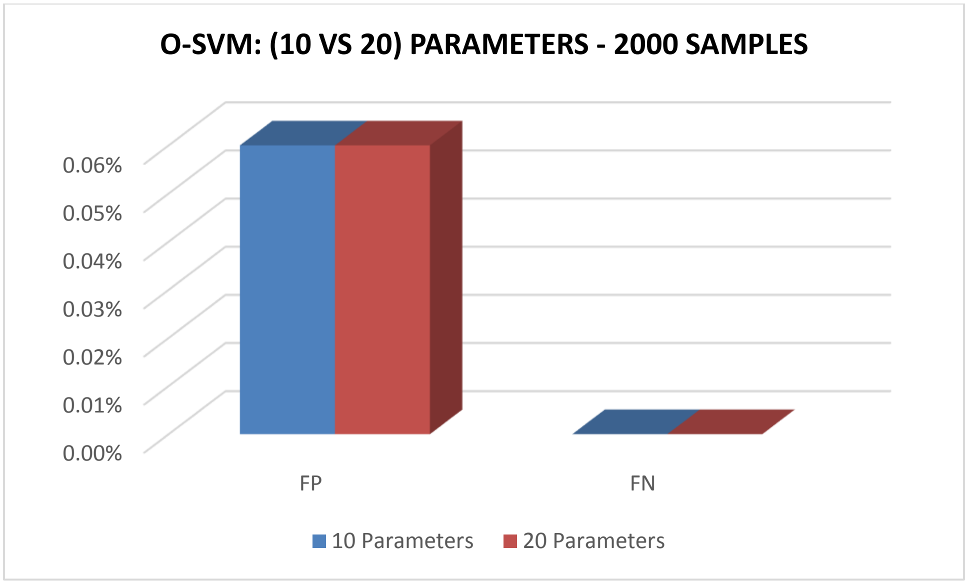

Figure 4.11: Effect of 10 and 20 SVM Parameters on O-SVM - 2000 Samples (FP and FN)

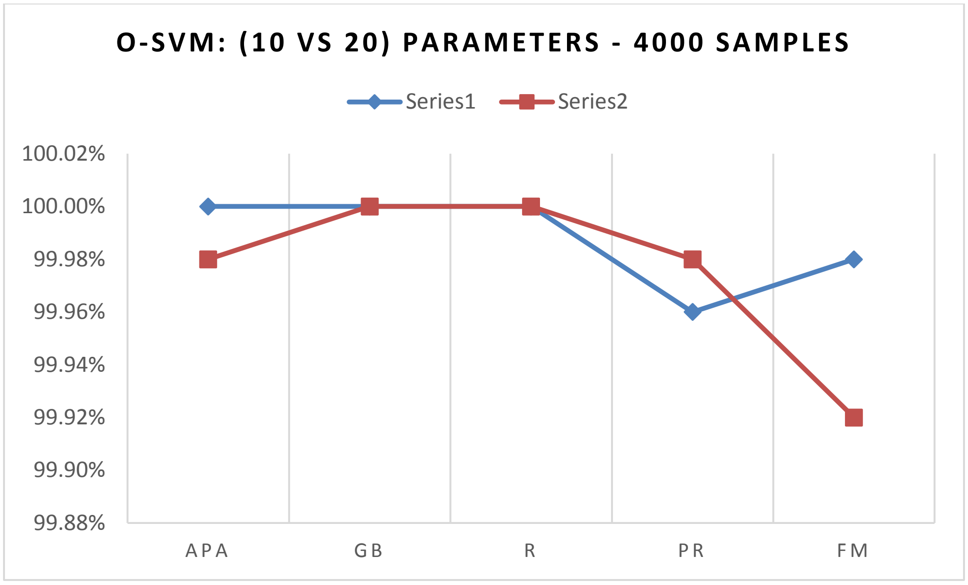

Figure 4.12: Effect of 10 and 20 SVM Parameters on O-SVM - 4000 Samples

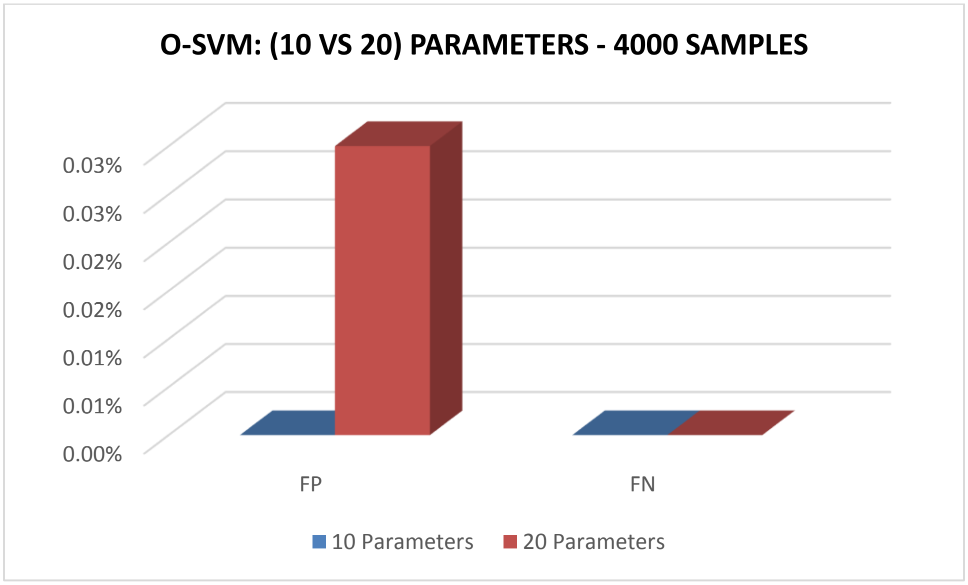

Figure 4.13: Effect of 10 and 20 SVM Parameters on O-SVM - 4000 Samples (FP and FN)

