A hybrid firefly and support vector machine classifier for phishing email detection

Oluyinka Aderemi Adewumi and Ayobami Andronicus Akinyelu
School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Durban, South Africa

Abstract

Purpose – Phishing is one of the major challenges faced by the world of e-commerce today. Thanks to phishing attacks, billions of dollars has been lost by many companies and individuals. The global impact of phishing attacks will continue to be on the increase and thus a more efficient phishing detection technique is required. The purpose of this paper is to investigate and report the use of a nature inspired based-machine learning (ML) approach in classification of phishing e-mails.

Design/methodology/approach – ML-based techniques have been shown to be efficient in detecting phishing attacks. In this paper, firefly algorithm (FFA) was integrated with support vector machine (SVM) with the primary aim of developing an improved phishing e-mail classifier (known as FFA_SVM), capable of accurately detecting new phishing patterns as they occur. From a data set consisting of 4,000 phishing and ham e-mails, a set of features, suitable for phishing e-mail detection, was extracted and used to construct the hybrid classifier.

Findings – The FFA_SVM was applied to a data set consisting of up to 4,000 phishing and ham e-mails. Simulation experiments were performed to evaluate and compared the performance of the classifier. The tests yielded a classification accuracy of 99.94 percent, false positive rate of 0.06 percent and false negative rate of 0.04 percent.

Originality/value – The hybrid algorithm has not been earlier apply, as in this work, to the classification and detection of phishing e-mail, to the best of the authors' knowledge.

Keywords Optimization techniques, Algorithms, Classification, Artificial intelligence, Intelligent agents

Paper type Research paper

1. Introduction

One of the major threats faced by vast majority of online users and businesses is fraud. There are three types of online fraud commonly perpetrated, namely: e-mail spam, phishing and network intrusion (Behdad *et al.*, 2012). Among these three, phishing has led to greater financial loss. 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. In 2013, phishing attacks rose from 279,580 in 2011 to 448,126 in 2013, leading to an estimated loss of about US$5.9 million (RSASecurity, 2014).

Also, Figure 1 (APWG, 2013), reveals that payment service industries and financial institutions are one of two major target industries in recent times. Phishing is generally perpetrated by sending deceitful well-composed e-mails to users. These e-mails contain links to cloned websites, and clicking on these links may re-direct users to a phishing or malware hosting website. Malwares hosting websites are infected with malicious codes that can gain access to private information of users and also cause damages to users' computers. Due to vast number of e-mails received by various users in recent times, e-mail filtering is a challenging task. Therefore, the need for a quicker, robust and effective filtering technique cannot be overemphasized. Several approaches have been proposed in literature including: network-based approach, blacklist, whitelist and

Kybernetes
Vol. 45 No. 6, 2016
pp. 977-994

0368-492X
DOI 10.1108/K-07-2014-0129

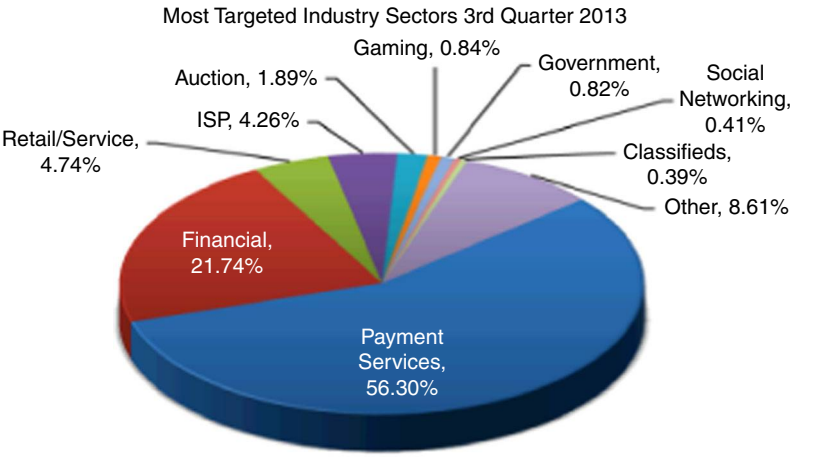

Figure 1.
Target industries by phishers

content-based approach. Network-based approach are costly to implement, difficult to maintain and time consuming (Johnson, 2003). Blacklist and whitelist approach yield high false positive (FP) and false negative (FN) rates because, they are incapable of detecting new phishing attacks as they emerge. According to APWG, the average uptime for phishing website is 44.39 hours (i.e. less than two days). Content-based approach aims to capture content and structural properties of data. According to White *et al.* (2012), blacklist approach is the widely used. Nevertheless, Bergholz *et al.* (2008) pointed out that content-based techniques is the most accurate and secured, because, they are capable of discovering new fraudulent patterns in large data sets as they evolve. Support vector machine (SVM) is arguably one of the best content-based technique. It analyses data, recognizes patterns in data and classifies the data into one of two given classes. Classification accuracy of SVM is based on the features and parameters used for training, hence, parameter selection and feature selection stages are very important.

In this paper, firefly algorithm (FFA) is used for parameter selection. The primary aim of this study is to design an improved phishing e-mail detection system, capable of accurately detecting known and emerging phishing patterns. The hybrid system is evaluated and it yielded excellent result.

2. Related work

Some existing phishing detection techniques are discussed in this section.

