Journal of Ambient Intelligence and Humanized Computing (2019) 10:143–152
https://doi.org/10.1007/s12652-017-0621-2

ORIGINAL RESEARCH

Clustered negative selection algorithm and fruit fly optimization for email spam detection

Ramdane Chikh[1] · **Salim Chikhi**[2]

Received: 2 May 2017 / Accepted: 6 November 2017 / Published online: 24 November 2017

Abstract
At present, spam is an actual and increasing problem that compromises email communications across the world. Thus, several solutions have been proposed to stop or reduce the amount of this threat. However, methods based on negative selection algorithm (NSA) lack continuous adaptability and suffer from low detection performance. Moreover, these methods require a large number of detectors to cover all non-self spaces. Thus, this study proposes a new e-mail detection approach based on an improved NSA called combined clustered NSA and fruit fly optimization (CNSA–FFO). The system combines actual NSA with k-means clustering and FFO to enhance the efficiency of classic NSA. Experiments results in spam benchmark show that the performance of CNSA–FFO is better than the classic NSA and NSA–PSO, especially in terms of detection accuracy, positive prediction, and computational complexity.

Keywords Artificial immune system · Negative selection algorithm · E-mail spam · Fruit fly optimization · K-means clustering

1 Introduction

E-mail is an important means of digital communication today; it is easy, quick, reliable, and lowest cost. However, e-mail spam is becoming a serious problem for e-mail users. The number of internet spam is continuously increasing daily. Various approaches and techniques have been proposed to resolve or reduce the amount of this threat. One of these techniques is artificial immune system (AIS), which is successfully applied for e-mail spam detection.

An AIS can be defined as a computational technique inspired by the principles and processes of the human immune system (HIS). Therefore, an AIS inspires ideas from HIS operations and applies them to computational problems (Forrest et al. 1994). Negative selection algorithm (NSA) is one of the main algorithms in AIS and has been successfully applied in various domains. The main focus of researchers in this field is to extract NSA properties that will be useful in designing an automatic solution for classification problems, such as detection of computer intrusions and anomaly detection.

The most important property of NSA is to distinguish the difference between self (normal) and non-self (abnormal) state using only normal data. This property has attracted the interest of the artificial intelligence community, and several methods have been proposed to improve the efficiency of such algorithms and extend their application. In an NSA-based spam detection approach, the self in the system is considered as non-spam, whereas the non-self is treated as spam.

In the present work, a new model based on NSA, called combined clustered NSA and fruit fly optimization (CNSA–FFO), is proposed to detect e-mail spam. This method initially employs k-means clustering to generate the self-set clusters and then uses the FFO algorithm (FFOA) for the training stage to optimize the random generated detectors. Two sets of detectors are generated, namely, the boundary self-detectors and non-self detectors. In the testing phase of the proposed CNSA–FFO, both cluster and detector sets are used to classify whether an e-mail sample is a spam. If the sample is undetected by any of the detectors,

✉ Ramdane Chikh
chikh_ram@univ-setif.dz; chikhram@yahoo.fr

Salim Chikhi
salim.chikhi@univ-constantine2.dz

[1] Faculty of Technology, Sétif 1 University, Sétif, Algeria

[2] MISC Laboratory, Constantine 2 University, Constantine, Algeria

then it should be assigned to the nearest set and added as a new detector. This mechanism can resolve the problem of the existence of holes and ensure the continuous adaptability of the model. Furthermore, the training and testing processes are performed only in a small part of space, which is obtained by clustering technique. Restricting the search can significantly decrease the number of detectors, computation time, and space complexity compared with the classic NSA.

The rest of the paper is structured as follows. Section 2 discusses the background and related works in NSA and spam detection. Section 3 presents the proposed work and its constituent framework. Section 4 explains the experimental results. The paper is concluded in Sect. 5.

2 Background and related works

2.1 Fundamentals of AIS

The AIS is a bio-inspired computational Intelligence. It developed by (Forrest et al. 1994). An AIS uses ideas from HIS operation and applies them to computational problems. A more detailed presentation of the HIS and AIS can be found in (Shelly and Wolfgang 2010; Dasgupta et al. 2011; Forrest et al. 1994; De Castro and Timmis 2003). One of the interesting mechanisms of the adaptive immune system is the self/non-self recognition. HIS can recognize which cells are its own (self) and which are foreign (non-self). Hence, it can build its defense against an attacker instead of self-destructing. Currently, the major theories of AIS research include NSA, clone selection, immune network theory, danger theory and positive selection. The NSA is one of the most applied and discussed model especially in anomaly detection (Ramdane and Chikhi 2014).

2.1.1 NSA

The NSA was inspired from the negative selection process of the adaptive immune system; it is the main algorithm of the AIS. This algorithm has the capability to differentiate between self-space (e.g., cells that are owned by the system) and non-self space (e.g., foreign entities to the system). This capability is obtained by the T-cells, which are a set of non-self reactive detectors.

NSA was first proposed by Forrest et al. (1994) who presented a framework that discriminates normal and abnormal entities. This algorithm has a special characteristic, that is, it only requires normal data in the training stage.

Generally, classic NSAs consist of two stages. First, the NSAs generate a detector set in the non-self space. If the detector does not match with the known self states, then it becomes a mature detector and is added to the detector set. Second, the unseen states are tested by the detector set. If an unknown state is matched by any mature detector, then the NSA indicates the presence of an anomaly. Although a diverse family of NSA has been developed, the essential characteristics of the original NSA introduced by Forrest et al. (1994) still remain. However, the original NSA has large time and space complexities (Forrest et al. 1994). Self and non-self detector representations reveal two types of NSA, namely, binary NSAs (BNSA) and the real-valued NSAs (RNSA).