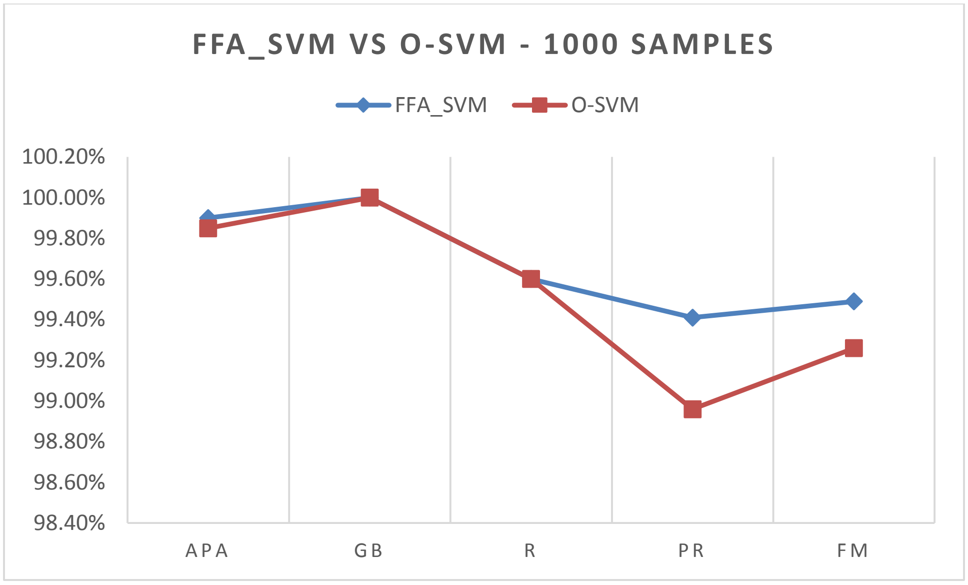

Figure 4.14: Comparison between FFA_SVM and O-SVM - 1000 Samples

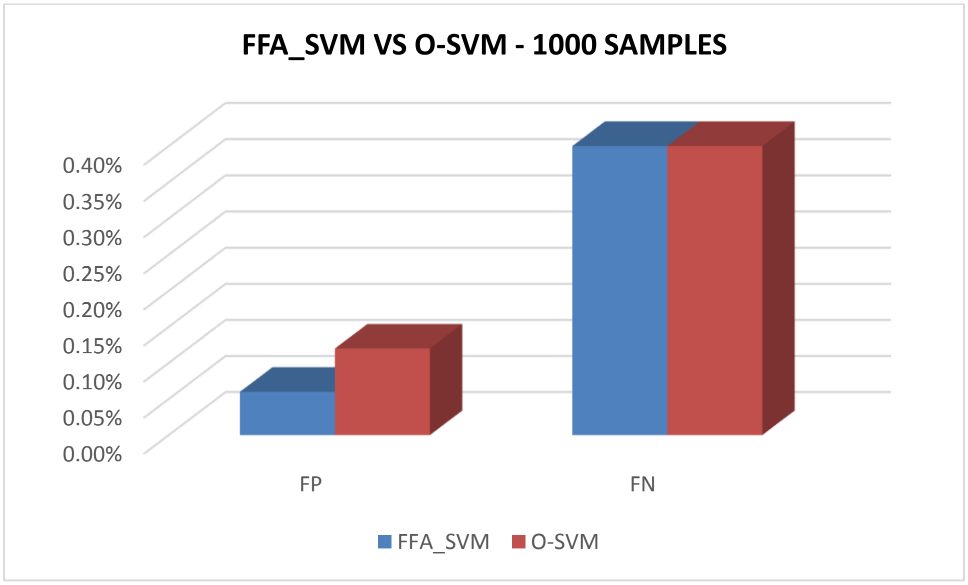

Figure 4.15: Comparison between FFA_SVM and O-SVM - 1000 Samples (FP and FN)

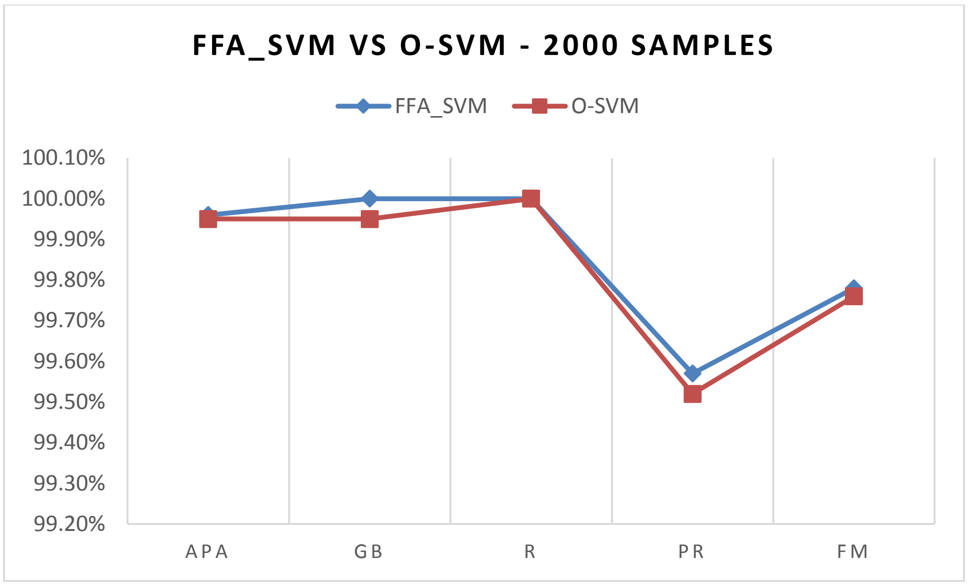

Figure 4.16: Comparison between FFA_SVM and O-SVM - 2000 Samples

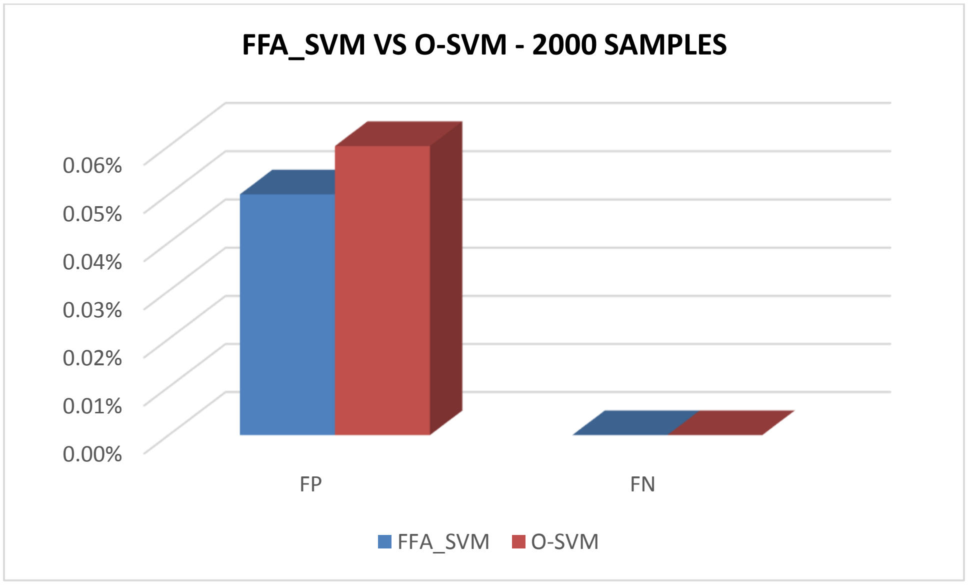

Figure 4.17: Comparison between FFA_SVM and O-SVM - 2000 Samples (FP and FN)

