Efficient and Secure Intrusion Detection System Based on Feature Subset Selection with Optimized Machine Learning for Wireless Sensor Network

P. Nirmaladevi*, and A. Tamilarasi**

ABSTRACT

Network Intrusions are grave issues in computer and network systems. Several intrusion detection approaches are present to determine these severe problems but the major problem is performance. To increase performance, it is significant to rise the detection rates and reduce false alarm rates in the area of intrusion detection. The recent approaches use Principal Component Analysis (PCA) towards project features space to principal feature space and select features corresponding to the highest eigenvalue, but the features corresponding to the highest eigenvalue may not have the optimal sensitivity for the classifier due to snubbing many sensitive features. Instead of using traditional approach of selecting features with the highest eigenvalue such as PCA, proposed method applied a Bacteria Foraging (BF) to search the principal feature space for Bacteria eigenvectors that offers a subset of features with optimal sensitivity and the highest discriminatory power. To improve the network security and easy detection of malicious attacks at the classification stage this paper proposed an effective method that uses the modified artificial bee colony (mABC) to optimize the Kernel Extreme Learning Machine (KELM). With proper hidden layer neuron number, KELM could enhance the accuracy and speed of the intrusion detection. To verify the proposed technique, experimental tests have been implemented in this work. The test result demonstrates that the proposed mABC-KELM can detect the network intrusion efficiently and its performance is superior compared to existing methods.

Keywords: Intelligent Detection, Network Intrusion, Artificial Neural Network, Principal Component Analysis, Bacteria Foraging, Kernel Extreme Learning Machine.

1. INTRODUCTION

Wireless Sensor Network (WSN) is a kind of network that have many (from dozens to thousands) minute devices sensing and collecting detailed information about the physical environment. Due to WSNs economical cost and simple propagation characteristics, they are used for many different fields of science, health, military, security to sense and gather data respecting various activities, for instance exploring the battlefield (e.g.-Boomerang Sniper Identifying System), monitoring highway traffic, identifying NBC (Nuclear, Biological, Chemical) attacks, fire alarm system, learning wildlife and oceans (Great Duck Island-GDI Project), home automation systems, agriculture, transportation, space exploration and many others. With the enormous growth of WSN-based computer services and the huge increase in the number of requests running on networked systems, the adoption of appropriate security measures to protect against computer and network intrusions is aessential issue in a computing environment. Intrusions into or attacks on a computer or network system are actions or attempts to destabilize it by compromising security in confidentiality, availability or integrity of the system. As defined in [1], an Intrusion Detection System (IDS) monitors events occurring in a network and examines them for signs of intrusions.
The current internet-based information processing systems are prone to different types of threats which lead to various types of damages resulting in significant losses. Therefore, the significance of information security is evolving quickly. The most basic goal of information security is to improve defensive information systems which are protected from unauthorized access, use, disclosure, disruption, modification, or destruction. Furthermore, information security minimizes the risks related to the three main security goals namely, integrity, confidentiality and availability. Many systems have been designed in the past to identify and block the Internet-based attacks.

The furthermost important systems among them are intrusion detection systems (IDS) since they resist external attacks effectively. Furthermore, IDSs provide a wall of defense which overcomes the attack of computer systems on the Internet. IDS could be used to identify different types of attacks on network communications and computer system usage where the traditional firewall cannot execute well. Intrusion detection is based on an assumption that the behavior of intruders differ from a legal user [2]. Generally, IDSs are broadly classified into two categories namely anomaly and abuse detection systems based on their detection approaches [3]. Anomaly intrusion detection determines whether deviation from the established normal usage patterns can be labelled as intrusions.

On the other hand, misuse detection systems detect the violations of permissions effectively. Intrusion detection systems can be built by using intelligent agents and classification methods. Most IDSs work in two phases namely preprocessing phase and intrusion detection phase. The intrusions identified by the IDSs can be prevented effectively by evolving an intrusion prevention system. IDS systems can be divided into two methods: abuse detection and anomaly detection [4]. Misuse detection can detect the attacks based on famous susceptibilities and patterns of intrusions (attacks signatures) stored in a database. It matches the current conduct against the previous knowledge of those known attack patterns [5]. Therefore, this technique may not able to aware the system administrator in case of a new attack. Conversely, Anomaly detection creates a normal conduct profile and detects the intrusions based on important deviations from this normal profile [6]. Thus, anomaly detection methods can detect new types of attack, but it suffers from a high rate of false alarms to train in highly dynamic surroundings. Many challenges need to be considering when building an IDS, such as data collection, classification accuracy and data preprocessing. Classification is the prediction of the category labels of instances that are classically described through a set of features (attributes) in a dataset.