2.1.1.1 BNSA The representing of self and detector is the basic step in designing NSA model. Forrest et al. (1994) developed the binary representation and R-Contiguous Matching Function, where both self and detectors elements are implemented as binary strings. The detector matches the self element or the tested sample if the binary strings have same bits in at least r contiguous places. For binary representation, there are several types of matching rules used to calculate this affinity: r-contiguous bits (rcb), r-chunks, landscape-affinity matching, Hamming distance and its variations. The BNSAs are suitable for the representation and search for discrete space. Despite its simplicity implementation, the binary representations of NSAs have some limitations for the real world problems (Ramdane and Chikhi 2014).

2.1.1.2 RNSA The limitation of BNSAs motivated (Gonzalez et al. 2002, 2003) to develop RNSA. RNSA employs real-value representation and Euclidean distance matching rule in generating real detectors. This high-level representation provides certain advantages, such as increased expressiveness, the possibility of extracting high-level knowledge from generated detectors, and in some cases, improved scalability.

Two RNSA models are discussed in the literature, namely, NSA with constant-sized detectors and NSA with variable-sized detectors. In the first model, the radii size of detector is constant and requires a large number of detectors to cover all the non-self spaces. To overcome this problem, new models of NSA with variable-sized detector are proposed. The next model of RNSA was proposed by (Zhou and Dasgupta 2009) and is called the V-detector, which is the latest and most mature version of the NSA. This model considered the several advantages of other versions and is currently the framework for numerous studies.

2.1.1.3 Recent improvements in NSA Recently, many variants of NSA have been proposed to overcome the drawbacks of the classic version, and most of them focused on detector generation mechanism. Researchers in this field aim to improve the algorithm efficiency by covering a non-self space with the optimal number of detectors and covering the holes by detectors with a small radius.

The following is the acronyms list of recent improvements and models of NSA:

- ANSA (Jinquan et al. 2009)
- EvoSeedRNSA (Jie et al. 2009)
- ORNSA (Guiyang et al. 2010)
- Optimized NSA (Aiqiang et al. et al. 2011)
- FtNSA (Maoguo et al. 2012)
- IVRNSA (Wu and and Zheng 2012)
- CB–NSA (Chen et al. 2013)
- PRR–2NSA (Zhenga et al. et al. 2013)
- EvoSeedRNSAII (Jie and Wenjian 2014)
- GF–RNSA (Wen et al. 2014)
- NSA–DE (Ismaila et al. 2014)
- HNSA–IDSA (Ramdane and Chikhi 2014)
- NSA–PSO (Ismaila et al. 2015)
- I-detector and OALI-detector (Li et al. 2015)
- IO–RNSA (Xiao et al. et al. 2015)
- BIORV–NSA (Lin et al. 2015)
- NSA-II (Abdolahnezhad and Banirostam 2016)
- OALFB–NSA and FB–NSA (Dong et al. 2016)

2.1.1.4 Applications of NSA Since its emergence, NSA has attracted the attention of many researchers and has been applied in various real-world applications. Its application domains are generally similar of those of computational intelligence approaches, such as artificial neural networks, evolutionary algorithms, and fuzzy systems.

NSA is applied in computer security (Forrest et al. 1994; Kim 2002; Wang and Zhang 2007; Ramdane and Chikhi 2014; Ismaila and Ali 2014; Ismaila et al. 2015), anomaly detection (Li et al. 2015, 2016; Jinquan et al. 2009; Gonzalez et al. 2002), data mining (Puteh et al. 2008), and optimization (Vieira et al. 2008).

2.2 K-means clustering

Clustering is an unsupervised learning technique that aims to partition an unlabeled data set into groups according to the similarities among its objects. Clustering has been applied successfully in many applications, including text mining, social web analysis, information discovery, bio-informatics, and image segmentation (Gang et al. 1979). The most used clustering algorithms are k-means, fuzzy c-means, hierarchical clustering, and mixture of Gaussians. This section focuses solely on k-means, which is one of the oldest, simplest, and widely used clustering algorithm.

K-means algorithm requires a matrix of S data points in n dimensions and a matrix of K initial cluster centers in n dimensions as inputs. The number of data points in cluster C_j is denoted by NC_j. D (S_i, C_j) is the Euclidean distance between point S_i and cluster C_j. The general procedure is to search for a k-partition with locally optimal within-cluster sum of squares by moving points from one cluster to another (Hartigan and Wong 1979).

The main disadvantage of k-means algorithm is that it may take a large number of iterations through dense datasets before it can converge to produce the optimal set of centroids.

2.3 FFOA

FFOA is a new global optimization algorithm proposed by (Pan 2011) and (Bo and Wen-Jing 2014); it is a bio-inspired evolutionary algorithm inspired by the food finding behavior of fruit flies. The sensory perception of fruit flies is better than that of other species, especially the sense of smell and vision. The olfactory organ of a fruit fly can gather various smells from the air and even a food source 40 km away. Afterward, the insect flies toward the food, uses its acute vision to find the food and where its fellows gather, and then it flies in that direction (Pan 2011).

Compared to other intelligent optimization algorithms, FFOA is simpler to understand, its adjustment parameters are less, its convergence speed is faster, and it is easier to implement. FFOA has been successfully applied in solving various problems in different domains, including function optimization, generalized regression network parameter optimization, and gray neural network parameter optimization. In addition, it can be combined with other techniques, such as decision trees, Bayesian theorem, fuzzy math, gray system, neural network, and AIS. However, the original algorithm still has some disadvantages, such as being easily trapped into a local optimal value and low accuracy. To overcome these limitations and extend its application area, many improvements and variants have been proposed for FFOA, such as adaptive mutation FFOA (Han and Liu 2013), binary FFOA (Wang et al. 2013), and modified FFOA (Liu et al. 2012). More details about FFOA can be found in the study of (Hazim and Mesut 2015).