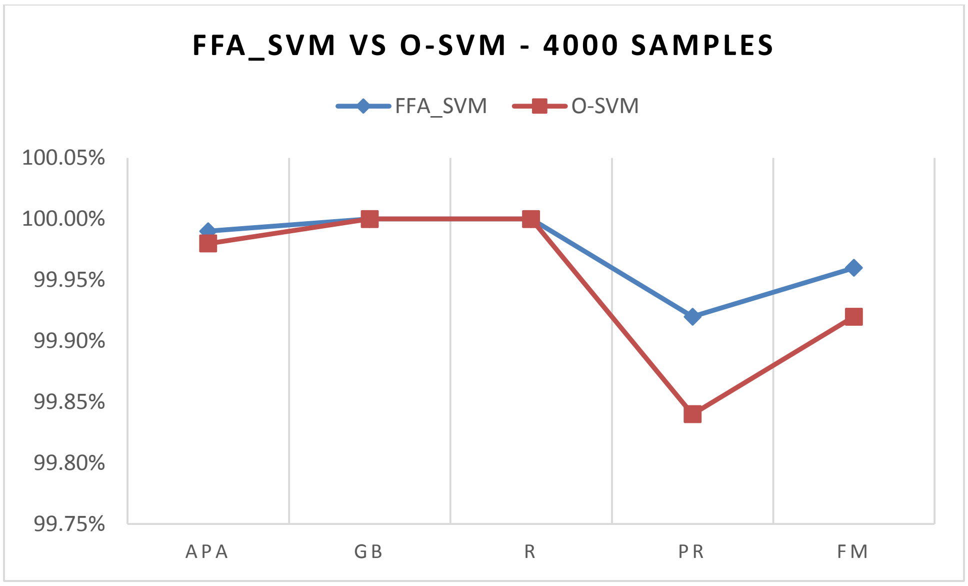

Figure 4.18: Comparison between FFA_SVM and O-SVM - 4000 Samples

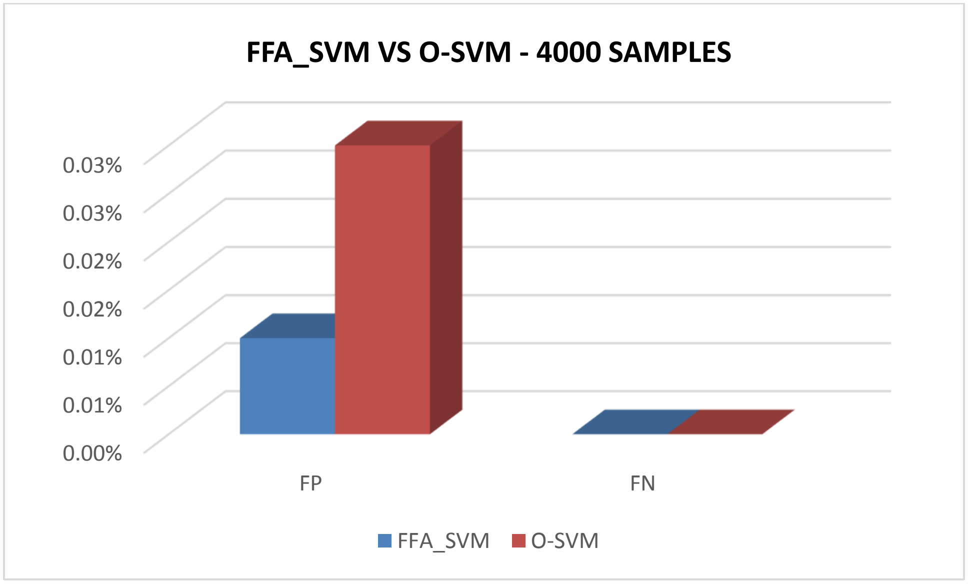

Figure 4.19: Comparison between FFA_SVM and O-SVM - 4000 Samples (FP and FN)

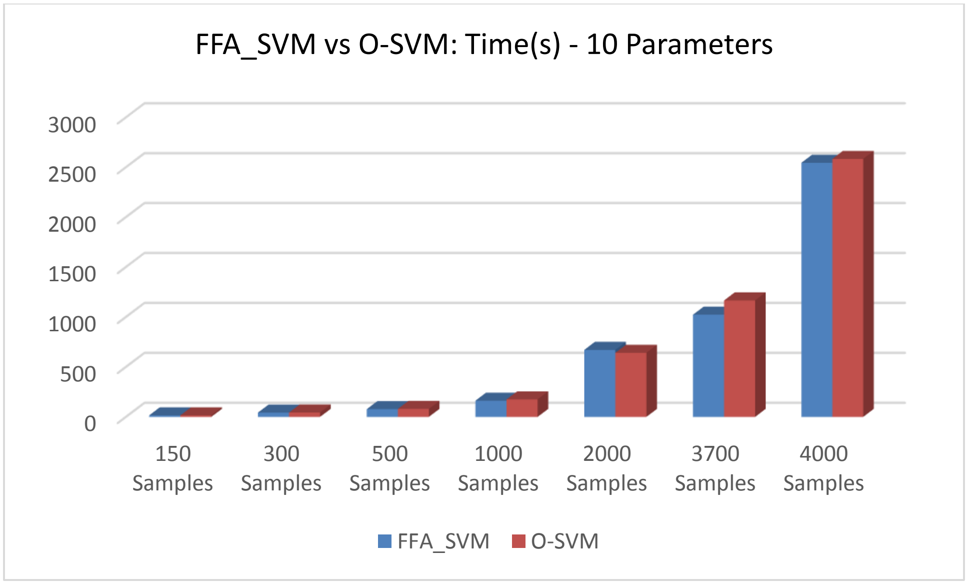

Figure 4.20: Time consumed by FFA_SVM and O-SVM (10 Parameters)

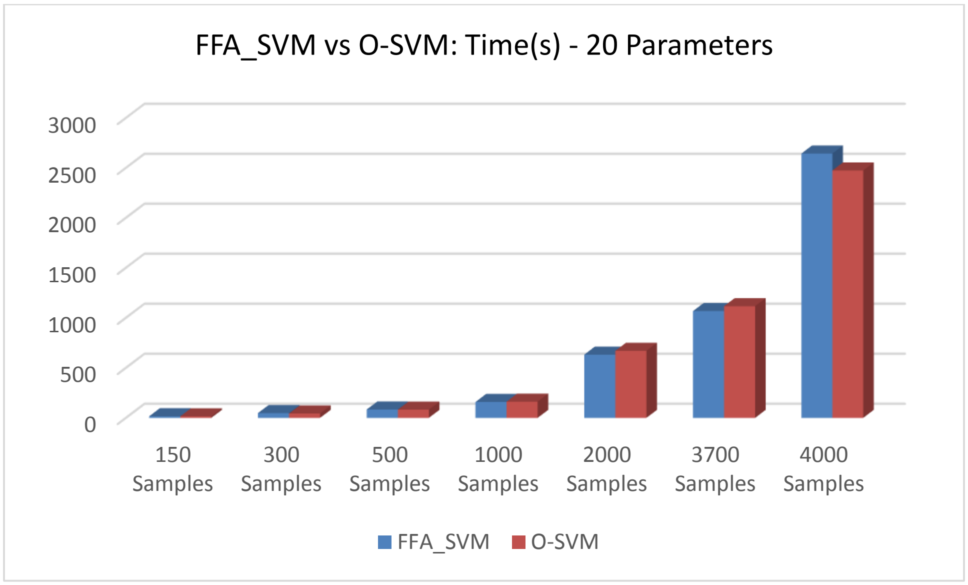

Figure 4.21: Time consumed by FFA_SVM and O-SVM (20 Parameters)

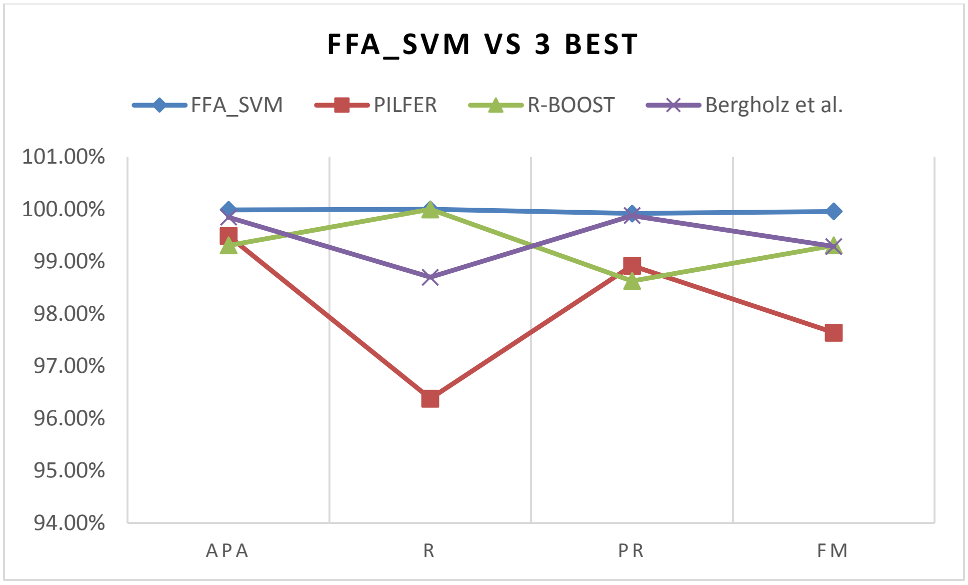

Figure 4.22: FFA_SVM vs PILFER, R-Boost and Bergholz et al.