Numerous classification techniques have been proposed for the improvement of IDS; with Fuzzy Logic (FL) [7], Neural Networks (NN) [8], Support Vector Machines (SVM) [5, 6] and Decision Tree (DT) [9]. Another important problem for constructing IDS is dealing with data comprising large number of features. Data in high dimensional space may lead to reduction the classification accuracy of the IDS. Therefore, feature selection is mandatory as a pre-processing phase for high dimensional data before solving the classification problems. Feature selection aims to decrease the number of irrelevant and redundant features.

An anomaly network intrusion detection system is proposed in this paper using Bacteria Foraging (BF) based feature selection method and the modified artificial bee colony (mABC) to optimize the Kernel Extreme Learning Machine (KELM). The effectiveness of the proposed network IDS is assessed by conducting numerous experiments on NSL-KDD network intrusion dataset. The results show that the proposed mABC-KELM rises the accurateness and speeds up the detection time. The rest of this paper is prepared as follows: Section 2 presents a background of the used approaches; Section 3 describes feature selection method and the proposed mABC-KELM IDS system. Section 4 gives the execution results and analysis. Finally, Section 5 contains the conclusion remarks.

2. RELATED WORK
Security threats take place in wireless sensor network are different from wired network threats because of structure of WSN and constrictions which it has such as limited battery life. Hence IDS implemented in
WSN has different approaches [10]. In this section, it is described that approaches pointed out by some important studies achieved in recent years. All classifications of detection approaches made by different researcher occur from public IDS taxonomy (Misuse Detection, Anomaly Detection). Due to different features of WSNs from wired and non-energy constrained wireless networks, different classification types is pointed out in this section.

In [11], classifying is made as intrusion type, intruder type, detection methods, source of the collected data, analyzing location of the collected data, usage frequency and this categorizing is the most comprehensive in the literature. In a network, intruder type is grouped into two categories. These categories are internal intruder (selfish or malicious node) and external intruder (An outside attacker trying to reach the system). In WSN, consistent with intrusion type, intrusion can be by stealing the data, by creating false data and so altering the system, by rejecting to access the system, by influencing the energy efficient. For detection approaches it has been described above as misuse and anomaly detection but additionally some papers point out hybrid or specification based detection.

In A Game-Theoretic Framework for Robust Optimal Intrusion Detection in Wireless Sensor Networks-2014, it is demanded that instead of approaches using heuristic and adhoc solutions, there is an increase to use analytical methods for security issues in WSN. Hence authors propose a nonzerosum discounted robust stochastic game framework to analyse intrusion detection difficult in WSN. Game’s parameters are modelled by features of WSN and it’s environment [12].

In Anomaly Detection and Localization in UWB Wireless Sensor Networks-2013, author has been proposed an anomaly detection solution specifically designed for the ultrawideband (UWB) technology. In the paper, it is described that UWB is a key answer to serve low power consumption while wireless connectivity. To identify intrusions, a rule based approach is accepted and performance of the proposed algorithm is studied by simulations. The algorithm projected in the paper, uses a round-based (There are particular phases) approach towards cluster structure and rule based anomaly detection. The test outcomes shown in paper point out a successful detection accuracy [13].

In Applying Data Mining Methods to Intrusion Detection in Wireless Sensor Networks-2013, it is proposed that the application using data mining approaches for intrusion detection system in wireless sensor network and proposed system can perform both irregularity detection method and misuse detection method. The IDS consists of a Central Agent and several Local Agents, which are located on the sensors and carry out intrusion detection activities. Data mining approach is used on everyagents (Local Agents, Central Agents). The test outcomes show that high detection accuracy is obtained while keeping an acceptable, however not negligible false positives rate [14].

In Intrusion Detection in WSN Using Watchdog Based Clonal Selection Algorithm [15], the watchdog approach is used to detect whether a node has abnormal behaviour while transmitting data. All nodes in the WSN isanswerable for monitoring the neighbours and transferring the info about behaviour. Misbehaviour of nodes affect act of WSN negatively. With using watchdog based method.

