Email Spam Detection Using Differential Evolution Negative Selection Algorithm

[1]Ismaila Idris, [2]Ali Selamat

1, First Author *Software Engineering Research Group (SERG), Knowledge Economy Research Alliance and Faculty of Computing, Universiti Teknologi Malaysia, 81310 UTM ismi_idris@yahoo.co.uk*

*2,Corresponding Author *Software Engineering Research Group (SERG), Knowledge Economy Research Alliance and Faculty of Computing, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia. aselamat@utm.my*

Abstract

In this paper, we propose a modification of machine learning techniques inspired by human immune system called negative selection algorithm (NSA) with differential evolution (DE) code-name NSA-DE; in order to deal with the growing problem of unsolicited email in the mail box. The evolutionary algorithm generates detectors at the random detector generation phase of negative selection algorithm. NSA-DE uses local differential evolution for detector generation and local outlier factor (LOF) as fitness function to maximize the distance between generated detector and non-spam space. The theoretical analysis and the experimental result show that the proposed NSA-DE model performs better than the standard NSA.

Keywords*: Detectors, email, spam, non-spam, negative selection algorithm, differential evolution.*

1. Introduction

Email is now part of millions of people life in the world today. It has change the way man collaborate and work; it is the most cheapest, popular and fastest means of communication [1]. The task of email classification based on machine learning is shared into sub-tasks [2, 3]. The initial task is the collection of data and email message representation. Secondly is the selection of email feature and dimensional reduction of features [4], finally is the mapping of both training and testing set for classification of email. The essence of classification is to distinguish between spam and non-spam email [5]. Quite a lot of machine learning techniques for email spam detection model have been proposed but no one ever combined negative selection algorithm (NSA) and differential evolution (DE).This paper proposes an improved solution for email spam detection inspired by artificial immune system by adapting spam detection generation techniques with negative selection algorithm and differential evolution. The random generation of detector in negative selection algorithm is replaced with differential evolution for detector generation. The organization of the paper is as follows: introduction is in section 1, Section 2 discusses the related work in negative selection algorithm, the proposed improved model and its constituent framework was discussed in Section 3. Empirical studies and data analysis is presented in Section 4, Section 5 discusses the experimental results, and the conclusion and recommendation is in Section 6.

2. Related Work

Artificial immune system based email spam detection was proposed by [6] with the use of spam-assassin corpus and Bayesian combination for the detector weight. Messages were scored by simple sum of the message matched by non-spam in the detector space and also the use of bayes scores. Words from the dictionary and patterns extracted from set of messages are considered in detector generation aside the commonly used filters in order to be assured of the message classification. The approach of scoring features or feature weighting during and after the matching process does not help in the selection of important features for spam detection due to its computational cost. A genetic optimized spam detection using artificial immune system (AIS) was proposed by [7]. The genetic

algorithm optimized AIS to cull old lymphocytes (Replacing the old lymphocyte with new ones) and also check for new interest for users in a way that is similar. In updating intervals such as the number of receive messages with respect to time, user request and so on, many choices are used in selecting the update intervals which is the aim of using the genetic algorithm. The experimental work was done with spam-assassin with 4147 non-spam message and 1764 spam message. An improved negative selection algorithm that introduces a novel training is also proposed by [8]. The technique was implemented in the training phase to generate candidate detectors in order to cover the non-self region. The paper proposed a further training negative selection algorithm whose unique training and testing stage produce good classification performance. The main aim of the technique is to modify the testing stage by reducing the computational cost while improving the self region coverage.

3. The proposed improved model and its constituent frameworks

3.1 Implementation of negative selection algorithm

The real value negative selection algorithm is encoded in real valued for classifying non-spam and spam. The dataset used in this research is implemented in real value, there is need to define the non-spam and the spam space. The non-spam space is the normal state of a system while the spam space is the abnormal state of a system. The candidate detector is randomly generated and then compared to the non-spam samples. Candidate detectors that do not match any sample of the non-spam set are accepted as viable detectors. Candidate detectors that matches sample of the non-spam set are discarded as unwanted detector. The non-spam sample in a real value negative selection algorithm is represented in N-dimensional points and a non-spam radius Rs, as training dataset. In clearer terms, let equation (2) represents the non-spam space.

$$S = \{X_i | i = 1, 2, \cdots m; Rs = r\} \quad (1)$$

$X_i$are some point in the normalized N-dimensional space.

$$X_i = \{x_{i1}, x_{i2}, x_{i3} \cdots x_{iN}\}, i = 1, 2, 3 \cdots m \quad (2)$$

The entire normalized sample space $I \subset [0,1]^N$, the spam space can then be represented as $S = I - NS$ where S is spam and NS is non-spam.