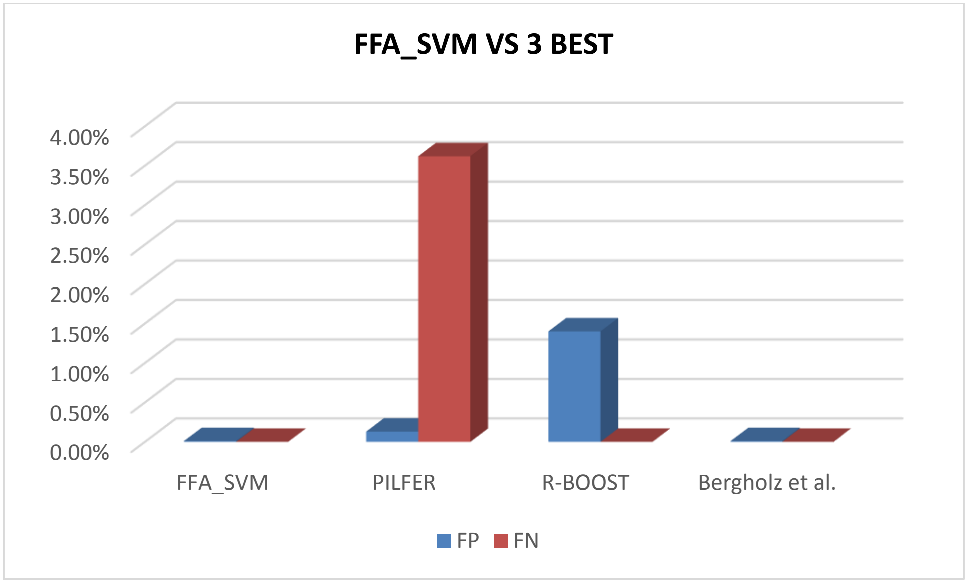

Figure 4.23: FFA_SVM vs PILFER, R-Boost and Bergholz et al (FP&FN).

4.4 Summary

In this chapter, the experimental setup for the RF-based technique and the firefly based technique is described. Furthermore, information on the dataset used to carry out the experiments is provided. Also, the results obtained from all the experiments that are carried out on the

RF-Based technique and the firefly based SVM technique are discussed in this chapter. All the results revealed that the two techniques compare fairly and better with some results in literature.

Chapter Five

Summary, Conclusion and Further Work

5.1 Summary

Much work has been done by the research community to significantly reduce phishing attacks with great accuracy. In this research, three popular types of e-fraud are reviewed alongside some proposed ML-based and NI-based techniques that are used for them. Furthermore, some basic concept on NI techniques, ML, ANN and email filtering are presented. Moreover, some general challenges of e-fraud detection are examined and some techniques that can be used to improve the performance of the examined detection techniques are presented. Additionally, two improved phishing email detection techniques are proposed. The first technique is based on RF algorithm, and the second technique is based on FFA and SVM (known as FFA_SVM). In the second technique, FFA is hybridized with SVM. FFA is used to search for the optimized hyper-parameter pair, which is then used by SVM for classification. To further enhance the speed and accuracy of FFA_SVM, a hyper-parameter selection algorithm is developed. This algorithm generates exponentially growing sequence of C and γ pair of values.

To evaluate the performance of the two techniques, several experiments are carried out on datasets of different sizes. The first set of experiments are carried out on the RF-based classifier and the second set of experiments are carried out on the firefly based SVM classifier. The experiments yielded excellent results. The RF-based classifier yielded a classification accuracy of 99.70%, while FFA_SVM yielded a classification accuracy of 99.99%. Also, FFA_SVM produced a FP-rate and FN-rate of 0.01% and 0.00% respectively, while the RF-based classifier produced a FP-rate and FN-rate of 0.06% and 2.50% respectively.

5.2 Conclusion

The urgent need for a robust, secure, fast and accurate e-fraud detection system cannot be overemphasized, because of the huge financial losses incurred by many organizations and individuals owing to online fraud. A rapid change in fraud patterns is one of the major challenges faced by researchers in the field of e-fraud detection. There are three major types of e-fraud, namely: email spam, phishing and network intrusion. This research focus on phishing detection. Phishing remains a serious threat to global security and economy, as many organizations and individuals worldwide have fallen victims. It is a lucrative trade that will keep blooming as technology advances. The fast rate of emergence of new phishing websites and distributed phishing attacks has made it difficult to keep blacklists up to date. Therefore, a detection technique that can adapt to new fraud patterns is highly required. In this research, two phishing email detection classifiers are developed based on RF, FFA and SVM, and the results of all the experiments performed on both classifiers reveals that they have very high filtering capability and they are capable of accurately identifying phishing emails. The classifiers are also capable of adapting to new or emerging phishing trends. Moreover, the two proposed phishing email techniques can be integrated with existing email filters, which will no doubt reduce phishing attacks to the barest minimum and also ensure a secured online environment for organizations and users engaged in various online transactions.

5.3 Future Work

Due to the rapid change in phishing attack patterns, current phishing detection techniques need to be greatly enhanced in order to effectively combat emerging phishing attacks. In the course of this research, it was observed that the classification speed of SVM is very slow, especially when classifying dataset with large volume of data. This is because, for the classification of one data sample, the kernel function is, usually, evaluated for each of the identified support vectors [186]. This is not acceptable, especially for fraud detection systems that require real-time processing. For the future, a plan is in place to come up with an improved model for SVM. In the model, our primary concern will be how to reduce the time consumed during the training process. We also plan to hybridize more NI techniques using ML techniques. The hybridized systems could yield better results.

REFERENCES

[1] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," Proceedings of the 16th international conference on World Wide Web, pp. 649-656, 2007.

[2] M. Behdad, L. Barone, M. Bennamoun, and T. French, "Nature-Inspired Techniques in the Context of Fraud Detection," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 42, pp. 1273-1290, 2012.

[3] Symantec. (2011), "Norton Study Calculates Cost of Global Cybercrime: $114 Billion Annually". available at: http://www.symantec.com/about/news/release/article.jsp?prid=20110907_02 (accessed 26 June 2014).

[4] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "Phishnet: predictive blacklisting to detect phishing attacks," INFOCOM 2010: Proceedings of the 29th conference on Information communications, pp. 346-350, 2010.

[5] M. Hara, A. Yamada, and Y. Miyake, "Visual similarity-based phishing detection without victim site information," Computational Intelligence in Cyber Security, 2009. CICS'09. IEEE Symposium on, pp. 30-36, 2009.

[6] A. Bergholz, J. H. Chang, G. Paaß, F. Reichartz, and S. Strobel, "Improved Phishing Detection using Model-Based Features," Proceedings of the Conference on Email and Anti-Spam (CEAS), 2008.

[7] R. Jensen and Q. Shen, "Fuzzy-rough data reduction with ant colony optimization," Fuzzy Sets and Systems, vol. 149, pp. 5-20, 2005.

[8] H.-H. Gao, H.-H. Yang, and X.-Y. Wang, "Ant colony optimization based network intrusion feature selection and detection," Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on, vol. 6, pp. 3871-3875, 2005.

[9] A. R. Behjat, A. Mustapha, H. Nezamabadi-pour, M. N. Sulaiman, and N. Mustapha, "A PSO-Based Feature Subset Selection for Application of Spam/Non-spam Detection," Soft Computing Applications and Intelligent Systems, vol. 378, pp. 183-193, 2013.

[10] S. Srinoy, "Intrusion detection model based on particle swarm optimization and support vector machine," Proceedings of the 2007 IEEE symposium on computational intelligence in security and defense applications (CISDA 2007), pp. 186-192, 2007.

[11] K. Lamond. (1996), "Credit Card Transactions Real World and Online". available at: http://www.virtualschool.edu/mon/ElectronicProperty/klamond/credit_card.htm (accessed 18 September 2014).

[12] RSA. (2014), "2013 A Year in Review". available at: http://www.emc.com/collateral/fraud-report/rsa-online-fraud-report-012014.pdf (accessed 05 August 2014).

[13] APWG. (2013), "Phishing Activity Trends Report". available at: http://docs.apwg.org/reports/apwg_trends_report_q3_2013.pdf (accessed 05 August 2014).

[14] J. KING. (2011), "10 Most Common Types of Internet Scams". available at: http://blogs.lawyers.com/2011/08/10-most-common-types-of-internet-scams/ (accessed 08-09-2014).