The execution of genetic algorithms on top of information theory to enrich intrusion detection has been proposed by Xiao, et. al. [16]. Genetic algorithms have been used for classification of Smurf attack labels in training data set, achieving a false positive rate as low as 0.2% by Goyal and Kumar [17]. Abdullah, et. al. [18] used genetic algorithms for procurementclassification rules for intrusion detection. Ojugo, et. al. [19], have used genetic algorithms to improve rule-based intrusion detection. The suitability function has been used to evaluate the rules. In one existing research work by Liu et al [20] and his colleagues, PCA is practical for classification and neural networks are meant for online computing. They selected 22 principal modules as features subset selection to obtain the best presentation. But there is a possibility to miss many vital principal modules having sensitive information for intrusion detection during selection phase.
P. Nirmaladevi, and A. Tamilarasi

Horng et al [21] and his co-workers observed the important features based on the accuracy and the number of false positives of the system considered with and without the feature. In additional words, the feature selection of is “leave-one-out”, remove one feature from the original dataset, redo the test, then relate the new results with the original result, if any case of the described cases happens. The feature is regarded as significant; else it is regarded as insignificant. Since there are 41 features advised in the KDD-cup99, the test is repeated 41 times to certify that each feature is either essential or inconsequential. This process involved obstacle as well as overheads on massive dataset. One of the most important works is done by Tong et al [22] and his contacts in which they employed the Radial Basis Function (RBF) network is a real-time pattern classification and the Elman network is applied to restore the memory of earlier events. They used full featured KDD-cup dataset. This rises training and testing overheads on the system.

3. PROPOSED MODEL

The proposed model contain different parts; dataset used for experiments, feature transformation and organization, classification architectures, optimal feature subset selection, implementation, training and testing, and results comparison. The proposed method communicate with cluster based medium and watchdog mechanism is monitored and updated the neighbor nodes information’s in each node information table. The block diagram of proposed model is shown in the Figure 1.

3.1. Dataset used for Experiments

The proposed system used ADFA dataset for tests. The selection of this dataset is due to its standardization, content richness and it helps to evaluate proposed results with existing methods of intrusion detection system. ADFA is developed using a modern operating system and contemporary attacks, for further estimation of the semantic algorithm. This new dataset is available for public use without restriction, and can be collected from j.hu@adfa.edu.au.ink. The dataset contains 833 normal traces for training the IDS, 4373 normal traces for valuing FAR and 60 different attack sets, each consisting of multiple traces.

3.2. Dataset pre-processing

First, the data are pre-processed, and then it is given to the selected classifiers. The raw dataset is pre-processed for removing symbolic values and feature transformation using PCA. Finally, optimal features subset selection using BF.

![Figure 1: overall process of proposed system](image-url)
3.2.1. Feature transformation and organization

In pre-processing second stage, PCA is applied for data reduction, but in this work, PCA used for feature transformation into principal components feature space and it is organized in descending order. PCA purposely is to reduce the dimension of the data while retaining as much as probable of the variation present in the original dataset. It provides a way of identifying patterns in data and state the data in such a way as to highpoint their similarities and differences [23]. However, PCA here was used to transform the input vectors to the new search space. On the other hand, the choosing of number of principal components is done by BF. PCA Algorithm is represented as follows:

**PCA Algorithm**

Input: \( x = (x_1, x_2, ..., x_M) \) M – Maximum number of features

1. Find mean \( \bar{x} = \frac{1}{M} \sum_{i=1}^{M} x_i \)

2. Calculate deviation \( \Phi_i = (x_i - \bar{x}) \)

3. Find Covariance matrix \( A \) // \( A \) is \( (N \times M) \) matrix \( \rightarrow \Phi_i (i = 1 ... M) // \) through \( C = \frac{1}{M} \sum_{i=1}^{M} \Phi_i \Phi_N = AA^T \)

4. Compute eigenvalue \( \lambda \) as \( \lambda_1 > \lambda_2 \ldots \lambda_N \)

5. Compute eigenvector \( \mu_i \) as \( \lambda_1 > \lambda_2 \ldots \lambda_N // C \) is symmetric form a basis, (i.e. any vector \( x \) or \( x_i - \bar{x} \) actually, can be mentioned as a linear combination of the eigenvectors): \( x_i - \bar{x} = b_1\mu_1 + b_2\mu_2 \ldots b_N\mu_N = \sum_{i=1}^{N} b_i\mu_i \)