$$d_j = (C_j, R^d j) \quad (3)$$

Equation (3) denote one detector where $C_j = \{C_{j1}, C_{j2}, C_{j3} \cdots C_{jN}\}$ is the detector center respectively, R_j is the detector radius. The Euclidean distance is used as the matching measurement. The distance between non-spam sample X_i and the detector d_j can be defined as:

$$L(X_i, d_j) = \sqrt{(x_{i1} - C_{j1})^2 + \cdots + (x_{iN} - C_{jN})^2} \quad (4)$$

$L(X_i, d_j)$ is compared with the non-spam space threshold Rs, obtaining the matching value of $\ltimes$

$$\ltimes = L(X_i, d_j) - Rs \quad (5)$$

The detector d_j fails to match the non-spam sample X_i if $\ltimes > 0$, therefore if d_j does not match any non-spam sample, it will be retained in the detector set. The detector threshold R^d, j of detector d_j can be defined as:

$$R^d, j = \min(\ltimes), \text{if} \ltimes \leq 0 \quad (6)$$

If detector d_j matches the non-spam sample, it will be discarded. This will not stop the generation of detector until the required detector set is reached and the required spam space covered. The generated detector set can then be used to monitor the entire system.

3.2. The proposed improved negative selection algorithm model

The detector generation as shown in real valued negative selection algorithm in section 3.1 is vital in enhancing the performance of negative selection algorithm. Random generation of detector by the real value negative selection algorithm was improved with the introduction of differential evolution (DE) for generating detectors and the local outlier factor (LOF) as fitness function to maximize the spam space. This is as a result of the quest for efficiently trained negative selection algorithm model for purely normal detectors. The local outlier factor maximized the distance between the generated spam detector and non-spam space.

3.2.1 Definition of non-spam space

In the case of real value negative selection algorithm, there is need to define the non-spam and the spam space. The non-spam space is the normal state of a system while the spam space is the abnormal state of a system.
Let's assume the non-spam space to be S
S is defined as:

$$s = (s_{1\ldots}s_n) = \begin{bmatrix} s_{11} & \cdots & s_{1m} \\ \vdots & \ddots & \vdots \\ s_{n1} & \cdots & s_{nm} \end{bmatrix} \quad (7)$$

$S_{ij} \in K^m, \ \ i = 1, \cdots, n; \ \ j = 1, \cdots, m$
S is normalized as follows:

$$S_i = \frac{s_i}{\|s_i\|} \quad (8)$$

Therefore, s_i is the i^{th}non-spam unit; and s_{ij} is the j^{th} vector of the i^{th}non-spam unit.

3.2.2 Detector generation Parameters and implementation:

Population size: 100; the mutation factor F= random number between 0.5 and 1. Preferred value 0.7; Crossover rate C= random number between 1 and 0 Preferred value 0.5. Initializing the population we set:
j = 0; $i = 1,2\ 3 \cdots, P$; where P is the size of the population.
A differential candidate vector is added to the population of the vector by mutation. For each target detector vector $x_{i,}[J\ ]$, a mutation vector is produced using equation (9).

$$v_i[J] = X_{r_1}[J\ ] + F(X_{r_2}[J\ ] - X_{r_3}[J\ ]) \quad (9)$$

F is the mutation factor; it provides the amplification between two individual differences. $(X_{r_2}[J\ ] - X_{r_3}[J\ ])$. It is usually taking in the range [0,1] to avoid search stagnation where $r_{1,}r_{2,}r_3 \in \{1,2,3 \cdots, p\}$ are choosing randomly, where p is the number of population. By replacing parameters from the target candidate detector vector to generate a trial candidate detector vector with the corresponding parameters for randomly generated mutant, we apply recombination to the population.
Therefore; crossover constant C = $(0 \leq c \leq 1)$
rand $J \in [0,1]$; is a random number that is less than c.

$$t_i[J] = \begin{cases} v_i[J] & if\ rand \leq c \\ x_i[J] & otherwise \end{cases} \quad (10)$$

Where J = 1, 2, 3⋯,d, d is the number of parameter to be optimized.
If the trial candidate detector vector $t_i[J]$ has equal or lower value than the target candidate detector vector $x_i[J]$, the target candidate detector vector is replaced in the next generation. e.g. replaces x_i with t_i or else x_i is retained in the population for at least one more generation. This is represented in equation (11)

$$f(t_i[J]) \leq f(x_i[J]) \quad (11)$$

The process of mutation, recombination and selection are required once a new population is installed until specific termination criteria are reached. From equation (8) of the normalized non-spam space, the non-spam space is represented in equation (1) with radius Rs in section 3.2.1. Computing the generation of candidate detector of differential evolution in the spam space is as shown in section 3.1 if detector d_j match the non-spam sample, it will be discarded. This will not stop the generation of detector until the required detector set is reached and the required spam space covered. After the generation of detectors in the spam space, the generated detectors can then monitor the status of the system. If some other new email (test) samples matches at least one of the detectors in the system, it is assume to be spam which is abnormal to the system but if the new email (test) sample does not match any of the generated detectors in the spam space, it is assume to be a non-spam email.