[15] M. Khonji, Y. Iraqi, and A. Jones, "Phishing Detection: A Literature Survey," IEEE Communications on Surveys & Tutorials, vol. 15, pp. 2091-2121, 2013.

[16] D. L. Cook, V. K. Gurbani, and M. Daniluk, "Phishwish: a stateless phishing filter using minimal rules," Financial Cryptography and Data Security, pp. 182-186, 2008.
[17] H. Zhang, G. Liu, T. W. Chow, and W. Liu, "Textual and visual content-based anti-phishing: a Bayesian approach," Neural Networks, IEEE Transactions on, vol. 22, pp. 1532-1546, 2011.
[18] C. Whittaker, B. Ryner, and M. Nazif, "Large-Scale Automatic Classification of Phishing Pages," NDSS, 2010.
[19] Y. Cao, W. Han, and Y. Le, "Anti-phishing based on automated individual white-list," Proceedings of the 4th ACM workshop on Digital identity management, pp. 51-60, 2008.
[20] L. Ma, B. Ofoghi, P. Watters, and S. Brown, "Detecting phishing emails using hybrid features," Ubiquitous, Autonomic and Trusted Computing, 2009. UIC-ATC'09. Symposia and Workshops on, pp. 493-497, 2009.
[21] M. Mitchell and C. E. Taylor, "Evolutionary computation: an overview," Annual Review of Ecology and Systematics, vol. 30, pp. 593-616, 1999.
[22] D. Dasgupta and Z. Michalewicz, *Evolutionary algorithms in engineering applications*: Springer, 1997.
[23] Y.-f. Zhu and X.-m. Tang, "Overview of swarm intelligence," Computer Application and System Modeling (ICCASM), 2010 International Conference on, vol. 9, pp. V9-400-V9-403, 2010.
[24] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," IEEE computer, vol. 29, pp. 31-44, 1996.
[25] J. Mason, "The apache spamassassin public corpus," 2005.
[26] J. Nazario. (2006), "Phishing Corpus Homepage". available at: http://monkey.org/~jose/wiki/doku.php?id=phishingcorpus (accessed 05 August 2014).
[27] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," 2003.
[28] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of IEEE international conference on neural networks, vol. 4, pp. 1942-1948, 1995.
[29] X.-S. Yang, *Nature-inspired metaheuristic algorithms*: Luniver press, 2010.
[30] M. Batouche and S. Meshoul, "Nature Inspired Intelligent Techniques for Problem Solving."
[31] D. L. Applegate, "The traveling salesman problem: a computational study," 2006.
[32] T. R. Jensen and B. Toft, "Graph coloring problems," vol. 39, 2011.
[33] M. N. Arbatskaya, K. Mukhopadhaya, and E. B. Rasmusen, "The parking lot problem," Available at SSRN 571101, 2006.
[34] A. O. Adewumi and M. M. Ali, "A multi-level genetic algorithm for a multi-stage space allocation problem," Mathematical and Computer Modelling, vol. 51, pp. 109-126, 2010.
[35] I. K. Nikolos, K. P. Valavanis, N. C. Tsourveloudis, and A. N. Kostaras, "Evolutionary algorithm based offline/online path planner for UAV navigation," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 33, pp. 898-912, 2003.
[36] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments-a survey," Evolutionary Computation, IEEE Transactions on, vol. 9, pp. 303-317, 2005.
[37] P. J. Fleming and R. C. Purshouse, "Evolutionary algorithms in control systems engineering: a survey," Control engineering practice, vol. 10, pp. 1223-1241, 2002.
[38] S. W. Wilson, "Classifier fitness based on accuracy," Evolutionary computation, vol. 3, pp. 149-175, 1995.

[39] T. O. Ayodele, "Types of machine learning algorithms," New Advances in Machine Learning, pp. 19-48, 2010.
[40] Wikipedia. "Machine learning". available at: http://en.wikipedia.org/wiki/Machine_Learning (accessed 05 August 2014).
[41] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," arXiv preprint cs/9605103, 1996.
[42] A. G. Barto, "Reinforcement learning: An introduction," 1998.
[43] V. Sharma, S. Rai, and A. Dev, "A Comprehensive Study of Artificial Neural Networks," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, pp. 278-284, 2012.
[44] Q. J. Zhang and K. C. Gupta, "Neural Networks for RF and Microwave Design," p. 392, 2000.
[45] Wikipedia. "Email filtering". available at: http://en.wikipedia.org/wiki/Email_filtering (accessed 31 July 2014).
[46] A. Bergholz, J. De Beer, S. Glahn, M.-F. Moens, G. Paaß, and S. Strobel, "New filtering approaches for phishing email," Journal of computer security, vol. 18, pp. 7-35, 2010.
[47] T. N. Report. (2013), "The Nelson Report". available at: http://www.nilsonreport.com/publication_newsletter_archive_issue.php?issue=1023 (accessed 31 July 2014).
[48] T. U. C. Association. "The UK Cards Association". available at: http://www.theukcardsassociation.org.uk/news/EOYFFfor2013.asp (accessed 31 July 2014).
[49] A. B. O. Statistics. (2008), "Personal fraud, 2007". available at: http://www.abs.gov.au/ausstats/abs@.nsf/cat/4528.0 (accessed 09 September 2014).
[50] T. Fawcett, "In vivo spam filtering: a challenge problem for KDD," ACM SIGKDD Explorations Newsletter, vol. 5, pp. 140-148, 2003.
[51] K. Shafi and H. A. Abbass, "Biologically-inspired complex adaptive systems approaches to network intrusion detection," Information Security Technical Report, vol. 12, pp. 209-217, 2007.
[52] M. Jakobsson and S. Myers, "Phishing and countermeasures: understanding the increasing problem of electronic identity theft," 2007.
[53] L. Khan, M. Awad, and B. Thuraisingham, "A new intrusion detection system using support vector machines and hierarchical clustering," The VLDB Journal, vol. 16, pp. 507-521, 2007.
[54] K. Gao. (2012), "What is Email Spam". available at: http://emailmarketing.comm100.com/email-marketing-ebook/email-spam.aspx (accessed 31 July 2014).
[55] Z. Duan, Y. Dong, and K. Gopalan, "A differentiated message delivery architecture to control spam," Parallel and Distributed Systems, 2005. Proceedings. 11th International Conference on, vol. 2, pp. 255-259, 2005.
[56] B. C. Johnson. (2003), "Intrusions and their Detection: Addressing Common Hacker Exploits". available at: http://systemexperts.com/media/pdf/hackerid.pdf (accessed 18 September 2014).
[57] J. S. White, J. N. Matthews, and J. L. Stacy, "A method for the automated detection phishing websites through both site characteristics and image analysis," SPIE Defense, Security, and Sensing, pp. 84080B-84080B-11, 2012.

[58] A. Martin, N. Anutthamaa, M. Sathyavathy, M. M. S. Francois, and P. Venkatesan, "A Framework for Predicting Phishing Websites Using Neural Networks," International Journal of Computer Science Issues (IJCSI), vol. 8, pp. 330-336, 2011.

[59] B. Adida, S. Hohenberger, and R. L. Rivest, "Lightweight encryption for email," In Proceedings of the USENIX Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI 2005), pp. 93-99, 2005.

[60] R. Dhamija, J. D. Tygar, and M. Hearst, "Why phishing works," Proceedings of the SIGCHI conference on Human Factors in computing systems (CHI), vol. 1, pp. 581-590, 2006.

[61] R. Dhamija and J. D. Tygar, "The battle against phishing: Dynamic security skins," Proceedings of the 2005 symposium on Usable privacy and security, pp. 77-88, 2005.

[62] W. L. Buntine, "A theory of learning classification rules," Doctoral dissertation, School of Computing Science, University of Technology, Sydney, Australia., 1992.

[63] A. Almomani, T.-C. Wan, A. Altaher, A. Manasrah, E. ALmomani, M. Anbar, *et al.*, "Evolving Fuzzy Neural Network for Phishing Emails Detection," Journal of Computer Science, vol. 8, pp. 1099–1107, 2012.