6. Arrange \( (\lambda_i, \mu_i) \) // descending order // \( pc_1 > pc_2 \ldots > pc_l \) // \( l \) is the transformed number of features //

3.2.2. Feature Subset Selection

In third step of pre-processing, Bacteria Foraging (BF) applied for optimal features subset selection from principal components search space. This is a main contribution that positively impact on the act of intrusion detection analysis engine. BF is inspired by the biological mechanisms of reproduction. In BF, E.coli bacteria searching principles are: swarming, chemotaxiselimination dispersal and reproduction.

Chemotaxis-for the duration of chemotaxis in a nutrient location, an E.coli bacterium \( \theta \) tumble an unit step, \( C \) (i) in a random direction, given by a random vector \( \Delta(i) \in \mathbb{R}^2 \) whose each component is among \([-1, 1]\). This tumbling behavior is specified by (3). If this is found to be favorable then \( \theta \) swims for a period of time (swim length, \( N_s \)) in that direction. The number of chemotactic steps is described by \( N_c \).

\[
\theta_i(j+1, k, l) = \theta_i(j, k, l) + c(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}
\]

(1)

Swarming: When an E.coli cell moves up the nutrient gradient, it releases an attractant. Because of this, some cells of E.coli form steadyspatio-temporal patterns of concentric rings (swarms). The cell-to-cell signaling \( J_{cc} \) by a bacteria \( \theta \) to the total population \( P \) is specified by (4). This is added to the fitness function \( J(\theta) \) (i.e. optimal feature selection) to provide a time-varying fitness.

\[
J_{cc}(\theta, P(j,k,l))
\]

(2)
\[ J(J_j, J_k, J_l) = \sum_{i=1}^{NP} J_{cc}\left(\theta_i, \theta_j, (j, k, l)\right) \]

\[ = \sum_{i=1}^{NP} \left[ -d_{\text{attr}} \exp \left(-w_{\text{attr}} \sum_{m=1}^{2D} (\theta_j^m - \theta_i^m)^2 \right) \right] \]

\[ + \sum_{i=1}^{NP} \left[ h_{\text{repellent}} \exp \left(-w_{\text{repellent}} \sum_{m=1}^{2D} (\theta_j^m - \theta_i^m)^2 \right) \right] \]

Reproduction: From the total population, the smallest healthy half dies (deleted) and the healthiest half asexually splitting into two (copied) to possess the population size constant. It is to be noted, higher value of \( J \) here described as lower health of the bacteria \( \theta \). This iterates for \( N_r \) steps. After every single reproduction phase, chemotaxis and swarming repeats.

Elimination-dispersal: Sometimes, sudden change in atmosphere kills a few bacteria and to balance this nature diffuses some bacteria at a new place. For simulation, a bacteria is abolished with probability \( p \). If a bacterium is abolished, another bacteria is dispersed randomly at any place on the optimization domain. This continues for \( N_e \) steps where each step is followed by chemotaxis, swarming and reproduction. The main aim of feature subset selection is to use less features to attain the same or better performance.

Therefore, the fitness valuation contains two terms: (i) accuracy and (ii) the number of selected features. The features are defined the traces in dataset.

3.3. Classification Architectures

3.3.1. Modified Artificial Bee Colony algorithm

In a natural bee colony, several tasks are executed by specialized entities. These particular bees attempt to maximize the nectar amount stored in the hive using effectual division of labour and self-organization. The Artificial Bee Colony (ABC) algorithm, projected by Karaboga in 2005 for real-parameter optimization it is simulates the foraging behavior of a bee colony [24]. The forage selection in a honey bee colony, which the ABC algorithm consists of three kinds of bees such as employed bees, onlooker bees and scout bees. The colony contain half of employed bees and other half of onlooker bees. Employed bees are answerable for exploiting the nectar sources discovered earlier and giving information to the onlooker bees in the hive concerning the quality of the food source sites which they are exploiting.

