A New Binary Particle Swarm Optimization for Feature Subset Selection with Support Vector Machine

Amir Rajabi Behjat[1], Aida Mustapha[1], Hossein Nezamabadi-Pour[2], Md. Nasir Sulaiman[1], and Norwati Mustapha[1]

[1] Faculty of Computer Science and Information Technology
University Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia
rajabi.amir6@gmail.com,
{aida_m,nasir,norwati}@upm.edu.my
[2] Department of Electrical Engineering, Shahid Bahonar University of Kerman
P.O. Box 76169-133, Kerman, Iran
nezam@mail.uk.ac.ir

Abstract. Social Engineering (SE) has emerged as one of the most familiar problem concerning organizational security and computer users. At present, the performance deterioration of phishing and spam detection systems are attributed to high feature dimensionality as well as the computational cost during feature selection. This consequently reduces the classification accuracy or detection rate and increases the False Positive Rate (FPR). This research is set to introduce a novel feature selection method called the New Binary Particle Swarm Optimization (NBPSO) to choose a set of optimal features in spam and phishing emails. The proposed feature selection method was tested in a classification experiments using the Support Vector Machine (SVM) to classify emails according to the various features as input. The results obtained by experimenting on two phishing and spam emails showed a reasonable performance to the phishing detection system.

Keywords: Particle swarm optimization, feature selection, phishing, spam, social engineering, SVM.

1 Introduction

Many IDSs are using database of well-known actions to compare normal and abnormal data or activities for sending alerts when a match is detected [1], [2]. Attackers evade Intrusion Detection Systems (IDS) using various ways, such as using the old unknown attack, hiding an attack in a concealed or encrypted channel, as well as posing as social engineering attacks [3]. Social engineering (SE) is a developing science that capitalizes on human trusty nature and is a serious threat to all organizations [4], [5]. Most familiar social engineering attacks include phishing and spam emails that convince users to open emails with abnormal links, pictures, videos, and even URLs [3], [6]. Phishing, spam, and

T. Herawan et al. (eds.), *Recent Advances on Soft Computing and Data Mining SCDM 2014*, Advances in Intelligent Systems and Computing 287,
DOI: 10.1007/978-3-319-07692-8_5,

even legitimate emails are basically similar in the style and content. Nonetheless, beyond the content, the structural and other special features will be able to make a distinction whether the emails are phishing emails or spam emails. Phishing is considered as a subcategory of spam [7].

In detecting phishing and spam/legitimate emails, the feature selection quality along with computational methods are required to guarantee the effectiveness of a classification/detection system [3]. This means the elimination of irrelevant features via the feature reduction process will increase the accuracy and reduce the false positive rate during detection since a smaller number of feature sets up the speed of the computation [8], [9]. Most studies have considered various features of phishing and spam emails [1], [10], [11], [12], [13]. The accuracy of phishing and spam detection showed a good result in a number of studies [12], [14], [15] while the number of features multiplies the computational cost and decreases the accuracy [16]. Generally, the lack of knowledge related to the false positive and the impact of features on the accuracy will reduce the performance of phishing detection [14].

The main objective of this study is to select a combination of features in phishing emails and evaluate the impact of these features on the basis of computational cost, false positive rate, and accuracy percentage in detecting phishing emails. This study will propose a New Binary Particle Swarm Optimization (NBPSO) for feature selection and will test the performance via a classification experiment with an existing Support Vector Machine (SVM) classifier. This study attempts to prove that the high classification accuracy and low false positive rate are possible through feature reduction that should result in lower dimensionality in feature sets, which covers important parts of emails such as subjects, bodies, links, URLs and attached files within the email body.

The paper organization keeps on as follows. Section 2 begins with the principle of Support Vector Machine (SVM). Section 3 introduces the principles of Particle Swarm Optimization (PSO) and Binary PSO preparation. Section 4 details out the experimental results, Section 5 discusses the ROC curve and AUC analysis, and finally Section 6 concludes the work and sets future research.