2.1 Phishing website detection

Prakash *et al.* (2010) designed a phishing detection system, consisting of two components: a Uniform Resource Locator (URL) prediction component and an approximate URL matching component. The URL prediction component was used to generate new URLs from different blacklisted URLs and also used to test (with the aid of DNS queries and content matching) whether the newly generated URLs are malicious or not. Furthermore, the URL matching component was used to compare the generated malicious URLs against a blacklist of malicious URLs. Authors evaluated the performance of PhishNet by using it to generate 18,000 URLs from 6,000 URLs provided by Jose Nazario (2009). Afterwards, authors tested the generated URLs against a database of blacklisted URLs collected from Phishtank (2016) and

SpamCutter (Anderson *et al.*, 2007). The test yielded a FP and FN rate of 3 and 5 percent, respectively. In another study, Medvet *et al.* (2008) proposed a phishing detection solution. In the study, first, authors captured the image of a suspected webpage and also captured the image of a legitimate webpage (previously visited by the user). Furthermore, authors compared both images and displays a message if the visual comparison of both images is similar. Authors evaluated the approach and it produced a FP rate of 17.4 percent and FN rate of 8.3 percent.

Chou *et al.* (2004) developed a web browser toolbar (in form of a plug-in) called SpoofGuard. SpoofGuard monitors various internet activities of users, and with the aid of some heuristics, it identifies phishing web pages by performing series of analysis and tests on webpage URLs. During the analysis, first, SpoofGuard examines the domain name of web pages opened by users and compare them to domain names of all websites recently visited by user. Furthermore, the toolbar examines the URL and webpage content to identify suspicious URLs, password fields, embedded links and images. Moreover, SpoofGuard analyzes each of the identified links using some heuristics described in Chou *et al.* (2004). Additionally, each of the detected images are compared to images of previously visited web pages stored on the web browser. Furthermore, SpoofGuard then computes the weighted sum of the results obtained from the previous tests, and displays a warning message to the user if the computed score is higher than a pre-defined threshold value. Authors evaluated the performance of SpoofGuard and it yielded a FP and FN rate of 38 and 9 percent, respectively.

2.2 Phishing e-mail detection

Yu *et al.* (2009) proposed a heuristic-based algorithm for phishing e-mail detection, called PhishCatch. PhishCatch is divided into four components. The first and second component is responsible for e-mail extraction and e-mail filtering, respectively. The third component is used to send alerts. The last component is used to store suspicious phishing e-mails. Authors tested the proposed algorithm on 2,000 phishing e-mails, and it produced a FP and FN rate of 1 and 20 percent, respectively. Similarly, Cook *et al.* (2008) developed a rule-based phishing e-mail filter called PhishWish. In the study, authors designed 11 rules for identifying and analyzing login URLs in e-mails, e-mail headers and images. PhishWish was tested on 1,000 e-mails and the test yielded a FP and FN rate of 8.3 and 2.5 percent, respectively. In another study, Zhang *et al.* (2007) proposed a phishing e-mail detection solution based on term frequency/inverse document frequency (TF-IDF) algorithm. In the study, authors used TF-IDF to examine webpage contents. Authors evaluated the performance of the technique using 3,038 unique URLs and it yielded a FP and FN rate of 3 and 11 percent, respectively.

In another work, Fette *et al.* (2007) proposed a random forest (RF)-based phishing e-mail detection technique. In the study, authors extracted ten features from a data set consisting of 860 phishing e-mails and 6,950 ham e-mails. Furthermore, the extracted features were used to build a classifier. Authors evaluated the technique and it yielded a FP and FN rate of 0.13 and 3.62 percent, respectively. In different studies, Toolan and Carthy (2009) and Bergholz *et al.* (2008) proposed machine learning (ML)-based approaches and achieved better results compared to Fette *et al.* (2007). Toolan and Carthy (2009) worked with five features and achieved an accuracy, FP rate and FN rate of 99.31, 1.4 and 0 percent, respectively, while Bergholz *et al.* (2008) worked with 81 features and achieved an accuracy, FP rate and FN rate of 99.85, 0.00 and 1.30 percent, respectively.

In a different study, Huang *et al.* (2012) proposed a SVM-based model for phishing URL detection. The model was designed using 23 URL-based features. Authors evaluated

the performance of the model, and it produced a classification accuracy of 99.0 percent. Zhang *et al.* (2011), introduced a framework for content-based phishing web page detection, based on Bayesian network. The framework consists of three components, namely: a text classifier (based on naïve Bayes rule), an image classifier (based on earth mover's distance) and a fusion algorithm (based on Bayes theory). The fusion algorithm is responsible for combining the classification results obtained from the image classifier and text classifier. Furthermore, the text classifier and image classifier, respectively is responsible for handling the text content and visual content of the webpage. The framework was evaluated and it yielded promising results. In addition, Al-Mo *et al.* (2011), proposed a clustering-based technique for phishing e-mail detection. The clustering-based technique was used to create rules for e-mail classifications. Authors evaluated the model and it yielded a classification accuracy, FP rate and FN rate of 99.7, 0.01 and 0.01 percent, respectively.

2.3 FFA-based SVM optimization

Shamshirband *et al.* (2015) proposed a hybrid technique for optimizing lens system design. In the study, FFA was used to optimize SVM parameters. Authors evaluated the technique and compared its result to support vector regression, artificial neural networks (ANN) and generic programming methods. FFA-SVM outperformed the three techniques. Similarly, in other studies, Sharma *et al.* (2013) and Chao and Horng (2015) used FFA to optimize SVM performance. In both studies, FFA was used for SVM parameter setting. Both techniques were evaluated and they produced good results. In addition, Ch *et al.* (2014) proposed a FFA-based technique for malaria prediction. In the study, authors designed a model using FFA and SVM. FFA was used for parameter optimization. The model was evaluated and its result outperformed three other algorithms, ANN, SVM and auto-regressive moving average.