In behaviour of bees, Onlooker bees wait in the hive and decide on a food source to exploit dependson the information shared by the employed bees. Either Scout randomly searches the location with the aim of find a new food source depending on internal motivation or possible external clues [25]. The intelligent behaviour of foraging in bees is reviewed by given below

1. In foraging process, initially, the bees start to explore the atmosphere randomly to facilitate find a food source of bees.

2. If the bees are finding a food source means they will becomes an employed forager and begins to develop the discovered source. Then the employed bees are collected the nectar and returns to the hive and unload the nectar.

Following that unloading the nectar, she can go back to her discovered source site straight or she can share data or information concerning her source site by performing a dance on the dance area. If her source is drained means she becomes a scout and starts to randomly search for a new source.

3. Finally, the onlooker bees waiting in the hive watch the dances marketing the profitable sources and decide a primary site depending on the frequency of a dance in proportion to the excellence of the source.
Producing initial food source sites

In bee’s behaviour, if the search space is considered to be the atmosphere of the hive, it contains the food source sites, and this algorithm starts with randomly generating food source sites that correspond to the solutions in the search space. Primary food sources are taken as randomly within the range of the boundaries of the parameters.

\[
    x_{ij} = x_{ij}^{\min} + \text{rand}(0, 1) \left( x_{ij}^{\max} - x_{ij}^{\min} \right) \quad (3)
\]

In the above, \( i = 1 \ldots SN, j = 1 \ldots D \), where \( SN \) is the number of food sources and \( D \) is the number of optimization sources. In this phase, the counters which store the trials of solutions are reset to 0. Then, the population of the food sources or solutions is focused to repeat cycles of the search processes of the all three types of bees. The termination criteria for the mABC algorithm can be, attainment a Maximum Cycle Number (MCN) or achieving an error tolerance (\( \varepsilon \)).

Employed bees sending to the food source locations

In this phase, each employed bee is allied with only one food source site. Therefore, the number of food source locations is equivalent to the number of employed bees. An employed bee yields a modification on the position of the food source (solution) in her memory contingent on local information (visual information) and finds a neighboring food source and then evaluates its quality. In mABC, for each parameter \( x_{ij} \) an equivalently allocated random number, \((0 \leq R_{ij} \leq 1)\) is generated and if the random number is less than the Modification Rate (MR), then the parameter \( x_{ij} \) is modified as in the Equation (2).

\[
    v_{ij} = \begin{cases} 
    x_{ij} + \phi_{ij} \left( x_{ij} - x_{kj} \right), & \text{if } R_{ij} < \text{MR} \\
    x_{ij} & \text{otherwise} 
    \end{cases} \quad (4)
\]

Within the range \([1, D]\) of random integer is defined as \( j \) and \( k \in \{1, 2, \ldots SN\} \) is defined as randomly chosen index that has to be altered from and ModificationRate(MR) takes value between 0 and 1. In this process, a lower value of MR may cause results to improve slowly, and a higher value of MR may cause excessive diversity in a solution and hence in the population.

If a parameter value is exceeds its predetermined limits, the parameter can be fixed to an acceptable value. In this process, the parameter values are exceeding its boundary is set to its boundaries. If \( x_{i} > x_{i}^{\max} \) then \( x_{i} = x_{i}^{\max} \) also, if \( x_{i} < x_{i}^{\min} \) then \( x_{i} = x_{i}^{\min} \). After producing \( v_{i} \) within the boundaries, a fitness value for a minimization problem can be assigned to the solution \( v_{i} \) by Equation (3).

\[
    \text{fitness}_{i} = \begin{cases} 
    \frac{1}{1 + f_{i}} & \text{if } f_{i} \geq 0 \\
    1 + \text{abs}(f_{i}) & \text{if } f_{i} < 0
    \end{cases} \quad (5)
\]

In above equation \( f_{i} \) is the cost value of the solution \( v_{i} \). For maximization difficulties, the cost function can be directly used as a fitness solution. A greedy selection is applied between \( x_{i} \) and \( v_{i} \); then the better one is selected depending on fitness values signifying the nectar amount of the food sources at \( x_{i} \) and \( v_{i} \). If the source at \( v_{i} \) is superior to that of \( x_{i} \) in terms of profitability, the employed bee stores the new position and forgets the old one. Else, the previous position is kept in memory. If \( x_{i} \) cannot be enhanced, its counter asset the number of trials is incremented by 1, else, the counter is reset to 0.
**Probability values calculation for probabilistic selection**