2 Principles of Support Vector Machine (SVM)

In 1995, Guyon, Boser, and Vapnik [17] introduced the Support Vector Machine (SVM). SVM is based on statistical learning theory and is able to prevents overfitting in classification, hence is well-known for its high classification accuracy. The SVM classifier predicts a new instance into a predefined category based on given training examples as shown in Equation 1:

$$D = (o_i, y_i) | o_i \in R_p, y_i \in \{-1, 1\}_{i=1}^{pt} \tag{1}$$

where pt is the number of samples and (o_i, y_i) shows the i^{th} training sample with its corresponding labels. $o_i = (o_i, o_{i_2}, o_{i_3}, \ldots, o_{i_p})$ is a p-dimensional vector in the feature space as shown in Equation 2.

$$\min 1/2\langle w \cdot w\rangle + C\sum_{i=1}^{pt}\zeta_i, y_i(\langle w \cdot o_i\rangle + b)\zeta_i \geq 0 + \zeta_i - 1 \geq 0 \tag{2}$$

where C is the penalty parameter that controls the decision function complexity and the number of misclassified training examples. ζ_i is the positive slack variable. The hyperplane which has the largest distance to the nearest training data point will create a suitable separation. This model can be solved using the introduction of the Langrage multipliers $0 \leq \alpha_i \leq C$ for dual optimization model [18], [19], [20]. The classifier function and the optimal b^* and w^* can be defined after achievement of the optimal solution α_i based on Equation 3.

$$\text{sign}\langle w^* \cdot o_i + b^*\rangle \text{ or } \text{sign}(\sum_{i=1}^{pt} y\alpha_i^*\langle o_i \cdot o\rangle + b^*)oi.o + b*) \tag{3}$$

The SVM maps training data nonlinearly within a high-dimensional feature space by kernel function $k(o_i, o_j)$ where linear separation may be possible. The kernels will decrease a complex classification task by separating hyperplanes. The typical kernel function given as in Equation 4.

$$k(o_i, o_j) = \exp(\frac{-1}{\delta^2}\|o_i - o_j\|^2) = \exp(-\gamma\|o_i - o_j\|^2) \tag{4}$$

The SVM classifier then changes to the following model after choosing the kernel function shown in Equation 5.

$$\text{sign}(\sum_{i=1}^{pt}\gamma_i\alpha_i^*\langle o_i \cdot o\rangle + b^*) \tag{5}$$

The performance of an SVM classifier is highly dependent on C and γ, which are the hyper-parameters. These two parameters will affect the number of support vectors and the size of margin in the SVM [20].

3 Principles of Particle Swarm Optimization (PSO) and Binary PSO Preparation

Particle Swarm Optimization (PSO) was introduced in 1995 based on the behavior of swarming animals. This algorithm has been used for optimization in different fields such as data clustering, optimization of artificial neural network, and network wireless [15], [16]. The set of particles builds a population (swarm) of candidate solutions. PSO is similar to heuristic algorithms in the sense that it searches some solutions within the initialized population. However, unlike Genetic Algorithm (GA), PSO does not follow operators such as mutation and crossover [7], [16].

In PSO algorithm, each particle is a point in D-dimensional space, so the i^{th} particle is represented as $X_i = (x_{i_1}, x_{i_2}, \ldots, x_{i_s})$. Because PSO calculates the best fitness rate (*pbest*) according to previous position of each particle, the rate

for any particle is $P_i = (p_{i_1}, p_{i_2}, \ldots, p_{i_s})$. The global best and velocity of particle i are '*gbest*' and $V_i = (v_{i_1}, v_{i_2}, \ldots, v_{i_s})$, respectively. Meanwhile, the manipulation of each particle is continued as the following Equation 6 and Equation 7.

$$v_{id} = w * v_{id} + c1 * rand() * (p_{ad} - x_{id}) + c2 * Rand() * (p_{ad} - x_{id}) \quad (6)$$

$$x_{id} = x_{id} + v_{id} \quad (7)$$

where w is the inertia weight, c_1 and c_2 are the stochastic acceleration weighting that leads particles toward *pbest* and *gbest* positions. $rand()$ and $Rand()$ are the random functions between [0,1]. $Vmax$ shows the velocity of each particle.