2.4 Spam emails detection and NSA

E-mails are one of the most used forms of communication; they are simple, reliable, and economical. This simplicity and low cost qualify it to be the preferable way for advertising and sometimes employed as a mean to launch threats. One of these threats is spam e-mails; they are a problem that almost every e-mail user suffers from (Raed and Adel 2013).

The word "spam" usually denotes a particular brand of luncheon meat; however, in recent times, spam is used to represent a variety of junk or unwanted e-mails. Sending thousands of unsolicited messages to thousands of users all over the world with approximately no cost is now possible (Raed and Adel 2013).

Nowadays, spam has the potential ability to become a serious problem for the internet community. Thus, anti-spam community offers a wide array of techniques designed to help stop or reduce the huge amount of spam;

however, the amount of spam on the internet is still increasing.

The most common techniques are detecting techniques, which attempt to identify whether a message is a spam based on content and other characteristics of the message (Raed and Adel 2013). One of these techniques is NSA.

The possibility of using NSA in e-mail spam detection is incontestable. When NSA has emerged in the 1990s, it has been first applied in computer security and intrusion detection. Spam is often considered as a type of computer intrusion; thus, using NSA in spam detection automatically fits and has caught the attention of many researchers in this field. Ismaila et al. (2014) presented several techniques using NSA in spam detection. This study focuses on the few recent works that use optimization techniques in their mechanism.

Ismaila et al. (2014) proposed an improved NSA using differential evolution (DE) optimization called NSA–DE. In this method, the DE is implemented at the random generation phase of NSA. Local outlier factor (LOF) is implemented as a fitness function to maximize the distance of generated spam detectors from the non-spam space. The proposed framework was verified with spam dataset (Hopkins et al. 1999), and the results show that the detection accuracy of NSA–DE is better than the classic NSA model.

Another model combines NSA with particle swarm optimization (NSA–PSO; Ismaila et al. 2015; Ismaila and Ali 2014). This model was introduced to improve the random detector generation in the NSA. The combined NSA–PSO uses an LOF as the fitness function for detector generation. The evaluation results on spam base datasets (Hopkins et al. 1999) show that the accuracy of the NSA–PSO model is better than that of traditional NSA model.

Abdolahnezhad and Banirostam (2016) proposed an e-mail detection system based on the modified classic NSA called NSA-II. This model improves the random generation of a detector in NSA using spam and non-spam spaces. In the NSA-II training phase, two sets of detectors are generated, one for spam detectors and other for non-spam detectors. The detectors output from the two sets are used in the testing phase. If one of the spam detectors identified a new pattern, then the e-mail realizes the spam pattern; otherwise, the pattern is considered as a non-spam pattern. The experimental result in spam base dataset shows that the detection performance of NSA-II is higher than the conventional.

3 Constituents of the proposed approach

Hybrid systems have become extensively important in many real-world applications. Given the fact that an individual system has its weakness, a hybrid system is meant to avoid the weaknesses of these single intelligent systems; thus, the importance of a hybrid system is non-negotiable (Ismaila et al. 2014).

The classic NSA has several limitations that decrease its effectiveness in classification applications, especially in spam detection system. Its main drawbacks are: (I) generating a large number of detectors to cover all the non-self spaces; (II) using only non-self detectors in the testing stage and lacking of continuous adaptability; and (III) the existing problem of holes because obtaining a full coverage of self and non-self spaces is difficult.

Nevertheless, in practical situations, normal data are rarely distributed randomly in the entire system space; they are highly concentrated and occupied only a considerably small area of the space system.

In this study, a combination of NSA, and FFO using k-means clustering is realized to accumulate the strong points of the system component and reduce their individual drawbacks. In the proposed method, the self set should be initially clustered using k-means, there by establishing a set of clusters. Subsequently, the clustered self set is regarded as the initial evolutionary population of FFO. Then, the algorithm realizes FFO operations on this population and performs negative selection to obtain non-self detectors of the first level, locating far away from the self in the non-self space delimited by clusters where coverage rate is low. The process is repeated to obtain detectors of the second range, which are located close to the first-level detectors and away from the self region but nearer than the first-level detectors in the non-self space. In RNSA with variable-sized detectors, the detectors located far away from the self usually have a large radii and cover a large area of non-self space.

During the generation of non-self detectors, the self-elements located close to the generated detectors are assigned and saved as boundary self-detectors. At the end of the training stage of CNSA–FFO, three sets are obtained, namely, cluster set, non-self detector set, and self-boundary detector set. These sets are used together in the testing stage to classify whether a new e-mail is a spam. The CNSA–FFO algorithm adopts real detectors with variable size and maximum generated detectors of non-self space as the termination condition.

The stages of the proposed CNSA–FFO are as follows:

1. Definition of the system state space
2. Generation of self (non-self)
   (a) Training stage:
   (b) Generation of the cluster set
3. Generation of the non-self detectors
4. Detection phase

The details of each stage are as follows.

Stage 1: Definition of the system states space

For real-valued representation, self and non-self detectors are a real-valued vectors in n-dimensional space. These detectors can be viewed as a hyper sphere, where n is the number of parameters of the self or non-self detector. The center of this hyper sphere is the real-valued vector. The system state space, denoted by $E=[0,1]^n$, is an n-dimensional space. The normal set is denoted as $N \subset E$, and $AN \subset E$ is the complementary space of normal (abnormal) state, such that:

$$N \cap AN = \Phi, \quad N \cup AN = E,$$

A state of the system (i.e., e-mail messages) can be represented by a vector of features $x = (x_1, x_2,\ldots, x_n) \in E = [0,1]^n$, and each feature is normalized and scaled to [0,1] interval using maximum and minimum method. In (Ismaila et al. 2014 and; Hopkins et al. 1999) there are more details for features vector generation of e-mail messages.