[64] F. Schneider, N. Provos, R. Moll, M. Chew, and B. Rakowski. (2007), "Phishing Protection Design Documentation". available at: http://wiki.mozilla.org/PhishingProtection:_Design_Documentation (accessed 18 September 2014).

[65] M. Aburrous, M. A. Hossain, K. Dahal, and F. Thabatah, "Modelling intelligent phishing detection system for e-banking using fuzzy data mining," International Conference on CyberWorlds, pp. 265-272, 2009.

[66] C. Zhan, F. Zhang, and M. Zheng, "Design and Implementation of an Optimization System of Spam Filter Rule Based On Neural Network," Communications, Circuits and Systems, 2007. ICCCAS 2007. International Conference on, pp. 882-886, 2007.

[67] Y. Yang, "Anti-spam filtering using neural networks and Baysian classifiers," 2007 International Symposium on Computational Intelligence in Robotics and Automation, pp. 272-278, 2007.

[68] N. Chou, R. Ledesma, Y. Teraguchi, and J. C. Mitchell, "Client-Side Defense Against Web-Based Identity Theft," Proceedings of the 11th Annual Network and Distributed System Security Symposium, 2004.

[69] W. D. Yu, S. Nargundkar, and N. Tiruthani, "Phishcatch-a phishing detection tool," Computer Software and Applications Conference, 2009. COMPSAC'09. 33rd Annual IEEE International, vol. 2, pp. 451-456, 2009.

[70] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: a content-based approach to detecting phishing web sites," Proceedings of the 16th international conference on World Wide Web, pp. 639-648, 2007.

[71] N. Zhang and Y. Yuan, "Phishing Detection Using Neural Network."

[72] C. Group. (2006), "SpamAssassin Data". available at: http://www.csmining.org/index.php/spam-assassin-datasets.html (accessed 05 August 2014).

[73] J. Shun and H. A. Malki, "Network intrusion detection system using neural networks," Natural Computation, 2008. ICNC'08. Fourth International Conference on, vol. 5, pp. 242-246, 2008.

[74] P. Kachurka and V. Golovko, "Neural network approach to real-time network intrusion detection and recognition," Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2011 IEEE 6th International Conference on, vol. 1, pp. 393-397, 2011.

[75] H. Pohlheim, "Evolutionary algorithms: overview, methods and operators," GEATbx: Gentic & Evolutionary AlgorithmToolbox for Matlab, 2005.

[76] J. Dudley, L. Barone, and L. While, "Multi-objective spam filtering using an evolutionary algorithm," Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on, pp. 123-130, 2008.

[77] U. Sanpakdee, A. Walairacht, and S. Walairacht, "Adaptive spam mail filtering using genetic algorithm," Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference, vol. 1, pp. 441-445, 2006.

[78] W. W. Cohen. (2009), "Enron email dataset". (accessed 18 September 2014).

[79] V. Shreeram, M. Suban, P. Shanthi, and K. Manjula, "Anti-phishing detection of phishing attacks using genetic algorithm," Communication Control and Computing Technologies (ICCCCT), 2010 IEEE International Conference on, pp. 447-450, 2010.

[80] R. H. Gong, M. Zulkernine, and P. Abolmaesumi, "A software implementation of a genetic algorithm based approach to network intrusion detection," Proceedings of the sixth international conference on software engineering, artificial intelligence, networking and parallel/distributed computing and first ACIS international workshop on self-assembling wireless networks (SNPD/SAWN‘05), pp. 246-253, 2005.

[81] M. Crosbie and G. Spafford, "Applying genetic programming to intrusion detection," Working Notes for the AAAI Symposium on Genetic Programming, pp. 1-8, 1995.

[82] M. Dorigo and M. Birattari, "Swarm intelligence," Scholarpedia, vol. 2, p. 1462, 2007.

[83] R. S. Parpinelli and H. S. Lopes, "New inspirations in swarm intelligence: a survey," International Journal of Bio-Inspired Computation, vol. 3, pp. 1-16, 2011.

[84] M. Dorigo and M. Birattari, "Ant colony optimization," Encyclopedia of Machine Learning, pp. 36-39, 2010.

[85] M. Dorigo, "Optimization, learning and natural algorithms," Ph. D. Thesis, Politecnico di Milano, Italy, 1992.

[86] C. Kolias, G. Kambourakis, and M. Maragoudakis, "Swarm intelligence in intrusion detection: A survey," computers & security, vol. 30, pp. 625-642, 2011.

[87] K. Kameyama, "Particle swarm optimization-a survey," IEICE transactions on information and systems, vol. 92, pp. 1354-1361, 2009.

[88] E.-S. El-Alfy, "Discovering classification rules for email spam filtering with an ant colony optimization algorithm," Evolutionary Computation, 2009. CEC'09. IEEE Congress on, pp. 1778-1783, 2009.

[89] K. Bache and M. Lichman, "UCI machine learning repository," Irvine, CA: University of California, School of Information and Computer Science, 2013.

[90] H. Yin, F. Cheng, and D. Zhang, "Using LDA and ant colony algorithm for spam mail filtering," 2009 Second International Symposium on Information Science and Engineering, pp. 368-371, 2009.

[91] D. Radha and M. Valarmathi, "Experimental Study on Meta Heuristic Optimization Algorithms for Fake Website Detection " International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS), vol. 1, pp. 43-53, 2012.

[92] M. Radha Damodaram and M. Valarmathi, "Phishing Website Detection and Optimization Using Particle Swarm Optimization Technique," International Journal of Computer Science and Security (IJCSS), vol. 5, p. 477, 2011.

[93] M. Abadi and S. Jalili, "An ant colony optimization algorithm for network vulnerability analysis," Iranian Journal of Electrical and Electronic Engineering, vol. 2, pp. 106-120, 2006.

[94] Q. Zhang and W. Feng, "Network intrusion detection by support vectors and ant colony," Proceedings of the 2009 International Workshop on Information Security and Application (IWISA 2009), 2009.

[95] A. R. Behjat, A. Mustapha, H. Nezamabadi-Pour, M. N. Sulaiman, and N. Mustapha, "Feature Subset Selection Using Binary Quantum Particle Swarm Optimization for Spam Detection System," Advanced Science Letters, vol. 20, pp. 188-192, 2014.

[96] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, pp. 1-16, 2012.

[97] R. Sumathi and M. R. V. Prakash, "Prediction of Phishing Websites Using Optimization Techniques," International Journal of Modern Engineering Research (IJMER) vol. 2, pp. 341-348, 2012.

[98] M. Pandey and V. Ravi, "Phishing Detection Using PSOAANN Based One-Class Classifier," Iternational Conference on Emerging Trends in Engineering and Technology (ICETET), pp. 148-153, 2013.

[99] Z. Chen and P. Qian, "Application of PSO-RBF neural network in network intrusion detection," Intelligent Information Technology Application, 2009. IITA 2009. Third International Symposium on, vol. 1, pp. 362-364, 2009.

[100] X. Yi, P. Wu, D. Dai, L. Liu, and X. He, "Intrusion Detection Using BP Optimized by PSO," International Journal of Advancements in Computing Technology, vol. 4, 2012.

[101] H. Saxena and V. Richariya, "Intrusion Detection System using K-means, PSO with SVM Classifier: A Survey," International Journal of Emerging Technology and Advanced Engineering, vol. 4, pp. 653-657, 2014.

[102] N. Sanglerdsinlapachai and A. Rungsawang, "Web phishing detection using classifier ensemble," Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services, pp. 210-215, 2010.

[103] Z. Yang, A. Karahoca, N. Yang, and N. Aydin, "Network intrusion detection by using cellular neural network with tabu search," Bio-inspired Learning and Intelligent Systems for Security, 2008. BLISS'08. ECSIS Symposium on, pp. 64-68, 2008.