2.4 Mobile-based phishing

Hou and Yang (2012) developed a web-based phishing attack defense mechanism for mobile devices (iOS platform). The mechanism logs key strokes of users and displays a warning message if the user is at the verge of entering sensitive information into a malicious website. Authors tested the mechanism on different mobile applications and different phishing attacks, and reported that the mechanism works very well for all the applications. In another study, Felt and Wagner (2011) performed a risk assessment of phishing activities on mobile devices. Authors evaluated 85 websites and 100 mobile devices and noted that websites and mobile applications interact in ways that exposes users to phishing attacks. Interaction between many mobile applications and websites requires some form social sharing or payment (Felt and Wagner, 2011). Either social sharing or payment requires users to enter a username and password. Unfortunately, users cannot verify validity of websites or mobile applications before entering their credentials, leaving them at risk. In a different work, Virvilis *et al.* (2014) evaluated the performance of different phishing detection mechanisms based on Android and iOS. In the study, authors worked with different browsers and compared their performance. Result revealed that some mobile browsers do not provide adequate protection for phishing, hence exposing users to phishing attacks.

2.5 Survey summary

Different phishing detection techniques has been proposed in literature. Based on all the reviewed studies, ML-based approach proposed by Fette *et al.* (2007), Toolan and

Carthy (2009), Al-Mo *et al.* (2011) and Bergholz *et al.* (2008), yielded the best performance. Furthermore, the review revealed that very few studies explored the use of FFA in combination with SVM to tackle phishing e-mail. This paper therefore introduces a technique for phishing e-mail detection based on FFA and SVM. In the proposed technique, FFA is used for parameter optimization and SVM is used for e-mail classification. Brief introduction to SVM and FFA is given in the next section.

3. Problem statement

A brief introduction to ML, SVM and FFA is presented in this section.

3.1 ML approach

Among ML practitioners, there is no generally acceptable definition for ML, it has been defined by several authors in different ways. ML was defined by Arthur Samuel as "the field of study that gives computers the ability to learn without been explicitly programmed" (Simon, 2013). Similarly, Mitchell (1997) defined ML as "A computer program is said to learn from experience E w.r.t some task T and some performance measure P, if its performance on T as measured by P, improves with experience E."

There are several types of ML classifiers, including: RF, neural networks, SVM, naïve bayes. SVM is arguably one the best classifiers. It is capable of yielding excellent classification result even when input data are not linearly separable and non-monotone (Auria and Moro, 2008).

3.2 SVMs

SVMs are arguably the most successful classification method in ML. It was invented by Vladimir N. Vapnik in 1995 (Cortes and Vapnik, 1995) and it is suitable for classification problems involving two classes. Utilizing a set of labeled 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 hyperplane with the widest possible margin. Afterwards, SVM use this feature space to classify new data (mapped into the space) based on which side of the hyperplane they fall into. An SVM with the largest margin (i.e. distance between the hyperplane and the closest data point) will yield a better result. Support vectors (as shown in Figure 2) are the closest points to the hyperplane, they are the critical points that give the maximum margin for all the N-points in the data.

Given a set of training data set, SVM finds the best (or optimal) hyper-plain with the maximum margin, which split the data set into two different classes (e.g. positive and negative). To split the data set into two different classes (with a minimal number of training errors), an optimal hyperplane is constructed by solving the optimization problem below (Cortes and Vapnik, 1995):

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

Subject to:

$$y_n\left(w^T \cdot x_n + b\right) \geqslant 1 - \epsilon_i, \quad n = 1, 2, \ldots, N$$

$$\epsilon_n \geqslant 0, \quad n = 1, 2, \ldots, N$$

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

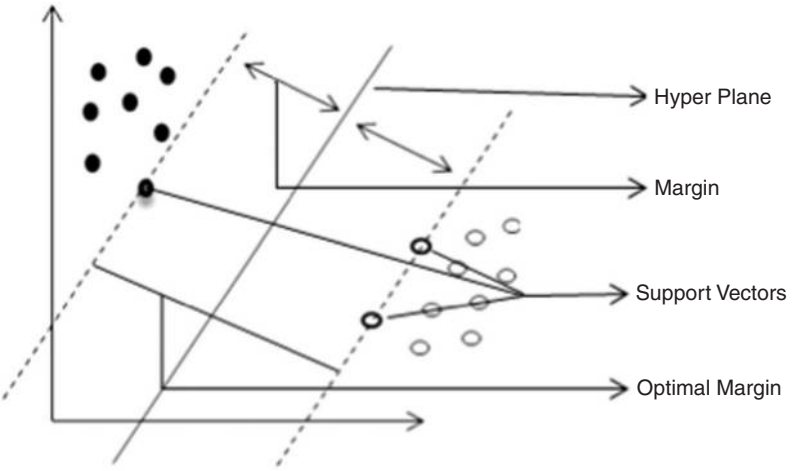

Figure 2.
Diagram showing support vectors, margin and hyperplane

where C is a constants (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 hyperplane. Function (1) describes an optimization problem that finds the best hyperplane (i.e. hyperplane with the optimal margin) which minimizes the number of training errors and also maximizes the margin with the correctly classified data points (Cortes and Vapnik, 1995). Hsu *et al.* (2003) pointed out that $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$ is called the kernel function. Generally, kernel functions provide simple bridge from linearity to non-linearity for various algorithms that can be expressed in terms of dot products (Souza, 2010).

There are four commonly used kernel functions (Hsu *et al.*, 2003):

(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: $K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right), \gamma > 0$.

(4) Sigmoid Function kernel: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$.

where γ, r and d are the kernel parameters.

In this study, RBF kernel was used, because, it has the ability to handle cases involving nonlinear relation between class labels and attribute (Hsu *et al.*, 2003). RBF kernel has two parameters: C (also called, the soft margin constant) and γ. These two parameters have a significant effect on the decision boundary of SVM and in turn have an effect on the classifier accuracy (Ben-Hur and Weston, 2010). The C and γ pair that will yield the best classification accuracy varies between problems, therefore, identifying the best C and γ pair for a given problem is an important aspect of SVM classification. The traditional method of achieving this is grid search using cross-validation. In grid search, a range of parameter values (i.e. C and γ) are chosen 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.