Stage 2: Generation of self

In real applications, obtaining all the normal samples, such as non-spam messages for every e-mail, is impossible. To construct the self space, only a part of the normal states is used to construct the system profile.

The self set is defined as a collection $S \in N$ of elements in space E, where S represents the subset of the normal system state that should be monitored. In the NSAs, a self-sample is composed of two parts, namely, a center s_c and a radius r_s. The center indicates the position of the self sample in E, and the radius determines its size or area covered by the sample. Self set S and self radius r_s should be determined before generating the detectors. In this study, the non-spam space is the normal state of the system, whereas the spam space is the abnormal state of the system.

– Assume the non-spam space to be N ($S \in N$ in RNSA), and each self sample $s = (s_c, r_s)$ has a center $s_c \in [0, 1]^n$ and a self-radius r_s, $r_s \in$ IR;

Notably, the self-radius variance has a direct impact on the classification performance of RNSA.

Stage 3: Training stage

Two techniques are introduced in the training stage, namely, clustering and evolutionary optimization.

(a) Generation of clusters set

In this step, the self-elements are divided into k-clusters using k-means algorithm. In applying this algorithm, the number of k-clusters and the set of self data S should be initially provided. Then, the generating the cluster set is given as follows:

- Let S be a self-set, C be a cluster set, and k be a cluster number.
- Use Algorithm 1 to generate set C. Each element of C has a center $c_c \in E'$ and a radius $r_c \in$ IR. This set will be used in the training and testing stages. E' is a subspace of E that is covered by the generated clusters.

Algorithm 1 Clustering the self

Input S: self-set, K: number of clusters

Output C: clusters set

Initialization $K \leftarrow$ number of clusters, $C \leftarrow \emptyset$

1- Choose a number K of cluster centers c_c (randomly select k elements from S), $s=1,2,..,k$.

2- Assign every element s_i in S to its nearest cluster center using Euclidean distance formula:

$$D(s_i, c_s) = \sqrt{\sum_{j=1}^{m} (s_{ij} - c_{sj})^2}$$

where $i=1, 2,\ldots, n$ and $j=1,2,\ldots,m$ are the number of self-elements and features, respectively.

3- Move each cluster center c_c to the mean of its assigned data points. The new feature of the cluster centroid from its assigned data points is recalculated as

$$C_{kj} = \frac{1}{nc_j} \sum_{i=1}^{nc_j} s_{ij}$$

where C_{kj} is the feature j of cluster k, nc_j is the total number of data vectors in cluster j, and S_{ij} is the feature j of assigned data points i to cluster k.

4- While cluster centers change, repeat Steps 2 to 4 until convergence is achieved. Then, return to C.

(b) Generation of detectors set using FFOA

In NSA, the detector coverage is necessary for obtaining a good classification accuracy, which is challenging to realize perfectly, because the problem of holes is difficult to resolve. In fact, many works have focused on generating the optimal detectors that cover the entire non-self space. Here, k-means clustering and real multidimensional FFO are used to generate detectors one after the other only in a part of space search. This architecture aims to obtain the best coverage with a small number of detectors. The steps of this process are given as follows:

1. Let E' be a subspace of E.
2. Let S be a self-set; each self-sample $s = (s_c, r_s)$ has a center $s_c \in E'$ and a self-radius r_s, $r_s \in$ IR.
3. Let BS be a boundary of self-detector set; BS is a subset of S that contains all the self-elements situated in the boundary of self, $BS \leftarrow \emptyset$.
4. Let C be a cluster set generated in stage 3-a; each cluster $c = (c_c, r_c)$ has a center $c_c \in E'$ and a cluster radius r_c, $r_c \in$ IR.
5. Let D be a set of generated detectors, $D \leftarrow \emptyset$.
6. Generate a random element $x \in S$ from the self-set; x is considered as the initial position of the fruit fly swarm.

7. Use Algorithm 2 to obtain the best detector, with *S*, *C*, *x*, and *D* as input parameters and d_{best} as the output parameter; d_{best} is the best solution (optimal detector).
8. If $d_{best} \neq \emptyset$, add a new detector to detector set $D \leftarrow D \cup \{\langle d_c, r_d \rangle\}$, $d_c \in E'$ is the center of d_{best} and r_d is its radius. Then go to Step 9. Otherwise, go back to Step 6.
9. Calculate *sb*, which is the nearest self-element to d_{best}; add it to the boundary self-set $BS \leftarrow BS \cup sb$.
10. If the number of detectors has not reached the maximum, then go to Step 6; otherwise, stop and return to D.

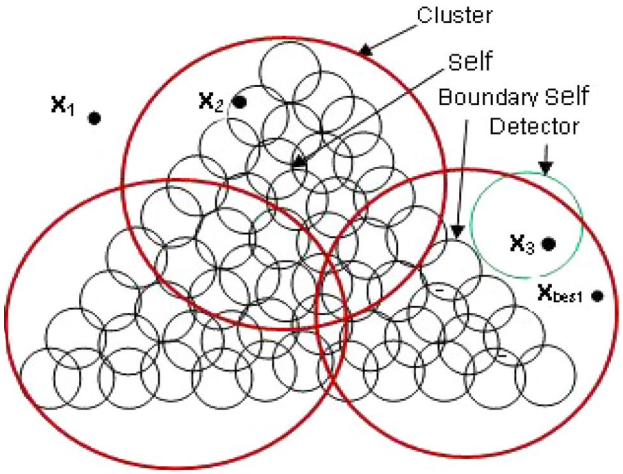

Fig. 1 Illustration of detectors generation in CNSA-FFO

Algorithm 2 FFOA

Input *C*: cluster set; *S*: self set; *D*: detector set;
x: sample
Output d_{best}: best detector
Initialization $K \leftarrow$ FFO number; $X \leftarrow \emptyset$, is the set of flies