3.2.3 Computation of fitness functions in the spam and non-spam space

One most important quality of spam and non-spam detector space is how distant the generated spam detector is from the non-spam space; this helps in improving the proposed model. We decided to employ the local outlier factor as a fitness function to maximize the distance between the generated spam detector and the non-spam space. An outlier can be defined as a data point that is not the same as the remaining data with respect to some measures. The technique will model the data point with the use of a stochastic distribution [9] and the point is determined to be an outlier based on its relationship with the model. The outlier detection algorithm was proposed as fitness function in this study in order to maximize the generated spam detector space which is very unique in computing the full dimensional distance from one point to another [10, 11] while computing the density of the local neighbourhood.

Step 1. Let's assume $k-$distance (i) to be the distance of the generated detector (i) to the nearest neighborhood (non-spam).
Step 2. Set of k nearest neighbor (non-spam element) includes all spam detectors at this distance.
Step 3. Set S of k nearest neighbor is denoted as $N_k(i)$; Here non-spam space = S
Step 4. This distance is used to define the reach-ability distance.
Step 5. Reach-ability-distance$_k(i,s) = \max\{k-distance\ (s), d_{i,s}\}$
Step 6. The local reach-ability density (LRD) of r is defined as follows:

$$\text{lrd}(i) = 1/\left(\frac{\sum s \in N_k(i) reachability-distance_k(i,s)}{|N_K(i)|}\right) \quad (12)$$

Equation (12) is the quotient of the average reach-ability distance of the generated detector i from non-spam element. It is not the average reach-ability of the neighbor from i but the distance from which it can be reached from its neighbor.We then compares the local reach-ability density with those of its neighbor using the equation below:

$$\text{LOF}_K(i) = \frac{\sum s \in N_k(i) \frac{lrd(s)}{lrd(i)}}{|N_k(i)|} = \frac{\sum s \in N_k(i) lrd(s)}{|N_K(i)|} / lrd(i) \quad (13)$$

Equation (13) shows the average local reach-ability density of the neighbor divided by the candidate detectors own local reach-ability density. In this scenario, values of spam detector

approximately 1 indicates that the detector is comparable to its neighbor (not an outlier) and value below 1indicates a dense region (which is an inliers) while value larger than 1indicates an outlier. The major idea of this technique is to assign to each detector the degree of been an outlier. The degree is called the local outlier factor (LOF) of the detector as shown in equation (13).

4. Empirical study and dataset analysis

Spam base dataset was required for the research. The entire dataset was divided using stratified sampling approach into training and testing set. 70% of the entire dataset was used for training and 30% of the remaining dataset was used for testing the model. The corpus bench mark is obtained from spam base dataset which is an acquisition from email spam messages. In acquiring this email spam message, it is made up of 4601 messages and 1813 (39%) of the message are marked to be spam messages and 2788 (61%) are identified as non-spam and was acquired by [12].

5. Experimental results and discussion

At 100, 200, 300, 400 and 500 generated detectors with threshold value of 0.4, figure 1 shows summary and comparison of results in percentage for NSA and NSA-DE model.

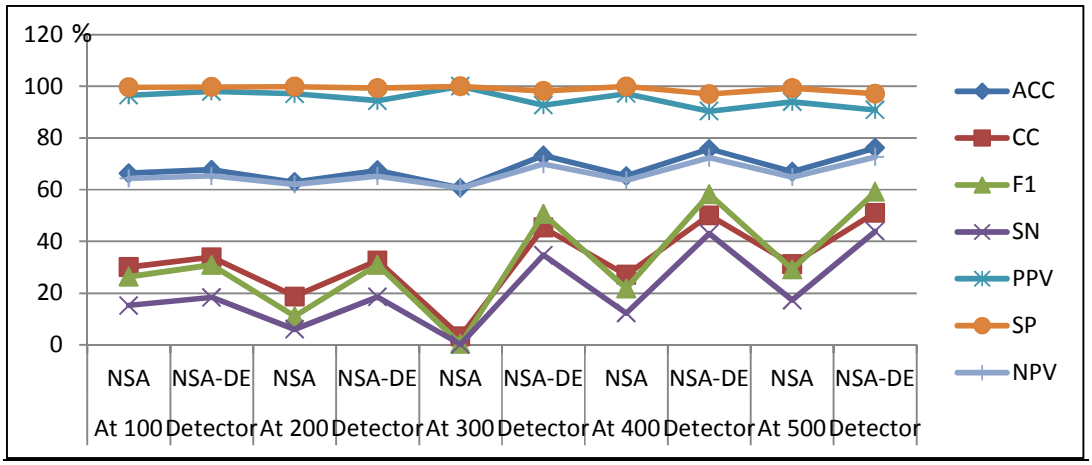

Figure 1. Result of NSA and NSA-DE model.