[104] Y. Wang, D. Gu, W. Li, H. Li, and J. Li, "Network intrusion detection with workflow feature definition using bp neural network," Advances in Neural Networks–ISNN 2009, pp. 60-67, 2009.

[105] M. Castellano, G. Mastronardi, and G. Tarricone, "Intrusion detection using neural networks: A grid computing based data mining approach," Neural Information Processing, pp. 777-785, 2009.

[106] M. S. Abadeh, J. Habibi, and C. Lucas, "Intrusion detection using a fuzzy genetics-based learning algorithm," Journal of Network and Computer Applications, vol. 30, pp. 414-428, 2007.

[107] S. M. Bridges and R. B. Vaughn, "Fuzzy data mining and genetic algorithms applied to intrusion detection," Proceedings twenty third National Information Security Conference, pp. 13-31, 2000.

[108] J. Gomez and D. Dasgupta, "Evolving fuzzy classifiers for intrusion detection," Proceedings of the 2002 IEEE Workshop on Information Assurance, vol. 6, pp. 321-323, 2002.

[109] A. Orfila, J. M. Estevez-Tapiador, and A. Ribagorda, "Evolving high-speed, easy-to-understand network intrusion detection rules with genetic programming," Applications of Evolutionary Computing, pp. 93-98, 2009.

[110] G. Suarez-Tangil, E. Palomar, J. M. de Fuentes, J. Blasco, and A. Ribagorda, "Automatic rule generation based on genetic programming for event correlation," Computational Intelligence in Security for Information Systems, pp. 127-134, 2009.

[111] K. Satpute, S. Agrawal, J. Agrawal, and S. Sharma, "A Survey on Anomaly Detection in Network Intrusion Detection System Using Particle Swarm Optimization Based Machine Learning Techniques," Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA), pp. 441-452, 2013.

[112] C.-C. Lai and C.-H. Wu, "Particle swarm optimization-aided feature selection for spam email classification," Proceedings of the Second International Conference on Innovative Computing, Information and Control, pp. 165-168, 2007.

[113] B. Xue, "Particle Swarm Optimisation for Feature Selection in Classification," 2014.

[114] X. Yue, A. Abraham, Z.-X. Chi, Y.-Y. Hao, and H. Mo, "Artificial immune system inspired behavior-based anti-spam filter," Soft Computing, vol. 11, pp. 729-740, 2007.

[115] C. R. Haag, G. B. Lamont, P. D. Williams, and G. L. Peterson, *An artificial immune system-inspired multiobjective evolutionary algorithm with application to the detection of distributed computer network intrusions*: Springer, 2007.

[116] K. Tamee, P. Rojanavasu, S. Udomthanapong, and O. Pinngern, "Using self-organizing maps with learning classifier system for intrusion detection," PRICAI 2008: Trends in Artificial Intelligence, pp. 1071-1076, 2008.

[117] M. Behdad, L. Barone, T. French, and M. Bennamoun, "On XCSR for electronic fraud detection," Evolutionary Intelligence, vol. 5, pp. 139-150, 2012.

[118] N. Horning, "Introduction to decision trees and random forests," American Museum of Natural History's, 2013.

[119] L. Breiman and A. Cutler, "Random forests–classification description," Department of Statistics Homepage, 2007.

[120] M. D. Steinberg and N. S. Cardell, "A Brief Overview to RandomForests," Salford Systems, 2004.

[121] Wikipedia. (2014), "Naive Bayes classifier. Wikipedia, the free encyclopedia". available at: http://en.wikipedia.org/wiki/Naive_Bayes_classifier (accessed 10 September 2014).

[122] Q. Wang, Y. Guan, and X. Wang, "SVM-Based Spam Filter with Active and Online Learning," TREC '06: Proceedings of the 15th Text Retrieval Conference, 2006.

[123] J. Moon, T. Shon, J. Seo, J. Kim, and J. Seo, "An Approach for Spam E-mail Detection with Support Vector Machine and n-Gram Indexing," vol. 3280, pp. 351-362, 2004.

[124] T. Chi-Yao and C. Ming-Syan, "Incremental SVM Model for Spam Detection on Dynamic Email Social Networks," Computational Science and Engineering, 2009. CSE '09. , vol. 4, pp. 128-135, 2009.

[125] M. Chandrasekaran, K. Narayanan, and S. Upadhyaya, "Phishing email detection based on structural properties," Proceedings of the NYS Cyber Security Conference, pp. 1-7, 2006.
[126] H. Huang, L. Qian, and Y. Wang, "A SVM-based Technique to Detect Phishing URLs," Information Technology Journal, vol. 11, 2012.
[127] S. Mukkamala, G. Janoski, and A. Sung, "Intrusion detection: support vector machines and neural networks," Proceedings of the IEEE international joint conference on neural networks (ANNIE), pp. 1702-1707, 2002.
[128] I. Koprinska, J. Poon, J. Clark, and J. Chan, "Learning to classify e-mail," Information Sciences, vol. 177, pp. 2167-2187, 2007.
[129] Y. Tang, S. Krasser, Y. He, W. Yang, and D. Alperovitch, "Support vector machines and random forests modeling for spam senders behavior analysis," Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE, pp. 1-5, 2008.
[130] M. McCord and M. Chuah, "Spam detection on twitter using traditional classifiers," Autonomic and Trusted Computing, pp. 175-186, 2011.
[131] D. DeBarr, V. Ramanathan, and H. Wechsler, "Phishing detection using traffic behavior, spectral clustering, and random forests," Intelligence and Security Informatics (ISI), 2013 IEEE International Conference on, pp. 67-72, 2013.
[132] A. Aggarwal, A. Rajadesingan, and P. Kumaraguru, "PhishAri: Automatic realtime phishing detection on twitter," eCrime Researchers Summit (eCrime), 2012, pp. 1-12, 2012.
[133] S. Sarju and R. Thomas, "Spam Email Detection using Structural Features," International Journal of Computer Applications, vol. 89, 2014.
[134] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 38, pp. 649-659, 2008.
[135] R. M. Elbasiony, E. A. Sallam, T. E. Eltobely, and M. M. Fahmy, "A hybrid network intrusion detection framework based on random forests and weighted k-means," Ain Shams Engineering Journal, vol. 4, pp. 753-762, 2013.
[136] S. Revathi and A. Malathi, "Network Intrusion Detection Using Hybrid Simplified Swarm Optimization and Random Forest Algorithm on Nsl-Kdd Dataset," International Journal Of Engineering And Computer Science, vol. 3, pp. 3873-3876, 2014.
[137] J. Hovold, "Naive Bayes Spam Filtering Using Word-Position-Based Attributes," CEAS, 2005.
[138] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian approach to filtering junk e-mail," Learning for Text Categorization: Papers from the 1998 workshop, vol. 62, pp. 98-105, 1998.
[139] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with naive bayes-which naive bayes?," CEAS, pp. 27-28, 2006.
[140] I. Santos, C. Laorden, B. Sanz, and P. G. Bringas, "Enhanced topic-based vector space model for semantics-aware spam filtering," Expert Systems with applications, vol. 39, pp. 437-444, 2012.
[141] D. M. Farid, M. Z. Rahman, and C. M. Rahman, "Adaptive Intrusion Detection based on Boosting and Naïve Bayesian Classifier," International Journal of Computer Applications, vol. 24, pp. 12-19, 2011.