Trying exponentially growing sequence of C and γ is a good way of identifying good parameters; for example, $C = 2^{-5}, 2^{-3}, 2^{-1}, \ldots, 2^{-15}$; $\gamma = 2^{-15}, 2^{-13}, 2^{-11}, \ldots, 2^{3}$ (Hsu *et al.*, 2003). Optimization techniques such as, FFA, particle swarm intelligence and ant colony optimization (ACO) can also be used for parameter optimization. In this study, FFA is used. A brief introduction to FFA is presented in Section 3.3.

3.3 FFA

FFA (developed by Yang, 2008) 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 produces short flashes (by a process called bioluminescence) at regular intervals (Yang, 2008). These flashlights are produced to attract potential mating partners and preys, and also to send warning signals to predators. FFA is stochastic in nature and can be used to solve difficult real world optimization and NP-hard problems (Fister *et al.*, 2013). 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 (i.e. $I \propto 1/r^2$). Also, as the distance increases, light is absorbed into the atmosphere which in turn reduces the light intensity. Yang (2008) explained that flashlight can be formulated in such a way as that it will be associated (or proportional) to the value of the fitness function to be optimized. FFA has a number of variants, but this paper focusses on the original version developed by Yang (2008). This algorithm was formulated using four 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 toward fireflies with brighter intensity.

(3) The light intensity of fireflies is affected or determined by the landscape of the objective function to be optimized.

(4) In FFA, there are two important issues that need to be defined: 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 $I(y) \propto F(y)$. Light intensity varies with distance between fireflies, and with the degree of absorption by the atmosphere, given by the following equation:

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

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

Algorithm 1: Firefly Algorithm

1. Define initial values of firefly parameters: NF, NG, β_o, α, and γ
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 attractiveness variance with distance r using exp(−γr)
         5.1.1.4. Calculate new fitness values for all fireflies
         5.1.1.5. Update firefly light intensity Li
      5.1.2. **end for**
   5.2. **end for**
6. **end while**
7. **Output** optimized firefly light intensity

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

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:

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

where d represent the dimensionality of the problem at hand. The movement of firefly i to a more attractive firefly j is controlled by following equation:

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

$\alpha \in [0, 1]$, $\gamma \in [0, \infty]$. 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. The full algorithm is shown in Algorithm 1.

3.3.1 Objective function. The objective function for FFA_SVM is defined by formula (6). Classification accuracy is the major criteria used in designing the objective function for FFA_SVM. This implies that a firefly with high classification accuracy will yield a high fitness value. The objective function has one pre-defined weight, W_A, for the classification accuracy. The weight can be adjusted from 80 to 100 percent depending on users' preference. In this study, 95 percent is used for all the evaluations. Similar approach was used in Huang and Dun (2008) and Huang and Wang (2006):

$$fitness_i = W_A \times SVMAcc_i \tag{6}$$

where W_A = pre-defined weight for SVM accuracy and $SVMAcc_i$ = SVM accuracy for each firefly generation.

4. Proposed FFA_SVM

As earlier mentioned, in this study, FFA is integrated with SVM to construct a hybrid classifier. Specifically, FFA is used in place of grid algorithm, for parameter optimization. The flow chart and pseudocode for FFA_SVM is presented in Sections 4.1 and 4.2.

4.1 Flow chart of FFA_SVM

The flow chart of the proposed classification system is presented in Figure 3.

4.2 FFA_SVM algorithm

Algorithm 2: 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, β_o, α, and γ
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 & γ)
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. $1/10^{th}$ 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

4.3 Features used in the e-mail classification

In this paper, a group of 16 features that frequently appear in phishing e-mails was identified from different literature and used. Although, number of features used is few, a high accuracy is achieved. These features are described in this section.

4.3.1 URLs containing IP address. Website address (or URL) used by many legitimate organizations often contains names of the organization. For example, www.yahoo.com, tells us that the URL is owned by Yahoo. On the contrary, phishers use IP-based URLs (such as http://167.88.12.1/signin.ebay.com) to mask their identity from users. Therefore the presence of IP-based URLs in an e-mail is an indication that the e-mail is a potential phishing e-mail. This feature was used in (Bergholz *et al.*, 2008) and (Akinyelu and Adewumi, 2014).

4.3.2 Disparities between "href" attribute and LINK text. HTML anchor (i.e. < a >) tag is used to establish a link to another website. Linking to another website can be accomplished by defining a "href" attribute. This attribute describes the website location to be linked to. Links are usually rendered to a browser after a "Link text" has been clicked (e.g. < a href = "URL Address" > Link Text < /a >). The link text could be a plain text (e.g. 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 e-mail 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 Bergholz *et al.* (2008) and Akinyelu and Adewumi (2014).

4.3.3 Presence of "Link," "Click," "Here" in link text of a link. Link text in most phishing e-mails generally contain words like "Click," "Here," "Login," "Update." Hence, text of all links present in an e-mail is checked and a binary value is recorded based on the presence or absence of the words: Click, Here, Login, Update and Link. Similar feature was used in Gao *et al.* (2005) and Bergholz *et al.* (2008).