The New Binary Particle Swarm Optimization (NBPSO) algorithm follows the action of chromosomes in GA, so it is coded such as a binary string. In the specific dimension, the particle velocity is used like a probability distribution with the main role to randomly produce the particle position. Updating the particle position follows Equation 8, whereby the sigmoid function is used to identify new particle position based on binary values.

$$S(v_{id}) = Sigmoid(v_{id}) = \frac{1}{1 + e^{-v_{id}}} \quad (8)$$

$$\text{If } rand < S(v_{id}(t+1)) \text{ then } x_{id}(t+1) = 1 \text{ else } x_{id}(t+1) = 0 \quad (9)$$

where $rand$ is a random value between [0, 1] and v_{id} is limited to $vmax()$. In each dimension, a bit value 1 shows the selected feature may participate for the next generation. On the other hand, a bit value of 0 is not required as a relevant for next generation [8].

3.1 Drawbacks of Current PSO

The particles in continuous Particle Swarm Optimization (PSO) algorithm are defined by x and v values. The particle at position (x) is a potential solution and v is the speed of each particle that shows the future position of a particular particle relative to its current position. Large value of v shows that the particle position is not suitable, hence the value should change towards an optimal solution. The small value of v demonstrates that the particle position is moving towards optimal solution or with 0 value.

There are different definitions of x and v in binary particle swarm optimization algorithm (BPSO). The speed of particle (v) in this algorithm shows 0 or 1 for the position of particle (x) instead of finding optimal solutions. In other word, the v_{id} identifies the x_{id} value (0 or 1). Since the probability of x_{id} should be between 0 and 1, then v_{id} uses the sigmoid function as previously shown in Equation 8.

In BPSO, the large value of x_{id} (towards positive values) meaning x_{id} is near to 1 and the small value (towards negative values) reduces the probability of

1 for x_{id}. On the other hand, if v_{id} is 0, then the value of x_{id} will be changed to 0 or 1 with the probability of 50%. In addition, the value of x_{id} is identified regardless the previous value or position. Based on these scenario, there are two drawbacks in the BPSO as algorithm deliberated as follows.

The first drawback lies in the sigmoid function. Conceptually, the large value of v_{id} towards negative or positive values shows that x_{id} position should change for a specific dimension. However, in the binary particle swarm optimization, v_{id} steers x_{id} towards 0 or 1. Additionally, the speed of particle (v) near to 0 shows that the position of particle (x) is satisfied and the sigmoid function demonstrates an equal probability of 0 or 1 for x_{id}.

The second drawback is the process to update particle position (x). In the average of initial iterations, all the particles come up the optimal solution. Nonetheless, these particles keep out the optimal solution even after several iterations. This means the optimal solution may be near to 0, but the probability of 0 or 1 decrease to 50% during such times.

3.2 Proposed New Binary Particle Swarm Optimization

Both drawbacks in the Binary Particle Swarm Optimization (BPSO) algorithm may be resolved using suitable functions and by updating the particle position (x_{id}) as shown in Equation 11. In this algorithm, the sigmoid function is replaced to $S'(v_{id})$ as shown in Equation 10.

$S'(v_{id})$ proves that the value of v_{id} towards positive values is the same as the negative values. Whenever the speed of particle (v_{id}) is near to 0 value, the output of function increases and moves to 0 too. On the other hand, for updating particle position in Equation 6 is replaced to the one in Equation 3. Finally, the large v_{id} value demonstrates that the particle position is not suitable and changes towards 0 or 1 while the small value of v_{id} decreases the probability of the changes in the position of particle ($x_i d$). On the other hand, the 0 value of v_{id} will fix the particle position.

$$S'(v_{id}) = |tanh(\alpha x)| \tag{10}$$

$$\begin{aligned} \text{If } rand < s'(v_{id}(t+1)) \text{ then } x_{id}(t+1) &= complement(x_{id}) \\ \text{else } x_{id}(t+1) &= x_{id}(t) \end{aligned} \tag{11}$$

In this study, NBPSO finds an optimal binary vector, where each bit is associated with a feature. If the i^{th} bit of this vector equals to 1, the i^{th} feature will be allowed to participate in the classification. If the bit is a zero (0), the feature cannot participate in the classification process. Each resulting subset of features will be evaluated according to its classification accuracy on a set of testing data in an SVM classifier. We will divide the entire features by their importance and eliminate irrelevant features, which is indicated by the lowest ranked during the process. In other words, we will select important features by using the variable of the importance value that is based on their repetition in two classes. This strategy

enables our approach to reduce the computational expenses of the dataset as well as to enhance the detection rates and reduce the feature dimensionality.