After all employed bees finish their searches and they distribute their information related to the nectar amounts and the locations of their sources with the onlooker bees on the dance area. It is defined as the multiple interaction features of the Artificial bees of mABC. An onlooker bee estimates the nectar data received from all employed bees and chooses a food source location with a probability related to its nectar quantity. This probabilistic selection based on the fitness values of the answers in the population. In mABC, roulette wheel selection scheme in which every slice is proportional in range to the fitness value is employed through Equation (6).

\[
p_i = \frac{\text{fitness}_i}{\sum_{i=1}^{SN} \text{fitness}_i}
\]  

(6)

**Food source fitness site selection**

In the mABC algorithm, a random real number within the range [0, 1] is caused for each source. If the probability value \( p_i \) in Eq. (6)) connected with that source is larger than this random number then the onlooker bee produces a modification on the position of this food source site by using Eq. (4) as in the case of the employed bee. Following that the source is assessed, greedy selection is applied and the onlooker bee either onestore the new position by replacing the old one or keeps the old one. If solution \( x_i \) cannot be enriched, its counter holding trials is incremented by 1; else, the counter is reset to 0. This process is frequent until all onlookers are distributed onto food source sites.

The pseudo code of the modified ABC is given below:

1. initialize population \( x_{ij} \)
2. evaluate population \( x_{ij} \)
3. cycle = 1
4. repeat//new food source population produced for employed bee//
5. for \( i = 0, i++, i = SN \)
6. do
7. Produce new food source by the below eq. \( v_{ij} = \begin{cases} x_{ij} + \Phi_j\left(x_{ij} - x_{kj}\right), & \text{if } R_j < MR \\ x_{ij} & \text{otherwise} \end{cases} \)
8. apply greedy \((x_i, v_i) \rightarrow \) select better one
9. if solution \( x_i \) does not improve \( trial_i = trial_i + 1 \)
10. else
11. \( trial_i = 0 \)
12. end for
13. calculate \( p_i \) by the blow Eq. \( p_i = \frac{\text{fitness}_i}{\sum_{i=1}^{SN} \text{fitness}_i} \)
   // a new food source population produced for onlookers//
14. \( t = 0, i = 1 \)
15. repeat
16. if random \( < p_i \)
17. then
18. go to step 7 to 11
19. \( t = t + 1 \)
20. end if
21. until \( t = SN \)
    // Determine Scout/
22. if \( \text{max}(\text{trial}_i > \text{limit}) \)
23. then
24. replace \( x_i \) with a new randomly produced solution by: \( x_j = x_{j_{\text{min}}} + r \) and \( (0, 1) \left( x_{j_{\text{max}}} - x_{j_{\text{min}}} \right) \)
25. end if
26. Memorize the best solution achieved so far
27. Cycle = cycle + 1
28. Until (cycle = Minimum Cycle Number)

In a cycle process, following all employed and onlooker bees complete their searches, this algorithm finds if there is any exhausted source to be unrestrained. In order to decide if a source is to be unrestrained, the counters which have been updated during search are used. If the value of the counter is larger than the control parameter of the ABC algorithm, identified as the boundary, then the source connected with this counter is assumed to be exhausted and is unrestrained. The food source unrestrained by its bee is changed by a new food source discovered by the scout, which denotes the negative feedback mechanism and fluctuations property in the self-organization of mABC. This is simulated by producing a site position randomly and replacing it with the unrestrained one. Assume that the unrestrained source is \( x_i \), and then the scout randomly finds out a new food source to be replaced with \( x_i \).

3.3.2. The Kernel Extreme Learning Machine (KELM)

Given samples \( \{(x_i, t_i): i = 1, 2, \ldots, N; x_i \in R^p, t_i \in R^q\} \), where \( x \) is the feature vector and \( t \) is the class label vector and the Single-hidden Layer Feed-forward Neural Network (SLFN) is used to identify the sample [26].

\[
\sum_{j=1}^{m} \beta_j g\left( \alpha_i^T x_j - b_j \right) = o_i, \ j = 1, 2, \ldots, N \tag{7}
\]