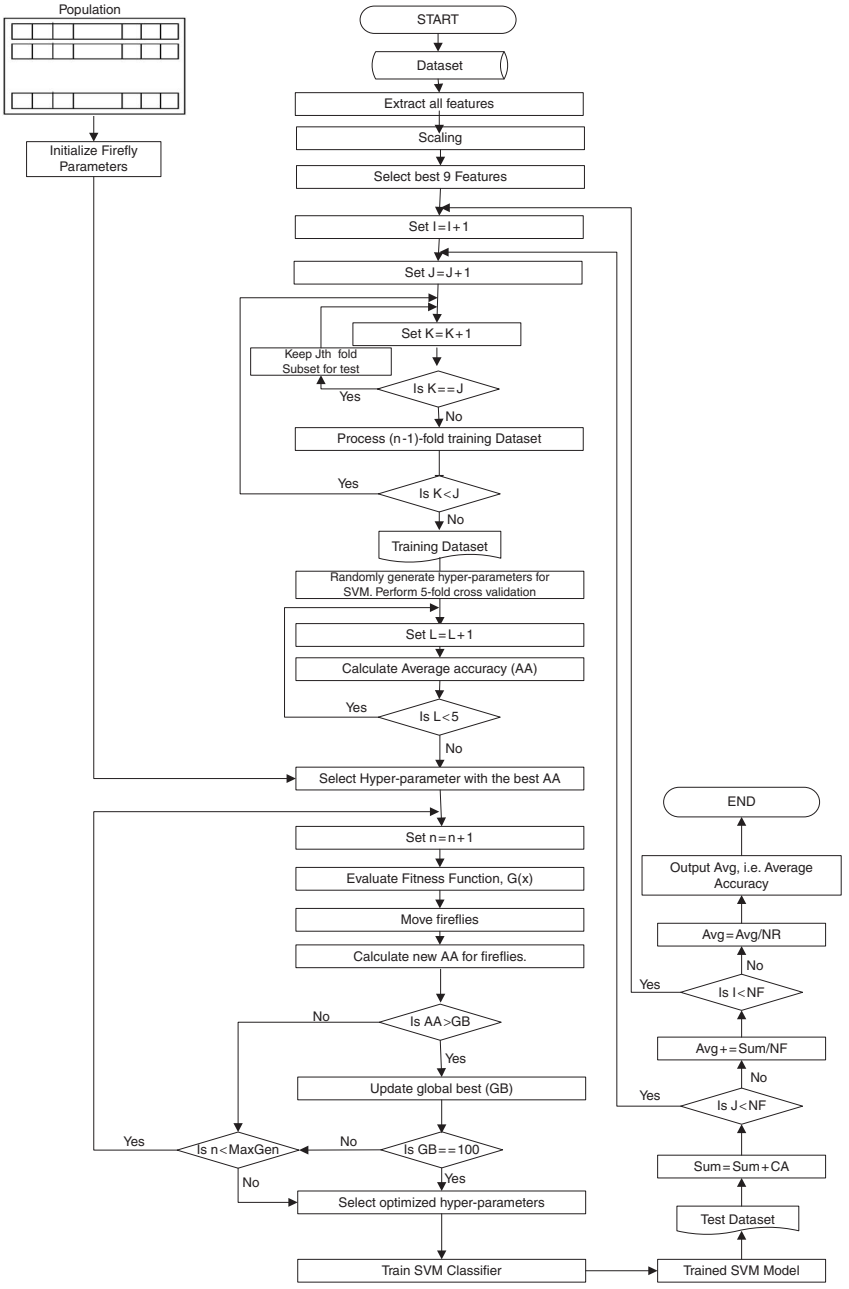

Figure 3.
Flow chart for FFA_SVM

4.3.4 Number of dots in domain name. Total number of dots that should be present in the domain name of a legitimate organization should not exceed three (Almomani *et al.*, 2012). For example, "www.yahoo.com" contains two dots. A binary value of 1 is recorded if an e-mail contains a URL whose domain name contains above three dots. This feature was used in Akinyelu and Adewumi (2014).

4.3.5 Html e-mail. E-mail format for each e-mail is defined by MIME standards. 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"). An e-mail is a potential phish e-mail if it contains a content-type with attribute "text/html" (Fette *et al.*, 2007). This is because, phishing attacks are launched using HTML links. This feature was used in Fette *et al.* (2007) and Akinyelu and Adewumi (2014).

4.3.6 Presence of javascript. Javascript can either be embedded in an e-mail (using the script (< script >) tag) or in a link (using the anchor (< a >) tag). Some phishers use Javascript to hide information from users. An e-mail is a potential phish e-mail if the "javascript" string is contained in either the body of an e-mail or in a link (Fette *et al.*, 2007). This feature was used in Fette *et al.* (2007) and Akinyelu and Adewumi (2014).

4.3.7 Number of links. In this study, total number of links embedded in an e-mail is extracted and used as a feature for classification. Phishing e-mails frequently contain multiple numbers of links to illegitimate websites (Zhang and Yuan, 2013). This feature was used in Zhang and Yuan (2013) and Akinyelu and Adewumi (2014).

4.3.8 Number of linked to domain. This feature refers to the total number of distinct domain names referenced by all the URLs in an e-mail. URLs contained in an e-mail are extracted, their domain names are extracted and counted, and resultant value is used a feature. Each domain name is counted once, duplicates are discarded. This feature is also used in Fette *et al.* (2007) and Akinyelu and Adewumi (2014).

4.3.9 From_Body_Matchdomain check. To extract this feature, all the domain names in an e-mail are extracted and each of these domain names is matched with the sender's domain (i.e. the domain name referred to by the "From" field of the same e-mail). If there is a disparity between any of the comparison, the e-mail is likely a phish e-mail (Almomani *et al.*, 2012). This feature was used in Akinyelu and Adewumi (2014) and Almomani *et al.* (2012).

4.3.10 Spam filter feature. SpamAssassin is a robust and effective open source spam filter already in use by some organizations. In this work, an off-line and untrained version of SpamAssassin was used to filter our data set and generate one binary feature. An e-mail is assigned a 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.* (2007) and Bergholz *et al.* (2008).

4.3.11 Word List Features. Some group of words that frequently appears in phishing e-mails were used as features. These words are grouped into six different groups and each of these groups is used as a single feature. For each group, presence of each word is counted and normalized. The groups of words include:

(1) Update, Confirm.

(2) User, Customer, Client.

(3) Suspend, Restrict, Hold.

(4) Verify, Account.

(5) Login, Username, Password.

(6) SSN, Social Security.

This feature is similar to the one proposed by Basnet *et al.* (2008).

5. Simulated experiments

The performance metric used in this study, the data set used and the results obtained are discussed in this section.