4 Experiments and Results

This study evaluated the classification accuracy or detection rate of New Binary Particle Swarm Optimization (NBPSO) algorithm for feature selection. The classification experiment used the Support Vector Machine (SVM) trained with the measurement vectors of 14,580 spam, ham, and phishing emails. A total of 1,620 measurements were available for testing. The experiments were performed using the Intel Pentium IV processor with 2.7GHz CPU, 4GB RAM, and Windows 7 Operating System with MATHWORK_R2010b development environment. The classification experiments used three well-known datasets in shown in Table 1.

Table 1. Dataset and the number of class

No.	Dataset	Size
1	SpamAssassin	6,954
2	SpamEmail	1,895
3	PhishingCorpus	4,563

In order to select a set of combined features in within the pool of phishing email, we applied the NBPSO to choose the best features within the extracted features as reported in the previous studies [1], [10], [11], [12], [13]. The results showed that NBPSO was able to search the complex space with a big number of features. Furthermore, NBPSO found an optimal binary vector, where each bit was associated with a feature. If the i^{th} bit of this vector equals to 1, the i^{th} feature will be allowed to participate in the classification process; but if the bit equals to 0 (zero), the feature cannot participate in the classification process. Each resulting subset of features was evaluated according to its classification accuracy on a set of testing data using the SVM classifier.

In order to evaluate the selected features based NBPSO as the feature selection method, we divided the features in each category and combined the categories in four classifiers such as 2C, 3C, 4C and 5C, so category 2, 3, 4, and 5 were divided into each classifier respectively. This analysis identified the best combination and the created detection rate by them. The best classifiers based on different categories with selected relevant features are related to 3C = C1, C4, and C5, 4C = C1, C2, C4, and C5, 5C = C1, C2, C3, C4, and C5 and 2C = C1 and C5 classifiers with respective detection rate as shown in Table 2. The best false positive rate (FPR) achieved was for 4C with 0.1%.

While previous studies either reported the FPR or complement the results with accuracy rate, we believe that we could improve these two rates near to 100 accuracy and and 0 for FPR, respectively. On the other hand, the number of

Table 2. The detection rate for each combination of features

Features	Feature Combinations	Detection Rate (%)
2C	C1,C5	91.49
3C	C1,C4,C5	98.99
4C	C1, C2,C4,C5	97.77
5C	C1,C2,C3,C4,C5	94.22

Table 3. CPU time and elapsed time to select the best combination of features

Features	Feature Combinations	Elapsed Time (s)	CPU Time (%)
2C	C1,C5	152.23	28
3C	C1,C4,C5	173.56	37.2
4C	C1, C2,C4,C5	198.38	46.45
5C	C1,C2,C3,C4,C5	215.67	63.65

features in each category influenced the performance and computational cost of the classifier. For example, 5C with high number of features has a higher CPU time and computational cost, which is 63% and elapsed time to 215.67(s) during the feature selection process as shown in Table 3.

For each dataset, the experiments were repeated in 10 runs based on 4 tests (refer to Figure 1). The detection rate is based on classification accuracy, hence the parameters for NBPSO algorithm are set as $\alpha = 1$. The population size is set to 20, C1 and C2 are set to 2 and the weight values lie between 0.5 to 1.5. Note that the SVM classifier was trained and tested by 90:10 percentage split in each dataset respectively. The obtained classification accuracy was illustrated by the form of 'average $\pm$ standard deviation'. The results showed that the proposed NBPSO-based feature selection with the SVM classifier resulted in higher detection rate across all datasets as shown in Table 2.

Figure 1 illustrates our evaluation based on 4 tests, which means in the first and the best test, the SVM is trained and tested by three categories 1C, 4C and 5C based on different features as input. After optimization of the classifier parameters, the best detection rate and FPR obtained are 98.99% and 0.2 respectively. On the other hand, the last test was based on 1C and 5C categories consisted of selected the relevant features. In this stage, the SVM created the performance up to 91.49%. This result indicated that the number of the relevant features eliminated could decrease the performance exactly.

