# AWSIKE 2014

## The First Asian Winter School on Information and Knowledge Engineering

February 12-14, 2014

Ba Ria - Vung Tau University, Vietnam

# Proceedings

Sponsored by:

# Evolutionary Modified Detector Generation Model in Negative Selection Algorithm for Email Spam Detection

Ismaila Idris[1], Ali Selamat[2]

[1] Software Engineering Department, Faculty of Computing,
Universiti Tecknologi Malaysia.
Ismi_idris@yahoo.co.uk
[2] Software Engineering Department, Faculty of Computing,
Universiti Tecknologi Malaysia.
aselamat@utm.my

***Abstract.*** To deal with the growing problem of unsolicited email in the mail box, a modification of machine learning techniques inspired by human immune system called negative selection algorithm (NSA) is proposed; differential evolution (DE) is implemented to improve the random detector generation in negative selection algorithm. The model is called NSA-DE. 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. 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]. Though, it recorded success in a lot of human activities, improving group communications, felt on the growth of business and also national development in a positive path. It is one of the technologies that as direct impact on human life. The major short coming of this technology is the increase in unsolicited email message that recipient receives. One significant and growing task that resulted from unsolicited email is the classification of email. This pose a problem among cooperate organizations and individuals trying to solve the menace of email spam. The task of email classification is shared into sub-tasks. The initial task is the collection of data and email message representation. Secondly is the selection of email feature and dimensional reduction of features [2], 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. Quite a lot of machine learning techniques for email spam detection model have been proposed with no work on 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 differential evolution (DE) was implemented to generate detectors in negative selection algorithm in order to cover the spam space instead of the original random generation of detector used in negative selection algorithm. The paper was organized in to six sections, Section 1 is the introduction, 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 were in Section 4, Section 5 discuss the experimental results while conclusion and recommendation was in Section 6.

## 2 Related Work

Artificial immune system (AIS) is a new mechanism implemented in the control of email spam [3], it uses pattern matching in representing detectors as regular expression in the analysis of message. A weight is assigned to detector which was decremented or Incremented when observing expression in spam message with the classification of the message based on threshold sum of the weight of matching detectors. The system is meant to be corrected by either increasing or decreasing of all matching detector weight with 1000 detector generated from spam-assassin heuristic and personal corpus. The results were acceptable base on few number of detectors used. A comparison of two techniques to determine message classification using spam-assassin corpus with 100 detectors was also proposed by [4]. This approach is like the previous techniques but the difference is the increment of weight where there is recognition of pattern in spam messages. Random generation of detector does not help in solving problem of best selected features; though, feature weights are updated during and after the matching process of the generated detectors. The weighting of features complicates the performance of the matching process. In conclusion, the present techniques are better than the previous due to its classification accuracy and slightly improved false positive rate. More experiment was performed by [5] 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 each non-spam in the detector space and also the use of Bayes scores. 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 combination of support vector machine (SVM) and artificial immune system (AIS) was proposed by [6]. With the use of binary features with same feature selection in [7]. The support vector acquired after training SVM are implemented in the generation of initial detector set of the AIS and then the AIS was used in classification. During classification with AIS, detector with smallest Euclidean distance to the message was added to committee set with the major voting of detector in the set as the classification. A genetic optimized spam detection using artificial immune system (AIS) was proposed by [8]. The implementation of different pattern recognition scheme inspired by biological immune system in order to identify uncommon situations like the email spam [8-12], unfortunately has not been able to produce outstanding result. An improved negative selection algorithm that introduces a novel training is also proposed by [13]. The technique was implemented in the training phase to generate candidate detectors to cover the non-self region.

## 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 detectors. 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.

$$\mathrm{S} = \{X_i | i = 1, 2, \cdots m; Rs = r\} \tag{1}$$

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

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

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.

$$d_j = (C_{j\,,} R^d j) \tag{3}$$

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:

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

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

$$\ltimes\text{=L}(X_i, d_j\,)\text{–}\,Rs \tag{5}$$

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:

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

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. 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) [14] and the local outlier factor (LOF) as fitness function. 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:

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

$S_{ij} \in K^m, \;\; i \; = 1, \cdots, n; \;\; j \; = 1, \cdots, m$

$S$ is normalized as follows:

$$S_i \; = \frac{s_i}{||s_i||} \tag{8}$$

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).

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

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 $r_{1,}r_{2,}r_3 \ \in \{1,2,3 \cdots, p\}$ 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.

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

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)

$$f(t_i\ [J\ ]) \leq f(x_i\ [J\ ]) \tag{11}$$

The process of mutation, recombination and selection are required once a new population is installed until specific termination criteria are reached.

J←J+1 determine the incremental features until the maximum number of generated detectors is reached. This makes the local differential evolution unique as best features are acquired one after the other in order to attain best combination. Figure 1 illustrate the framework of the proposed model.