**Note: ACC= Accuracy, CC= Correlation coefficient, F1= F measure, SN= sensitivity, PPV= Positive prediction value, SP= Specificity and NPV= Negative prediction value.

Accuracy measures the percentage of sample that is correctly classified. It can be observed that the proposed improved model performs better than negative selection algorithm model. Figure 1 shows best accuracy at 500 generated detectors with threshold value of 0.4. Accuracy for negative selection algorithm is at 66.98% while the improved negative selection algorithm with differential evolution is at 76.18%. Other measuring standard are as represented in the figure above.

The Average accuracy of the standard negative selection algorithm is at 65.147%, the improved negative selection algorithm and differential evolution model is at 69.383%. At 7000 generated detectors with threshold value of 0.4, accuracy for negative selection algorithm is 68.863% while improved negative selection algorithm and differential evolution is at 83.056%.

6. Conclusion and recommendation

In this research, a new improved model that combines negative selection algorithm (NSA) with differential evolution (DE) has been proposed and implemented. The uniqueness of this model is that DE was implemented at the random generation phase of NSA. The detector generation phase of NSA

determines how robust and effective the algorithm will perform. DE implementation with local outlier factor (LOF) as fitness function no doubt improved the detector generation phase of NSA. In totality, the empirical report has shown the superiority of the proposed NSA-DE improved model over the NSA model. The proposed improved systems will be useful in other applications since negative selection algorithm solves a vast number of complex problems. This research should be viewed as an improvement in the field of computational intelligence. Based on the promising result generated from the research; as future work, it is suggested that this research should be considered as a viable tool for any newly proposed system in email spam detection problem that is based on detector generation. Future work will be on parallel hybridization of two evolutionary algorithms to perform single task of detector generation.

Acknowledgements

The Universiti Teknologi Malaysia (UTM) and Ministry of Higher Education (MOHE) Malaysia, under research grant R.J130000.7828.4F087 are acknowledged for some of the facilities utilized during the course of this research.

References

[1] Whittaker S, Bellotti V, Moody P: Introduction to this special issue on revisiting and reinventing e-mail. . Human-Computer Interaction 2005; , 20 (1-2): :1-9.
[2] Duolin Liu: Research on Sentiment Classification of Chinese Micro Blog Based on Machine Learning. JDCTA: International Journal of Digital Content Technology and its Applications, 2013, 7(3): pp. 395 ~ 402.
[3] Xuesong Yan, Wei Chen, Qinghua Wu, Liu. H: Data Classification Algorithm Based on Differential Evolution Algorithm. International Journal of Digital Content Technology and its Applications 2013, 7(7):pp. 406 ~ 413.
[4] Awad WA, ELseuofi SM: Machine Learning Methods for Spam E-mail Classification. International Journal of Computer Science & Information Technology (IJCSIT) 2011, Vol 3, No 1,.
[5] Xing Gao, Lu. Y: Automatic Text Clustering via Particle Swarm Optimization. JDCTA: International Journal of Digital Content Technology and its Applications 2012, Vol. 6, (No. 23):pp. 12 ~ 21.
[6] Oda T, White T: Immunity from Spam: An Analysis of an Artificial Immune System for Junk Email Detection. In: Artificial Immune Systems. Edited by Jacob C, Pilat M, Bentley P, Timmis J, vol. 3627: Springer Berlin Heidelberg; 2005: 276-289.
[7] Mohammad AH, Zitar RA: Application of genetic optimized artificial immune system and neural networks in spam detection. Applied Soft Computing 2011, 11(4):3827-3845.
[8] Gong M, Zhang J, Ma J, Jiao L: An efficient negative selection algorithm with further training for anomaly detection. Knowledge-Based Systems 2012, 30(0):185-191.
[9] Sajesh TA, Srinivasan MR: Outlier detection for high dimensional data using the Comedian approach. Journal of Statistical Computation and Simulation 2011, 82(5):745-757.
[10] Ramaswamy S, Rastogi R, Shim K: Efficient algorithms for mining outliers from large data sets. SIGMOD Rec 2000, 29(2):427-438.
[11] Knorr EM, Ng RT: Algorithms for Mining Distance-Based Outliers in Large Datasets. In: Proceedings of the 24rd International Conference on Very Large Data Bases. Morgan Kaufmann Publishers Inc.; 1998: 392-403.
[12] Mark Hopkins, Erik Reeber, George Forman, Jaap Suermondt: "Spam Base Dataset," Hewlett-Packard Labs http://archiveicsuciedu/ml/datasets/Spambase, 1999 1999.