1. Initialize the positions of the fruit fly swarm

$$X\_init \leftarrow x$$

2. Calculate the locations of the fruit flies
for each i **in** k **do**

$$X_i \leftarrow X_i\_init + randomvalue;\ X \leftarrow X \cup \{X_i\}$$

3. Find and return the best fly X_i in *X* using Algorithm 3, as presented as follows:

$$d_{\text{best}} \leftarrow X_{\text{best}}$$

4. Return d_{best}

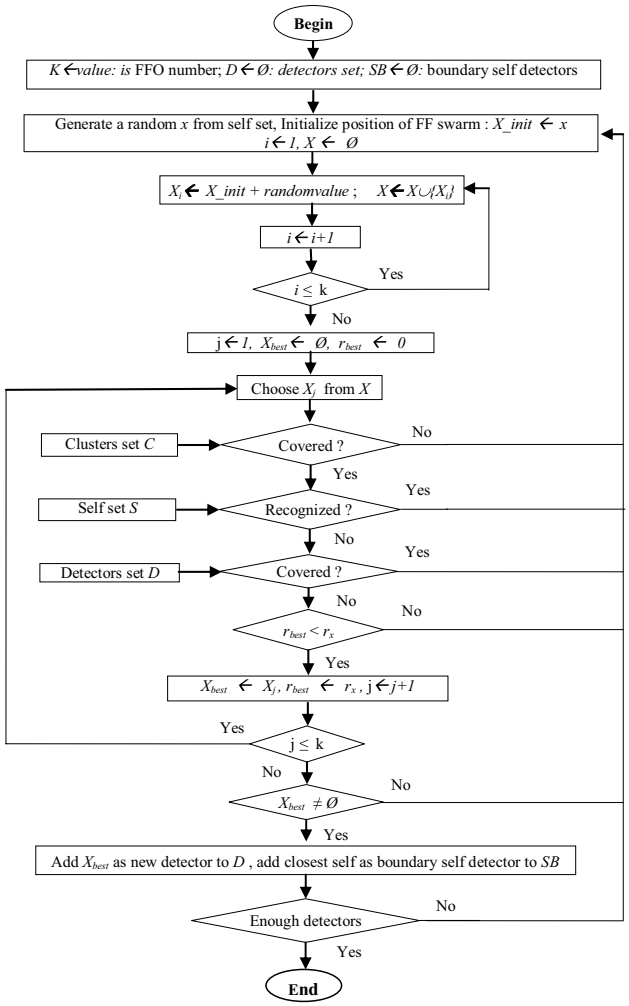

Fig. 2 Detector generation mechanism of CNSA-FFO

- Fitness function in CNSA-FFO

In this study, a multi-objective fitness function is used to obtain the optimal detector in the population set generated by the mutation operation of FFA. A detector *X* is the best (optimal) if it is covered by clusters, unrecognizable by self, not covered by the generated detectors, and the farthest from the self. The fitness algorithm is presented as follows.

Algorithm 3 Fitness function of CNSA–FFO

Input C: cluster set; S: self-set; D: detector set ;X: fruit fly set

Output X_{best}: best detector

Initialization K←FFO number, X_{best}←$\emptyset$, r_{best}←0

1. Select an element X_i from a set X, i=1,2,..., k.
2. If X_i (candidate detector) is covered by any cluster in C, that is,

$$D(x, c) = \sqrt{\sum_{i=1}^{n}(x_i - c_i)^2} \le r_c$$

where $D(x, c)$ is the Euclidean distance formula. Then, proceed to Step 3; otherwise, eliminate it (case X_1 in Figure 1) and go to Step 6.

3. If X_i is recognized by any self-element s_i, that is,

$$D(x, s) = \sqrt{\sum_{i=1}^{n}(x_i - s_i)^2} \le r_s$$

then eliminate it (case X_2 in Figure 1) and go to Step 6; otherwise, proceed to Step 4.

4. If X_i is recognized by any accepted detector, that is

$$D(x, d) = \sqrt{\sum_{i=1}^{n}(x_i - d_i)^2} \le r_d$$

then eliminate it (case X_3 in Figure 1) and go to Step 6; otherwise, proceed to Step 5.

5. If the radius of X_i is larger than r_{best}, then replace X_{best} by X_i and go to Step6; otherwise, go back to Step1.
6. If $X_i \neq \emptyset$, then go back to Step 1; otherwise proceed to Step 7.

Return X_{best}

Stage 4: Detection phase

In the detection stage of CNSA–FFO, the cluster set (C), the boundary self-detectors (SB), and non-self detectors (D) are used to check whether testing sample t is normal (non-spam). The steps of this process are as follow (Figs. 1, 2):