This study selected 12 relevant features from the extracted 20 features in the previous studies to improve the detection rate and to evaluate the ability of feature selection method (NBPSO). The results showed that the best performance was obtained with only the selected relevant features. Although in some combined categories they contain low number of relevant features such as C2, they achieved a reasonable performance to 91.49% detection rate and 0.3% FPR.

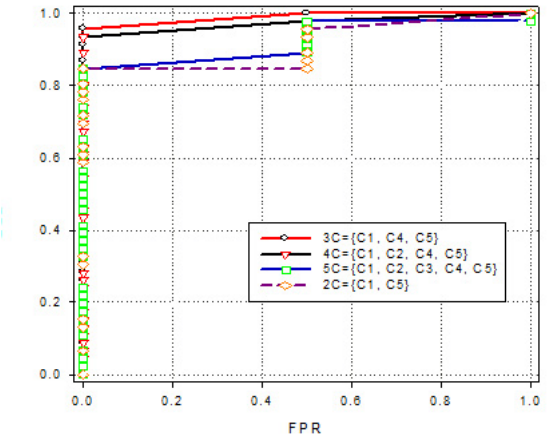

Fig. 1. ROC curve based on four (4) classifiers

5 ROC Curves and AUC Analysis

The receiver operating characteristics (ROC) graph as shown in Figure 1 illustrates classifier performance. Today, this technique has been used in machine learning because the accuracy of the classifier is not a robust measurement. ROC identifies the relationship between the false positive rate (FPR) and true positive rate (TPR). The best performance from the experiments were related to 3C, 4C, 5C, and 2C, respectively. Meanwhile, the best FPR and detection rate were 0.1% and 97.77% for 4C category. The CPU time, on the opposite, considered the impact of the number of features on the performance and computational cost. The time and cost were different by changing the number of features. Table 4 shows the changes in the CPU time and computational cost when the number of features decreased from 5 to 3, which means 63.65% to 37.2% and 215.67(s) to 173.56(s) respectively.

Table 4. The AUC results of combination features

Number of Features	AUC Value
2	94.30
3	98.21
4	98.90
5	97.68

The results in Table 4 indicated the AUC (Area Under the Curve) result is close to 1 for the 3C, 4C and 5C categories. In fact, the results showed that the best combination is related to 3C. Other combinations of features such as 2C,

4C and 5C present a good detection rate and the AUC result, although they do not contain all relevant features. Thus, the analysis presents an important point in which the categories that contain the above features may have a high detection rate and low false positive rate.

It is also noteworthy to mention that the detection rate is insufficient to assess a classifier's performance since the achieved results show that FPR has more to offer that the detection rate. This study proved this point by doing the famous statistical test, namely one-way ANOVA (Analysis of Variance). The different experiences executed based on various thresholds from 3 to 5. The F-ratio identified by ANOVA test is 9.24 ($p < 0.001$), which proved that a decrease in the detection rate is not the most important factor. Based on Table 3, eventhough 3C detects the phishing emails better than other classifiers to 98.99% of the detection rate, but the 4C classifier achieved a detection rate of 97.77% with 0.1 FPR that achieved a lower error rate in comparison with other classifiers. In addition, the best AUC of this classifier is 98.90% as compared to other classifiers.

6 Conclusions

In this study, an attempt was made to develop a spam/phishing email detection system to detect social engineering attacks. This paper proposed the New Binary Particle Swarm Optimization (NBPSO) algorithm for feature selection together with a Support Vector Machine (SVM) classifier for classification. The system was tested via a classification experiment using three datasets, namely SpamAssassin, SpamEmail, and PhishingCorpus. The experimental results, in comparison with results from previous studies, indicate that the detection system was able to reduce the number of features from 20 to 12 features, hence reducing its dimensionality. As the consequence, the accuracy rate has increased and the false positive rate (FPR) hit a lower percentage. Note that FPR represents system's reliability and has been proven by the literature that it is more important than the accuracy rate in the case of false alarm. In this work, FPR was accessed using the ROC and AUC curve.