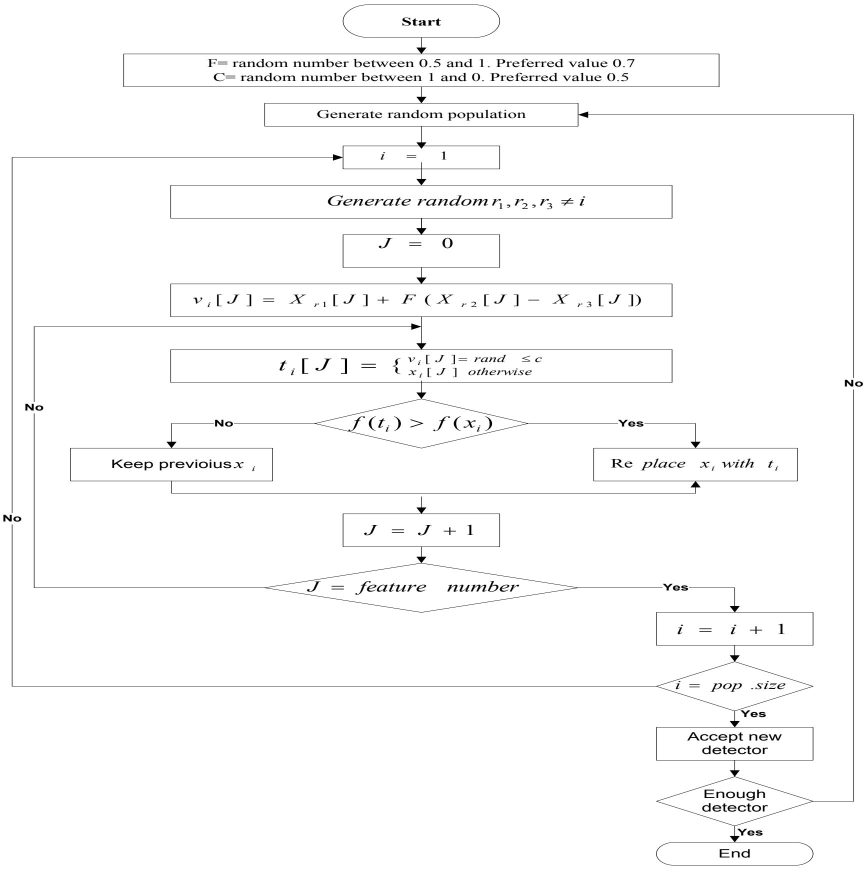
Start
F= random number between 0.5 and 1. Preferred value 0.7
C= random number between 1 and 0. Preferred value 0.5
Generate random population
$i = 1$
$Generate\ random\ r_1, r_2, r_3 \neq i$
$J = 0$
$v_i[J] = X_{r1}[J] + F(X_{r2}[J] - X_{r3}[J])$
$t_i[J] = \{ \begin{smallmatrix} v_i[J] = rand \leq c \\ x_i[J]\ otherwise \end{smallmatrix}$
$f(t_i) > f(x_i)$
No
Yes
Keep previoius $x_i$
Re $place\ x_i\ with\ t_i$
$J = J + 1$
$J = feature\ number$
Yes
No
No
$i = i + 1$
$i = pop.size$
Yes
Accept new detector
Enough detector
Yes
End
No

Figure 1: Framework of proposed improved NSA-DE   detector generation model.

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 particle swarm optimization 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 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 [15] 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 the 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 [16, 17] while computing the density of the local neighbourhood.

- Let's assume $k$ −distance $(i)$ to be the distance of the generated detector $(i)$ to the nearest neighborhood (non-spam).
- Set of $k$ nearest neighbor (non-spam element) includes all spam detectors at this distance.
- Set S of $k$ nearest neighbor is denoted as $N_k(i)$; Here non-spam space = S
- This distance is used to define the reachability distance.
- Reachability-distance$_k(i, s) = \max\{k - distance(s), d_{i,s}\}$
- The local reachability density (LRD) of r is defined as

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

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

$$\mathrm{LOF_K}\,(i) = \frac{\sum s\in N_k(i)\frac{lrd(s)}{lrd(i)}}{|N_k(i)|} = \frac{\sum s\in N_k(i) lrd\ (s)}{|N_K(i)|}/lrd(i) \tag{13}$$

Equation (13) shows the average local reachability density of the neighbor divided by the candidate detectors own local reachability 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 inlier) while value larger than 1indicates an outlier. The major idea of this technique is to assign to each detector the degree of being an outlier. The degree is called the local outlier factor (LOF) of the detector as shown in equation (13). The methodology of the computation of LOF for all detectors is explained in the steps below:

## 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 [18].

## 5 Experimental results and discussion

At 100, 200, 300, 400 and 500 generated detectors with threshold value of 0.4, figure 2 gives summary and comparison of results in percentage for NSA and NSA-DE model.