5.1 Evaluation method

ML involves two major phases: training phase and testing phase. Generally, prediction accuracy of classifiers depends on information gained during training. Information gain (IG) is one of the feature ranking metric prominently used in many text classification. In this study, IG (explained by Mitchell, 1997) is used for feature selection. The following evaluation metrics was used:

$$\text{FP Rate} = \frac{TP_n}{FP_n + TN_n} \tag{7}$$

$$\text{FN Rate} = \frac{FN_n}{TP_n + FN_n} \tag{8}$$

$$\text{Precision (Pr)} = \frac{TP_n}{TP_n + FP_n} \tag{9}$$

$$\text{Recall (R)} = \frac{TP_n}{TP_n + FN_n} \tag{10}$$

$$F\text{-Measure} = \frac{2 \times \text{Pr} \times R}{\text{Pr} + R} \tag{11}$$

where, TP_n, TN_n, FP_n, FN_n refers to the total number of true positives, true negatives, false positives and false negatives, respectively.

In this study, tenfold cross-validation was used and the cross-validation was repeated ten times. Doing this will correct the statistical dependency of all the individual instances in the data set (Bergholz *et al.*, 2008), and it will also lead to a good and accurate estimate of the evaluation. Table I shows a summary of the firefly parameters and the SVM parameters used in this study. The firefly parameters is similar to the parameters used by Yang (2008) and the (C, γ) pair is similar to the parameters suggested by Hsu *et al.* (2003).

SVM parameters (Hsu *et al.*, 2003)	C			γ	
	$[2^{-5}, 2^{15}]$			$[2^{-15}, 2^{3}]$	
Firefly parameters (Yang, 2008)	α	γ	β_0	N_f	N_g
	0.2	1	1	20	10

Notes: α, alpha; γ, gamma; β_0, beta; N_f, number of firefly; N_g, number of generations

Table I. SVM and firefly parameters

The computer used in running all the experiments is a 64-bit desktop, having a processor speed of 3.10GHz (Core i7) and a RAM size of 8.00 GB.

5.2 Data used

For the implementation and testing of FFA_SVM, a data set consisting of 3,500 ham e-mails (obtained from SpamAssassin; Group, 2006) and 500 phishing e-mails (obtained from Nazario, 2005) was used. Prior to classification, all the features described in Section 4.3 was programmatically extracted from the data set (using C#) and converted to vectors. Furthermore, the feature vectors were scaled (using Gaussian transform) and converted to the input format required by the SVM library used in this study (i.e. libSVM-library Chang and Lin, 2011). During extraction, all e-mails coming from the ham and phishing corpora is labeled appropriately. Afterwards, IG for all the 16 features was calculated and features with the best eight IG were selected and used to train FFA_SVM.

5.3 Result and discussion

Some experiments were performed to evaluate the performance of FFA_SVM. Each of the experiments were performed using ten times tenfold cross-validation (i.e. tenfold cross-validation is repeated ten times). Results from the experiments were compared to results of standard SVM. As shown in Table II, FFA_SVM yielded an average classification accuracy of 99.98 percent and standard SVM yielded an average classification accuracy of 99.96 percent. Also, FFA_SVM yielded a FP rate and FN rate of 0.01 and 0.08 percent, respectively, while standard SVM yielded a FP rate and FN rate of 0.03 and 0.16 percent, respectively. Additionally, FFA_SVM and standard SVM yielded a classification speed of 2,136.29 seconds and 2,268.31 seconds, respectively. These results revealed that the overall performance of FFA_SVM outperformed standard SVM, implying that FFA is a better parameter optimization technique compared to grid search algorithm. Additionally, as shown in Table III, FFA_SVM was compared to three best results discovered in literature and a classification accuracy of 99.98 percent was achieved, which is higher than the four classification accuracies (i.e. 99.49, 99.31, 99.71 and 99.85 percent) achieved by Fette *et al.* (2007), Toolan and Carthy (2009), Al-Mo *et al.* (2011) and Bergholz *et al.* (2008), respectively. Also, FFA_SVM performed better in terms of FP rate compared to Fette *et al.* and Toolan *et al.* To substantiate the result, a statistical analysis was carried out. FFA_SVM yielded a FP rate of 0.01 percent and a variance of 0.002399 using ten times tenfold cross-validation. This leads to a z-statistics of 17.3245. If it is assumed that the result of both Toolan *et al.* and Fette *et al.* comes from a distribution with similar variance, then it can be concluded with 99 percent level of certainty that result produced by FFA_SVM is statistically significant (Figure 4).

Table II. Tenfold cross-validation result for FFA_SVM

Method	Total e-mail	P:H ratio	APA (%)	FP (%)	FN (%)	R (%)	Pr (%)	F-M (%)	Time (seconds)
FFA_SVM	4,000	1:9	*99.98*	*0.01*	*0.08*	*99.92*	*99.94*	*99.93*	*2,136.29*
Standard SVM	4,000	1:9	99.96	0.03	0.16	99.84	99.80	99.82	2,268.31

Notes: APA, average prediction accuracy; SR, success rate; FP, false positive; FN, false negative; R, recall; Pr, precision; T, time; F-M, F-Measure; P:H, Phish:Ham; GB, global best

Technique	Average accuracy (%)	Method	FP rate (%)	FN rate (%)	Precision (%)	Recall (%)	F-measure (%)
Fette *et al.* (2007)	99.49	Random forest	0.13	3.62	98.92	96.38	97.64
R-Boost (Toolan and Carthy, 2009)	99.31	C5.0	1.4	0	98.63	100	99.31
PECM (Al-Mo *et al.*, 2011)	99.70	Clustering	0.01	0.01	–	–	–
Bergholz *et al.* (2008)	99.85	SVM	0.01	1.30	99.88	98.70	99.29
FFA_SVM	99.98	FFA +SVM	0.01	0.08	99.94	99.92	99.93