One of the important points in classification is the parameter optimization that needs to be tuned and tested with different datasets for better classifier performance. In the future, we hope to test other datasets and to apply other metaheuristic algorithms. Comparisons will be in terms of dimensionality reduction and complexity.

Acknowledgments. This project is sponsored by the Malaysian Ministry of Higher Education (MOHE) under the Exploratory Research Grant Scheme.

References

1. Miller, T.: Social Engineering: Techniques that can Bypass Intrusion Detection Systems, http://www.stillhq.com/pdfdb/000186/data.pdf
2. Gorton, A.S., Champion, T.G.: Combining Evasion Techniques to Avoid Network Intrusion Detection Systems. Skaion (2004)
3. Dodge, R.C., Carver, C., Ferguson, A.J.: Phishing for User Security Awareness. Computers & Security 26(1), 73–80 (2007)
4. Hoeschele, M., Rogers, M.: Detecting Social Engineering. In: Pollitt, M., Shenoi, S. (eds.) Advances in Digital Forensics. IFIP, vol. 194, pp. 67–77. Springer, Heidelberg (2005)
5. Ashish, T.: Social Engineering: An Attack Vector Most Intricate to Tackle. Technical Report, Infosecwriters (2007)
6. Olivo, C.K., Santin, A.O., Oliveira, L.S.: Obtaining the Threat Model for E-mail Ohishing. Applied Soft Computing (2011)
7. Ruan, G., Tan, Y.: A Three-Layer Back-Propagation Neural Network for Spam Detection using Artificial Immune Concentration. Soft Computing-A Fusion of Foundations, Methodologies and Applications 14(2), 139–150 (2010)
8. Engelbrecht, A.P.: Fundamentals of Computational Swarm Intelligence, vol. 1. Wiley, London (2005)
9. El-Alfy, E.S.M., Abdel-Aal, R.E.: Using GMDH-based Networks for Improved Spam Detection and Email Feature Analysis. Applied Soft Computing 11(1), 477–488 (2011)
10. Ma, L., Ofoghi, B., Watters, P., Brown, S.: Detecting Phishing Emails using Hybrid Features. In: Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, pp. 49–497 (2009)
11. Oliveira, A.L.I., Braga, P.L., Lima, R.M.F., Cornelio, M.L.: GA-based Method for Feature Selection and Parameters Optimization for Machine Learning Regression Applied to Software Effort Estimation. Information and Software Technology 52(11), 1155–1166 (2010)
12. Chandrasekaran, M., Narayanan, K., Upadhyaya, S.: Phishing Email Detection based on Structural Properties. In: NYS Cyber Security Conference, pp. 1–7 (2006)
13. Toolan, F., Carthy, J.: Phishing Detection using Classifier Ensembles. In: eCrime Researchers Summit, pp. 1–9 (2009)
14. Sirisanyalak, B., Sornil, O.: Artificial Immunity-based Feature Extraction for Spam Detection. In: International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, pp. 359–364 (2007)
15. Lai, C.H.: Particle Swarm Optimization-aided Feature Selection for Spam Email Classification, p. 165. IEEE, Kumamoto (2007)
16. Ramadan, R.M., Abdel-Kader, R.F.: Face Recognition using Particle Swarm Optimization-based Selected Features. International Journal of Signal Processing, Image Processing and Pattern Recognition 2(1), 51–66 (2009)
17. Guyon, I., Matic, N., Vapnik, V.: Discovering Informative Patterns and Data Cleaning. In: KDD Workshop, pp. 145–156 (1994)
18. Chen, J., Guo, C.: Online Detection and Prevention of Phishing Attacks. In: 1st Int. Conference on Communications and Networking, pp. 1–7 (2006)

19. Mukkamala, S., Sung, A.: Significant Feature Selection using Computational Intelligent Techniques for Intrusion Detection. In: Advanced Methods for Knowledge Discovery from Complex Data, pp. 285–306 (2005)
20. Macia-Perez, F., Mora-Gimeno, F., Marcos-Jorquera, D., Gil-Martinez-Abarca, J.A., Ramos-Morillo, H., Lorenzo-Fonseca, I.: Network Intrusion Detection System Embedded on a Smart Sensor. IEEE Transactions on Industrial Electronics 58(3), 722–732 (2011)