[142] N. B. Amor, S. Benferhat, and Z. Elouedi, "Naive bayesian networks in intrusion detection systems," Probabilistic Graphical Models for Classification, pp. 11-23, 2003.
[143] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Handling imbalanced datasets: A review," GESTS International Transactions on Computer Science and Engineering, vol. 30, pp. 25-36, 2006.
[144] G. L'Huillier, R. Weber, and N. Figueroa, "Online phishing classification using adversarial data mining and signaling games," Proceedings of the ACM SIGKDD Workshop on CyberSecurity and Intelligence Informatics, pp. 33-42, 2009.
[145] S. Sarafijanovic and J.-Y. Le Boudec, "Artificial immune system for collaborative spam filtering," Nature Inspired Cooperative Strategies for Optimization (NICSO 2007), pp. 39-51, 2008.
[146] S. J. Delany, P. Cunningham, A. Tsymbal, and L. Coyle, "A case-based technique for tracking concept drift in spam filtering," Knowledge-Based Systems, vol. 18, pp. 187-195, 2005.
[147] A. Di Pietro, L. While, and L. Barone, "Applying evolutionary algorithms to problems with noisy, time-consuming fitness functions," Evolutionary Computation, 2004. CEC2004. Congress on, vol. 2, pp. 1254-1261, 2004.
[148] T. A. Almeida, A. Yamakami, and J. Almeida, "Filtering spams using the minimum description length principle," Proceedings of the 2010 ACM Symposium on Applied Computing, pp. 1854-1858, 2010.
[149] Y. Sun, M. S. Kamel, A. K. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," Pattern Recognition, vol. 40, pp. 3358-3378, 2007.
[150] A. Voss and J. Voss, "A fast numerical algorithm for the estimation of diffusion model parameters," Journal of Mathematical Psychology, vol. 52, pp. 1-9, 2008.
[151] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," Biostatistics, vol. 9, pp. 432-441, 2008.
[152] M. A. Franco, N. Krasnogor, and J. Bacardit, "Speeding up the evaluation of evolutionary learning systems using GPGPUs," Proceedings of the 12th annual conference on Genetic and evolutionary computation, pp. 1039-1046, 2010.
[153] Z. Yang, X. Nie, W. Xu, and J. Guo, "An approach to spam detection by naive Bayes ensemble based on decision induction," Intelligent Systems Design and Applications, 2006. ISDA'06. Sixth International Conference on, vol. 2, pp. 861-866, 2006.
[154] A. K. Seewald, "An evaluation of naive Bayes variants in content-based learning for spam filtering," Intelligent Data Analysis, vol. 11, pp. 497-524, 2007.
[155] H. Zhang and D. Li, "Naive Bayes text classifier," Granular Computing, 2007. GRC 2007. IEEE International Conference on, pp. 708-708, 2007.
[156] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "Bayesian additive regression trees-based spam detection for enhanced email privacy," Availability, Reliability and Security, 2008. ARES 08. Third International Conference on, pp. 1044-1051, 2008.
[157] N. B. Amor, S. Benferhat, and Z. Elouedi, "Naive Bayes vs decision trees in intrusion detection systems," Proceedings of the 2004 ACM symposium on Applied computing, pp. 420-424, 2004.
[158] K.-C. Khor, C.-Y. Ting, and S.-P. Amnuaisuk, "A probabilistic approach for network intrusion detection," Proceedings of the Second Asia International Conference on Modelling and Simulation, pp. 463-468, 2008.

[159] C. Chui-Yu and Y.-T. Huang, "Integration of support vector machine with naïve Bayesian classifier for spam classification," Fuzzy Systems and Knowledge Discovery, 2007. FSKD 2007. Fourth International Conference on, vol. 1, pp. 618-622, 2007.
[160] D. S. Kim, H.-N. Nguyen, and J. S. Park, "Genetic algorithm to improve SVM based network intrusion detection system," Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference on, vol. 2, pp. 155-158, 2005.
[161] X. Xu and X. Wang, "An adaptive network intrusion detection method based on PCA and support vector machines," Advanced Data Mining and Applications, pp. 696-703, 2005.
[162] R. Chen, T. Chen, Y. Chien, and Y. Yang, "Novel questionnaire-responded transaction approach with SVM for credit card fraud detection," Advances in Neural Networks–ISNN 2005, pp. 916-921, 2005.
[163] R.-C. Chen, T.-S. Chen, and C.-C. Lin, "A new binary support vector system for increasing detection rate of credit card fraud," International Journal of Pattern Recognition and Artificial Intelligence, vol. 20, pp. 227-239, 2006.
[164] J. Zhang, G. Yang, L. Lu, M. Huang, and M. Che, "A novel visualization method for detecting DDoS network attacks," Visual Information Communication, pp. 185-194, 2010.
[165] T. Itoh, H. Takakura, A. Sawada, and K. Koyamada, "Hierarchical visualization of network intrusion detection data," Computer Graphics and Applications, IEEE, vol. 26, pp. 40-47, 2006.
[166] L. Wenyin, G. Huang, L. Xiaoyue, Z. Min, and X. Deng, "Detection of phishing webpages based on visual similarity," Special interest tracks and posters of the 14th international conference on World Wide Web, pp. 1060-1061, 2005.
[167] E. J. Lauria and G. K. Tayi, "Statistical machine learning for network intrusion detection: a data quality perspective," International Journal of Services Sciences, vol. 1, pp. 179-195, 2008.
[168] A. Y. Fu, L. Wenyin, and X. Deng, "Detecting phishing web pages with visual similarity assessment based on earth mover's distance (EMD)," Dependable and Secure Computing, IEEE Transactions on, vol. 3, pp. 301-311, 2006.
[169] C.-H. Tsang and S. Kwong, "Ant colony clustering and feature extraction for anomaly intrusion detection," Swarm Intelligence in Data Mining, pp. 101-123, 2006.
[170] A. A. Akinyelu and A. O. Adewumi, "Classification of Phishing Email using Random Forest Machine Learning Technique," Journal of Applied Mathematics vol. 2014, 2014.
[171] A. S. Foundation. "Apache Software Foundation. Spamassassin homepage". available at: http://spamassassin.apache.org/ (accessed 05 August 2014).
[172] K. Albrecht, N. Burri, and R. Wattenhofer, "Spamato-An Extendable Spam Filter System," in CEAS, 2005.
[173] M. Klassen and N. Paturi, "Web document classification by keywords using random forests," Networked Digital Technologies, pp. 256-261, 2010.
[174] T. M. Mitchell, "Machine learning. WCB," ed: McGraw-Hill Boston, MA:, 1997.
[175] S. Parameswaran and K. Q. Weinberger, "Large margin multi-task metric learning," Advances in neural information processing systems, pp. 1867-1875, 2010.
[176] C. Cortes and V. Vapnik, "Support-vector networks," Machine learning, vol. 20, pp. 273-297, 1995.

[177] C. R. Souza, "Kernel functions for machine learning applications," Creative Commons Attribution-Noncommercial-Share Alike, vol. 3, 2010.
[178] A. Ben-Hur and J. Weston, "A user’s guide to support vector machines," Data mining techniques for the life sciences, pp. 223-239, 2010.
[179] I. Fister, I. JrFister, X.-S. Yang, and J. Brest, "A comprehensive review of firefly algorithms," Swarm and Evolutionary Computation, vol. 13, pp. 34-46, 2013.
[180] C.-L. Huang and J.-F. Dun, "A distributed PSO–SVM hybrid system with feature selection and parameter optimization," Applied Soft Computing, vol. 8, pp. 1381-1391, 2008.
[181] C.-L. Huang and C.-J. Wang, "A GA-based feature selection and parameters optimizationfor support vector machines," Expert Systems with applications, vol. 31, pp. 231-240, 2006.
[182] R. Basnet, S. Mukkamala, and A. H. Sung, "Detection of phishing attacks: A machine learning approach," presented at the Soft Computing Applications in Industry, Berlin, Germany, 2008.
[183] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 2, pp. 1-27, 2011.
[184] M. Johnson. (2009), "SVM.NET". available at: http://www.matthewajohnson.org/software/svm.html (accessed 05 August 2014).
[185] F. Toolan and J. Carthy, "Phishing detection using classifier ensembles," eCrime Researchers Summit, 2009. eCRIME'09., pp. 1-9, 2009.
[186] J. Fehr, K. Z. Arreola, and H. Burkhardt, "Fast support vector machine classification of very large datasets," in Data Analysis, Machine Learning and Applications, ed: Springer, 2008, pp. 11-18.