Table III.
Comparative results of FFA_SVM with literature

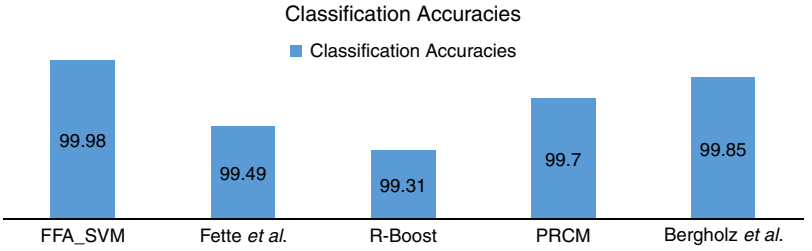

Figure 4.
Average accuracy of FFA_SVM vs four best results in literature

6. Conclusion

Phishing remains a threat to online users worldwide. It is a lucrative trade that will keep advancing as technology evolves, hence a robust and dynamic phishing e-mail detection technique is required. Much work has been done by the research community to significantly reduce phishing attacks with great accuracy. This paper proposes an efficient phishing e-mail filtering approach capable of handling both existing and emerging phishing attacks. In this approach, FFA is combined with SVM classifier. The hybrid classifier was evaluated on a data set consisting of 4,000 e-mails, and it produced an overall classification accuracy of 99.98 percent, FP and FN rate of 0.01 and 0.08 percent, respectively. This reveals that FFA is a reliable optimization technique that can be combined with SVM to build a robust classifier. In the future, we plan to introduce new features and also explore other nature inspired techniques, which will hopefully yield better results.

References

Akinyelu, A.A. and Adewumi, A.O. (2014), "Classification of phishing email using random forest machine learning technique", *Journal of Applied Mathematics*, Vol. 2014, 6pp.

Al-Mo, A.A.D., Wan, T.-C., Al-Saedi, K., Altaher, A., Ramadass, S., Manasrah, A., Melhiml, L.B. and Anbar, M. (2011), "An online model on evolving phishing e-mail detection and classification method", *Journal of Applied Sciences*, Vol. 11 No. 18, pp. 3301-3307.

Almomani, A., Wan, T.-C., Altaher, A., Manasrah, A., Almomani, E., Anbar, M., Alomari, E. and Ramadass, S. (2012), "Evolving fuzzy neural network for phishing emails detection", *Journal of Computer Science*, Vol. 8 No. 7, pp. 1099-1107.

Anderson, D.S., Fleizach, C., Savage, S. and Voelker, G.M. (2007), "Spamscatter: characterizing internet scam hosting infrastructure", *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, *Boston, MA*, pp. 1-14.

APWG (2013), "Phishing activity trends report", available at: http://docs.apwg.org/reports/apwg_trends_report_q3_2013.pdf (accessed April 5, 2016).

Auria, L. and Moro, R.A. (2008), "Support vector machines (SVM) as a technique for solvency analysis", available at: http://dx.doi.org/10.2139/ssrn.1424949 (accessed April 23, 2014).

Basnet, R., Mukkamala, S. and Sung, A.H. (2008), "Detection of phishing attacks: a machine learning approach", in Prasad, B. (Ed.), *Soft Computing Applications in Industry*, Springer, Berlin, pp. 373-383.

Behdad, M., Barone, L., Bennamoun, M. and French, T. (2012), "Nature-inspired techniques in the context of fraud detection", *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 42 No. 6, pp. 1273-1290.

Ben-Hur, A. and Weston, J. (2010), "A user's guide to support vector machines", in Carugo, O. and Eisenhaber, F. (Eds), *Data Mining Techniques for the Life Sciences*, Humana Press, Totowa, NJ, pp. 223-239.

Bergholz, A., Chang, J.H., Paaß, G., Reichartz, F. and Strobel, S. (2008), "Improved phishing detection using model-based features", *Proceedings of the Conference on Email and Anti-Spam (CEAS), Vol. 2008*, 10 pp.

Ch, S., Sohani, S.K., Kumar, D., Malik, A., Chahar, B.R., Nema, A.K., Panigrahi, B.K. and Dhiman, R.C. (2014), "A support vector machine-firefly algorithm based forecasting model to determine malaria transmission", *Neurocomputing*, Vol. 129, pp. 279-288.

Chang, C.-C. and Lin, C.-J. (2011), "LIBSVM: a library for support vector machines", *ACM Transactions on Intelligent Systems and Technology (TIST)*, Vol. 2 No. 3, pp. 1-27.

Chao, C.-F. and Horng, M.-H. (2015), "The construction of support vector machine classifier using the firefly algorithm", *Computational Intelligence and Neuroscience*, Vol. 2015, 2pp.

Chou, N., Ledesma, R., Teraguchi, Y. and Mitchell, J.C. (2004), "Client-side defense against web-based identity theft", *Proceedings of the 11th Annual Network and Distributed System Security Symposium (NDSS)*, *San Diego, CA*, pp. 1-16.

Cook, D.L., Gurbani, V.K. and Daniluk, M. (2008), "Phishwish: a stateless phishing filter using minimal rules", in Tsudik, G. (Ed.), *Financial Cryptography and Data Security*, Springer, Berlin and Heidelberg, pp. 182-186.

Cortes, C. and Vapnik, V. (1995), "Support-vector networks", *Machine Learning*, Vol. 20 No. 3, pp. 273-297.

Felt, A.P. and Wagner, D. (2011), "Phishing on mobile devices", IEEE Workshop on Web 2.0 Security and Privacy, Oakland, CA.

Fette, I., Sadeh, N. and Tomasic, A. (2007), "Learning to detect phishing emails", *Proceedings of the 16th International Conference on World Wide Web, ACM*, pp. 649-656.

Fister, I., Jrfister, I., Yang, X.-S. and Brest, J. (2013), "A comprehensive review of firefly algorithms", *Swarm and Evolutionary Computation*, Vol. 13 No. 1, pp. 34-46.