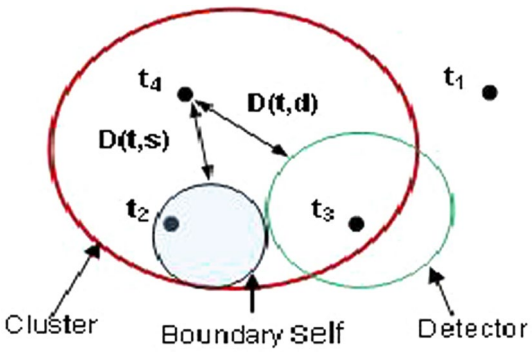

Fig. 3 Illustration of testing mechanism in CNSA-FFO

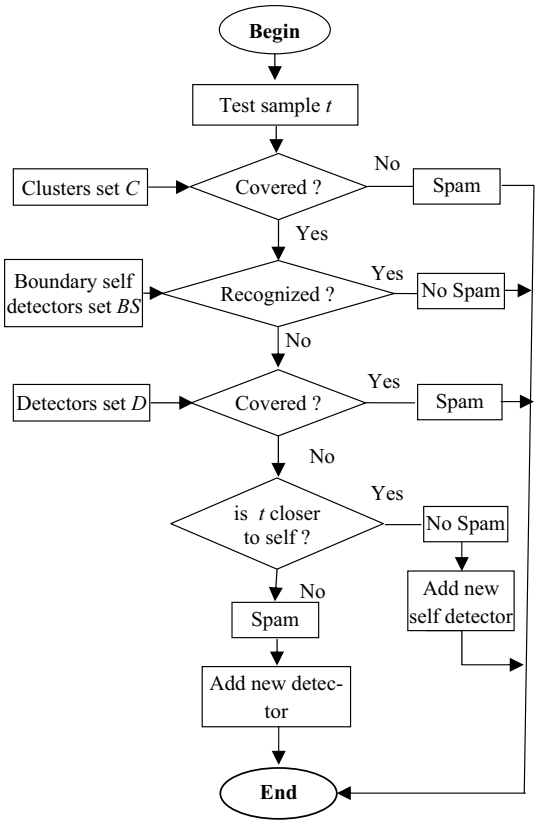

Fig. 4 Detection mechanism of CNSA-FFO

1. Let t be the testing sample (e-mail message) with the intention of determining whether it is a spam.
2. Use cluster set C, non-self detector set D, and boundary self-set BS, which have been obtained in the previous stages.
3. If sample t is not covered by any cluster (t_1 in Fig. 3), that is,

$$D(t, c) = \sqrt{\sum_{i=1}^{n}(t_i - c_i)^2} > r_c,$$

then classify t as abnormal (spam) and go to Step 7; otherwise proceed to Step 4.

4. If t is recognized by any self s in BS (t_2 in Fig. 3), that is,

$$D(t, s) = \sqrt{\sum_{i=1}^{n}(t_i - s_i)^2} \leqslant r_s,$$

then consider t as normal (non-spam) and go to Step 7; otherwise, proceed to Step 5.

5. If t is recognized by any detector d in D (Fig. 3, t_3), that is,

$$D(t, d) = \sqrt{\sum_{i=1}^{n}(t_i - d_i)^2} \leqslant r_d,$$

then classify t as abnormal (spam) and go to Step 7; otherwise, proceed to Step 6.

6. If $D(t, d) \leq D(t, s)$, then t is classified as abnormal (spam), and a new detector $d_{new} = < t, D(t, s) - r_s >$ is added to D. Otherwise, t is classified as normal (non-spam), and a new boundary self-element s_{new} should be added to BS:

$$S_{new} = \begin{cases} \langle t, r_s \rangle, \text{ if } r_s \leqslant D(t, d) - r_d \\ \langle t, D(t, d) - r_d \rangle, \text{ if } r_s \geqslant D(t, d) - r_d \end{cases}.$$

Then, go to Step7.

7. Return the t class (spam or non-spam).

The proposed CNSA–FFO model has specific characteristics compared with V-detectors and other models in the literature. (1) The training and testing stages are performed only in a small part of the system space. (2) Three types of detector are generated and used to check the test sample. (3) A system is adopted by a mechanism to eliminate the holes based on assigning the uncovered test sample to the nearest detectors. (4) The system can continuously update its profile.

4 Empirical study

The performance of the proposed model has been verified and evaluated using actual spam-based data set and has been compared with V-detectors (Zhou and Dasgupta 2009) and NSA–PSO (Ismaila et al. 2015).

4.1 Benchmark data analysis

The data used in this study were obtained from spam-based dataset of e-mail messages. This benchmark contains 4601 messages, in which 1813 (39%) of the messages are marked as spam, whereas 2788 (61%) are labeled as non-spam.

The proposed improved model was evaluated by dividing the dataset using a stratified sample approach with 70% training set and 30% testing set to investigate the performance of the new model on unseen data (Ismaila et al. 2015). More details about this benchmark are discussed by (Hopkins et al. 1999). The Table 1 gives a detail of training and testing samples of CNSA-FFO.

Table 1 Training and testing samples of CNSA-FFO

Training samples		Testing samples	
Spam	Non spam	Spam	Non spam
0	1952	1813	836

4.2 Criteria for performance evaluation

To evaluate the accuracy and performance of the proposed model for e-mail spam detection and compare it with RNSA (Zhou and Dasgupta 2009) and NSA–PSO (Ismaila et al. 2015), the same measures used by Ismaila et al. (2015) were utilized. The measures employed are sensitivity (SN), specificity (SP), positive prediction value (PPV), accuracy (ACC), negative prediction value (NPV), correlation coefficient (CC), and f-measure (F1). See Ismaila et al. (2014), Ismaila and Ali (2014) and Ismaila et al. (2015) for more detailed mathematical formulas. Moreover, the number of generated detectors, adaptability, and complexity are often used as measures in evaluating the NSA-based methods.

4.3 Experimental settings and implementation

The process of implementation did not use any ready-made code, and all required functions are coded using the same platform. The evaluation of the proposed hybrid model is implemented by dividing the data set using a stratified sample approach with 70% training set and 30% testing set to verify the performance of the new model on new data (Ismaila et al. 2015). The proposed model is implemented with a threshold value (self-radius) of 0.4, whereas the number of generated detectors is between 100 and 3500. In NSA-based models, the threshold value and number of generated detectors have a great impact on the final output measure. In the training stage, only the normal data (non-spam samples) are used in constructing the proposed model. The number of clusters is fixed at 30, and the size of fruit fly swarm is at 50. The performance of the model with both data of testing set (spam and non-spam messages) is evaluated.