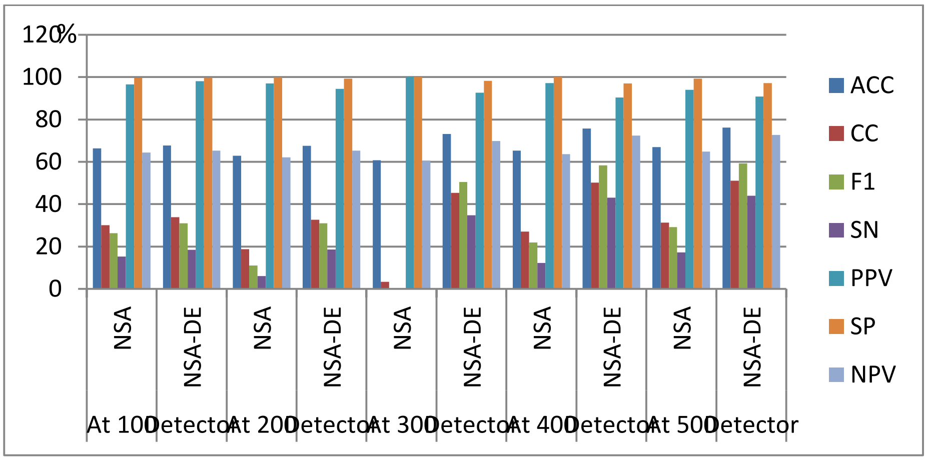

Figure 2. Result of NSA and NSA-DE model.

**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. The figure 2 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 as 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) under research grant 03H02 and Ministry of Science, Technology & Innovations Malaysia, under research grant 4S062 are hereby acknowledged for some of the facilities utilized during the course of this research work.

## 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. 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,.
3. Oda T, White T: Developing an Immunity to Spam. In Genetic and Evolutionary Computation — GECCO 2003. Volume 2723. Edited by Cantú-Paz E, Foster J, Deb K, Davis L, Roy R, O'Reilly U-M, Beyer H-G, Standish R, Kendall G, Wilson S et al: Springer Berlin Heidelberg; 2003:231-242.
4. Oda T, White T: Increasing the accuracy of a spam-detecting artificial immune system. In Evolutionary Computation, 2003 CEC '03 The 2003 Congress on: 8-12 Dec. 2003 2003; 2003:390-396 Vol.391.
5. Oda T, White T: Immunity from Spam: An Analysis of an Artificial Immune System for Junk Email Detection. In Artificial Immune Systems. Volume

3627. Edited by Jacob C, Pilat M, Bentley P, Timmis J: Springer Berlin Heidelberg; 2005:276-289.

6. Guangchen R, Ying T: Intelligent Detection Approaches for Spam. In Natural Computation, 2007 ICNC 2007 Third International Conference on: 24-27 Aug. 2007 2007; 2007:672-676.
7. Bezerra G, Barra T, Ferreira H, Knidel H, Castro L, Zuben F: An Immunological Filter for Spam. In Artificial Immune Systems. Volume 4163. Edited by Bersini H, Carneiro J: Springer Berlin Heidelberg; 2006:446-458.
8. 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.
9. Guzella TS, Mota-Santos TA, Uchôa JQ, Caminhas WM: Identification of SPAM messages using an approach inspired on the immune system. Biosystems 2008, 92(3):215-225.
10. Visconti A, Tahayori H: Artificial immune system based on interval type-2 fuzzy set paradigm. Applied Soft Computing 2011, 11(6):4055-4063.
11. Pérez-Díaz N, Ruano-Ordás D, Fdez-Riverola F, Méndez JR: SDAI: An integral evaluation methodology for content-based spam filtering models. Expert Systems with Applications 2012, 39(16):12487-12500.
12. Afaneh S, Zitar RA, Al-Hamami A: Virus detection using clonal selection algorithm with Genetic Algorithm (VDC algorithm). Applied Soft Computing 2013, 13(1):239-246.
13. 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.
14. Poikolainen I, Neri F: Differential Evolution with Concurrent Fitness Based Local Search. In Congress on Evolutionary Computation (CEC), 2013 IEEE 20-23 June 2013 2013; 2013:384-391.
15. 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.
16. Ramaswamy S, Rastogi R, Shim K: Efficient algorithms for mining outliers from large data sets. SIGMOD Rec 2000, 29(2):427-438.
17. 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.
18. Hopkins M, Reeber E, Forman G, Jaap S: Spam Base Dataset. Hewlett-Packard Labs 1999.