Gao, H.-H., Yang, H.-H. and Wang, X.-Y. (2005), "Ant colony optimization based network intrusion feature selection and detection", *IEEE Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, pp. 3871-3875.

Group, C. (2006), "SpamAssassin Data", CSMINING GROUP, available at: www.csmining.org/index.php/spam-assassin-datasets.html (accessed April 23, 2016).

Hou, J. and Yang, Q. (2012), "Defense against mobile phishing attack", Computer Security Course Project, University of California, Berkeley, CA, available at: www.personal.umich.edu/yangqi/pivot/mobilephishingdefense.pdf (accessed April 23, 2016).

Hsu, C.-W., Chang, C.-C. and Lin, C.-J. (2003), *A Practical Guide to Support Vector Classification*, Department of Computer Science, National Taiwan University, Taipei, Taiwan.

Huang, C.-L. and Dun, J.-F. (2008), "A distributed PSO-SVM hybrid system with feature selection and parameter optimization", *Applied Soft Computing*, Vol. 8 No. 4, pp. 1381-1391.

Huang, C.-L. and Wang, C.-J. (2006), "A GA-based feature selection and parameters optimizationfor support vector machines", *Expert Systems with Applications*, Vol. 31 No. 2, pp. 231-240.

Huang, H., Qian, L. and Wang, Y. (2012), "A SVM-based technique to detect phishing URLs", *Information Technology Journal*, Vol. 11 No. 7, pp. 921-925.

Johnson, B.C. (2003), "Intrusions and their detection: addressing common hacker exploits", available at: http://systemexperts.com/media/pdf/hackerid.pdf (accessed April 21, 2016).

Medvet, E., Kirda, E. and Kruegel, C. (2008), "Visual-similarity-based phishing detection", *Proceedings of the 4th International Conference on Security and Privacy in Communication Networks, ACM, Istanbul*, pp. 1-6.

Mitchell, T.M. (1997), *Machine Learning*, WCB/McGraw-Hill, Boston, MA.

Nazario, J. (2005), "Phishing corpus", available at: http://monkey.org/jose/wiki/doku.php?id=PhishingCorpus (accessed April 23, 2016).

Nazario, J. (2009), "Phishing corpus", available at: www.monkey.org/jose/wiki/doku.php?id=phishingcorpus (accessed April 23, 2016).

Phishtank (2016), "Join the fight against phishing", available at: www.phishtank.com (accessed April 23, 2016).

Prakash, P., Kumar, M., Kompella, R.R. and Gupta, M. (2010), "Phishnet: predictive blacklisting to detect phishing attacks", *INFOCOM 2010: Proceedings of the 29th Conference on Information communications, IEEE, San Diego, CA*, pp. 346-350.

RSASecurity (2014), "2013: a year in review", available at: www.emc.com/collateral/fraud-report/rsa-online-fraud-report-012014.pdf (accessed April 21, 2016).

Shamshirband, S., Petković, D., Pavlović, N.T., Ch, S., Altameem, T.A. and Gani, A. (2015), "Support vector machine firefly algorithm based optimization of lens system", *Applied Optics*, Vol. 54 No. 1, pp. 37-45.

Sharma, A., Zaidi, A., Singh, R., Jain, S. and Sahoo, A. (2013), "Optimization of SVM classifier using firefly algorithm", *IEEE Second International Conference on Image Information Processing (ICIIP), Shimla, December 9-11*, pp. 198-202.

Simon, P. (2013), *Too Big to Ignore: The Business Case for Big Data*, ISBN 978-1-118-63817-0, Wiley.

Souza, C.R. (2010), "Kernel functions for machine learning applications", Creative Commons Attribution-Noncommercial-Share Alike, available at: http://crsouza.com/2010/03/kernel-functions-for-machine-learning-applications/ (accessed April 23, 2016).

Toolan, F. and Carthy, J. (2009), "Phishing detection using classifier ensembles", *eCrime Researchers Summit, IEEE, Tacoma, WA*, pp. 1-9.

Virvilis, N., Tsalis, N., Mylonas, A. and Gritzalis, D. (2014), "Mobile devices – a phisher's paradise", in Obaidat, M.S., Holzinger, A. and Samarati, P. (Eds), *11th International Conference on Security and Cryptography (SECRYPT)*, ScitePress, pp. 79-87.

White, J.S., Matthews, J.N. and Stacy, J.L. (2012), "A method for the automated detection phishing websites through both site characteristics and image analysis", in Ternovskiy, I.V. and Chin, P. (Eds), *SPIE Defense, Security, and Sensing*, International Society for Optics and Photonics, Baltimore, MD, pp. 84080B-84080B-11.

Yang, X.-S. (2008), *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, Beckington.

Yu, W.D., Nargundkar, S. and Tiruthani, N. (2009), "Phishcatch – a phishing detection tool", *33rd Annual IEEE International Computer Software and Applications Conference, COMPSAC, Vol. 2*, pp. 451-456.

Zhang, H., Liu, G., Chow, T.W. and Liu, W. (2011), "Textual and visual content-based anti-phishing: a Bayesian approach", *IEEE Transactions on Neural Networks*, Vol. 22 No. 10, pp. 1532-1546.

Zhang, N. and Yuan, Y. (2013), "Phishing detection using neural network", Department of Computer Science, Department of Statistics, Stanford University, CA, available at: http://cs229.stanford.edu/proj2012/ZhangYuan-PhishingDetectionUsingNeuralNetwork.pdf (accessed April 23, 2016).

Zhang, Y., Hong, J.I. and Cranor, L.F. (2007), "Cantina: a content-based approach to detecting phishing web sites", *Proceedings of the 16th International Conference on World Wide Web, ACM, Banff*, pp. 639-648.

Corresponding author
Oluyinka Aderemi Adewumi can be contacted at: laremtj@gmail.com