5 Results and discussion

The proposed algorithm compares the classic NSA, NSA–PSO, and CNSA–FFO models. These models were evaluated using statistical measures to determine the best model for e-mail spam detection.

The testing results consist of the 2000 generated detectors and threshold value of 0.4 give summary and comparison of results in percentage for CNSA-FFO, NSA and NSA-PSO models in Table 2.

This table shows the values of the accuracy, correlation coefficient, F-measure, sensitivity, positive prediction value, specificity and negative prediction value, for NSA, NSA–PSO, and the proposed model. The results indicate that the proposed NSA can achieve a higher performance than that of the other two methods, specifically in accuracy, positive and negative prediction values, and correlation coefficient criteria (Fig. 4).

Table 2 Testing results of NSA, NSA–PSO, and CNSA–FFO at 2000 generated detectors

Model	NSA	NSA-PSO	CNSA-FFO
Measure			
ACC	68.86	91.22	93.88
CC	48.33	63.37	86.29
F1	36.01	74.95	45.34
SN	22.24	65.99	87.28
PPV	94.53	86.72	94.38
SP	99.16	93.43	97.31
NPV	66.24	80.86	93.66

In the NSA-based models, the most important measures that are usually used to calculate and compare the performance are accuracy and positive prediction value. Table 3 and Figs. 5 and 6 show only the best accuracy and positive prediction value of the three models.

For detection accuracy, the accuracy of the proposed model (93.88%) at only 3500 generated detectors is better than the classic NSA (68.86%) and NSA–PSO (82.62%) at 5000 generated detectors.

For positive prediction value, the CNSA–FFO is at 94.38%, whereas NSA–PSO is at 91.22%. Thus, CNSA–FFO is better than NSA–PSO in positive prediction value and close to that of classic NSA (94.53%).

From the obtained results and analysis, the CNSA– FFO model performs better than other existing detection models in many aspects. Therefore, the proposed spam detection mechanism was constructed based on the CNSA–FFO model. This model can be considered as a powerful tool and framework in detecting e-mail spam due to its architecture and adaptive nature.

Table 3 Best accuracy and positive prediction value of NSA, NSA–PSO at 5000, and CNSA–FFO at 3500 generated detectors

	NSA	NSA-PSO	CNSA-FFO
ACC	68.86	91.22	93.88
PPV	94.53	86.72	94.38

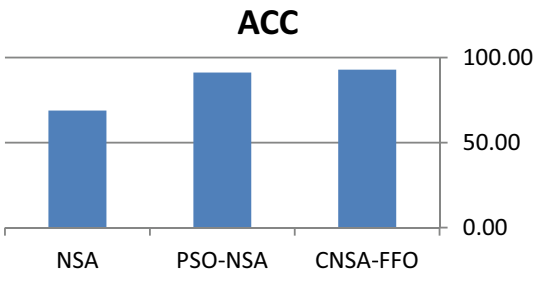

Fig. 5 Accuracy comparison of NSA, NSA-PSO and CNSA-FFO

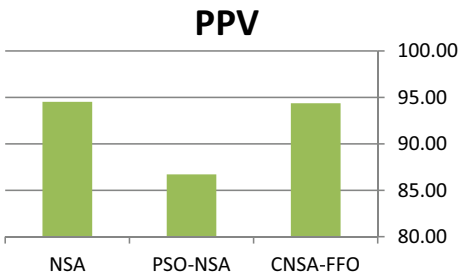

Fig. 6 Comparison of the positive prediction value of NSA, NSA–PSO and CNSA–FFO

6 Conclusion

In this study, a new and improved NSA has been proposed and implemented. The proposed model, CNSA–FFO, is applied in classifying whether e-mail messages are spam. CNSA–FFO is a hybrid method that combines NSA with k-means clustering and FFO. The goal is to enhance the performance of previously proposed solutions based on NSA.

The performance and accuracy test in the actual spam-based dataset has shown that the CNSA–FFO method can detect e-mail spam better than the conventional NSA method and other models. Also, the proposed model ensures continuous adaptability and significantly reduces the number of generated detectors.

References

Abdolahnezhad MR, Banirostam T (2016) Improved negative selection algorithm for email spam detection application. Int J Adv Res Electron Commun Eng 5:956–960

Aiqiang X, Yong L, Xiuli Z, chunying Y, Tingjun L (2011) Optimization and application of real-valued negative selection algorithm. Procedia Eng 23:241–246

Bo X, Wen-Jing G (2014) Fruit fly optimization algorithm. In: Innovative computational intelligence: a rough guide to 134 clever algorithms, Part II, vol 62. Springer, Berlin, pp 167–170

Chen W, Li T, Liu XJ, Zhang B (2013) A negative selection algorithm based on hierarchical clustering of self set. In: Information sciences, vol 56. Springer and Science, China, pp 1–13

Dasgupta D, Yu S, Nino F (2011) Recent advances in artificial immune systems. Soft Comput 11:1574–1587

Dong L, Shulin L, Hongli Z (2016) A boundary-fixed negative selection algorithm with online adaptive learning under small samples for anomaly detection. In: Engineering Applications of Artificial Intelligence, vol 50. Elsevier, pp 93–105

De Castro LN, Timmis JI (2003) Artificial immune systems as a novel soft computing paradigm. Soft Comput 7:526–544

Forrest S, Perelson A, Allen L, Cherukuri R (1994) Self-nonself discrimination in a computer. In: PW IEEE Symp on Research in Security and Privacy, IEEE

Gang C, Wenjian L, ao Z (1979) Evolutionary clustering with differential evolution. IEEE congress on evolutionary computation (CEC), IEEE, vol 28, 1, pp 100–108

Gonzalez F, Dasgupta D, Kozma R (2002) Combiningnegativeselectionandclassificationtechniquesforanomalydetection. In: Proceedings of the congress on evolutionary computation (CEC-2002), IEEE, 0-7803-7282-4

Gonzalez F, Dasgupta D, Niño LF (2003) A randomized real-valued negative selection algorithm. In: ICARIS 2003, Computer Science, vol 2787. Springer, pp 261–272

Guiyang L, Tao L, Jie Z, Haibo l (2010) An outlier robust negative selection algorithm inspired by immune suppression. J Comput 5(9)

Han JY, Liu CZ (2013) Fruit fly optimization algorithm with adaptive mutation. Appl Res Comput 30:1–6 **(in Chinese)**

Hartigan JA, Wong MA (1979) A k-means clustering algorithm. J R Stat Soc Ser C (Appl Statis)

Hazim I, Mesut G (2015) A survey on fruit fly optimization algorithm. In: 11th international conference on signal-image technology and internet-based systems, IEEE computer society

Hopkins M et al (1999) UCI machine learning repository: spam base data set. Hewlett-Packard Labs. https://archive.ics.uci.edu/ml/datasets/Spambase

Ismaila I, Ali S (2014) Improved email spam detection model with negative selection algorithm and particle swarm optimization. Appl Soft Comput 22:11–27

Ismaila I, Ali S, Sigeru O (2014) Hybrid email spam detection model with negative selection algorithm and differential evolution. Eng Appl Artif Intell 28:97–110

Ismaila I, Ali S, Ngoc TN, Sigeru O, Ondrej K, Kamil K, Marek P (2015) A combined negative selection algorithm–particle swarm optimization for an email spam detection system. Eng Appl Artif Intell J 39:33–44

Jie Z, Wenjian L (2014) EvoSeedRNSAII: an improved evolutionary algorithm for generating detectors in the real-valued negative selection algorithms. Appl Soft Comput 19:18–30

Jie Z, Wenjian L, Baoliang X (2009) Generating an approximately optimal detector set by evolving random seeds. In: The eighth IEEE international conference on dependable, autonomic and secure computing, Chengdu, China, IEEE, 978-0-7695-3929-4409

Jinquan Z, Xiaojie L, Tao L, Caiming L, Lingxi P, Feixian S (2009) A self-adaptive negative selection algorithm used for anomaly detection. Prog Nat Sci 19:261–266

Kim JW (2002) Integrating artificial immune algorithms for intrusion detection. PhD Thesis, University College London

Li D, Liu S, Zhang H (2015) A negative selection algorithm with online adaptive learning under small samples for anomaly detection. Neuro Comput J 149:515–525

Li D, Liu S, Zhang H (2016) A boundary-fixed negative selection algorithm with online adaptive learning under small samples for anomaly detection. Eng Appl Artif Intell 50:93–105

Lin C, Dechang P, Chuanming C (2015) BIORV-NSA: bidirectional inhibition optimization r-variable negative selection algorithm and its application. Appl Soft Comput 32:544–552

Liu Y, Wang X, Li Y (2012) A modified fruit-fly optimization algorithm aided PID controller designing. IEEE 10th world congress on intelligent control and automation, Beijing, China

Maoguo G, Jian Z, Jingjing M, Licheng J (2012) An efficient negative selection algorithm with further training for anomaly detection. Knowl Based Syst 30:185–191

Pan WT (2011) A new fruit fly optimization algorithm: taking the financial distress model as an example. Knowl Based Syst ACM 26:69–74

Puteh M, Hamdan AR, Omar K, Bakar A (2008) Flexible immune network recognition system for mining heterogeneous data. In: 7th international conference on artificial immune systems, Phuket, Thailand, Springer

Raed AZ, Adel H (2013) Genetic optimized artificial immune system in spam detection: a review and a model. Artif Intell Rev 40:305–377

Ramdane C, Chikhi S (2014) A new negative selection algorithm for adaptive network intrusion detection system. Int J Inf Secur Priv 8(4):1–25

Shelly XW, Wolfgang B (2010) The use of computational intelligence in intrusion detection systems: a review. Appl Soft Comput 10:1–35

Vieira LN, Lima BSLPD, Jacop BP (2008) Optimization of steel catenary risers for offshore oil production using artificial immune system. In: 7th international conference on artificial immune systems (ICARIS 2008), Phuket, Thailand, Springer

Wang B, Zhang S (2007) A new intrusion detection method based on artificial immune system. In: Network and parallel computing workshops, IEEE, pp 91–98

Wang L, Zheng X-L, Wang S-Y (2013) A novel binary fruit fly optimization algorithm for solving the multidimensional knapsack problem. Knowl Based Syst 48:17–23

Wen C, Xiaoming D, Tao L, Tao Y (2014) Negative selection algorithm based on grid file of the feature space. Knowl Based Syst 56:26–35

Wu P, Zheng X (2012) An improved variable-radius real-valued negative selection algorithm. J Inf Comput Sci 16:4713–4720

Xiao X, Li T, Zhang R (2015) An immune optimization based real-valued negative selection algorithm. Appl Intell J 42:289–302

Zhenga X, Zhoua Y, Fangb Y (2013) Dual negative selection algorithm based on pattern recognition receptor theory and its application in two-class data classification. J Comput 8:1951–1959

Zhou J, Dasgupta D (2009) V-detector: an efficient negative selection algorithm with"probably adequate" detector coverage. Inf Sci 179:1390–1406