Where, \( m \) is the number of hidden neuron; \( o_i \) is the output of \( j \)th sample; \( g(.) \) is the activation function \( b_i \) is the threshold of the \( i \)th hidden neuron; \( \alpha_i \) and \( \beta_i \) are the input and output weight vectors, correspondingly. If the output \( o \) can approximate \( t \), derive:

\[
\sum_{j=1}^{m} \beta_j g\left( \alpha_i^T x_j - b_j \right) = o_i, \ j = 1, 2, \ldots, N \tag{8}
\]

(6) can be written compactly as:

\[
G\beta = T \tag{9}
\]

Where,

\[
G = \begin{bmatrix} g\left( \tilde{c}_1^T x_1 - b_1 \right) & \cdots & g\left( \tilde{c}_m^T x_1 - b_m \right) \\ \vdots & \cdots & \vdots \\ g\left( \tilde{c}_1^T x_N - b_1 \right) & \cdots & g\left( \tilde{c}_m^T x_N - b_m \right) \end{bmatrix}
\]
\[ \beta = [\beta_1, \beta_2, \ldots, \beta_m]^T \text{ And } T = [t_1, t_2, \ldots, t_N]^T \]

To solve (7), the ELM adopts a least squares error to get solution \( \hat{\beta} \):

\[ \hat{\beta} = G^T \]

(10)

Where \( G^T \) is the Moore-Penrose generalized inverse of \( G \). Function \( g(.) \) is typically unidentified, incorporate kernel functions in \( g(.) \). This is the so-called KEML. The kernel matrix \( K = [K(x; x_1) \ldots K(x; x_N)]^T \) \((\cdot) \text{ is the kernel function)}\) is introduced into (9) and (10) to estimate the output of the KELM:

\[ o = KT \]

(11)

Herein, the Gaussian kernel function (RBF) is adopted.

\[ K(x_i; x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma}} \]

(12)

In above, \( \sigma \) is the width of RBF. The number of hidden neuron \( m \) needs to be optimized for KELM. To do so, the mABC is used to optimize \( m \) in the training processing of the KELM.

### 3.3.3. The New Method Based on mABC-KELM

The proposed network intrusion detection method can be summarized as follows:

**Step 1:** Format the intrusion data set into standard form through preprocessing step.

**Step 2:** Fuse the data using principal component analysis (PCA) to obtain the feature vector. PCA is used to determine a subset of features based on a feature reduction method.

**Step 3:** The selected principal components called bacteria principal components are the basis of feature subsets. BF is used to search the PCA space for feature selection. In turn, the feature set obtained through this process is presented to the classifier.

**Step 4:** Train the KELM using the feature vectors, and optimize the hidden neuron number using modified ABC.

**Step 5:** Test the performance of the ABC-KELM detection model. A workflow block of the proposed mABC-KELM intrusion detection method.

### 4. EXPERIMENTAL RESULTS

In this section, the proposed mABC-KELM performance is evaluated and compared with existing with existing intrusion detection algorithms such as Fuzzy Neural Network with Expectation Maximization (FNN-EM), Fuzzy based Secure Intrusion Detection System (FSIDS) and EAACK [27] in presence of malicious node environment.

The proposed IDS is simulated with Network Simulator tool (NS 2.34). In proposed simulation, 101 sensor nodes move in a 1000 meter x 1000 meter square region for 100 seconds simulation time. Here, each node moves independently with the same average speed. Every one of nodes has the same transmission range of 250 meters. The simulated traffic is Constant Bit Rate (CBR). Proposed simulation settings and parameters are summarized in table 1.

**Performance evaluation**

The performance is calculated based on the following parameters.
Detection Rate Comparison

Fig. 2 shows that the graphical representation of detection rate comparison between proposed mABC-KELM and existing methods. The proposed method has high detection rate compared than existing methods. Because of preprocess. In this preprocess, the traces sensitivity values with false alarm rates are extracted. The attack traces were then classified, with detection rate evaluated from the number of alerts emerge from this assessment. Therefore malicious activity is detected with efficient manner in proposed method.

Communication overhead

Fig. 3 shows that the graphical representation of overhead comparison between proposed mABC-KELM and existing methods. The mABC-KELM algorithm achieves less overhead compared than existing FNBN-EM, FSIDS and EAACK. The reason is that, when the data effective time is small enough, almost all the features meet the desired event data in the proposed method.

Packet Delivery Ratio (PDR)

Fig. 4 shows that the graphical representation of PDR comparison between proposed mABC-KELM and existing methods. PDR is defined as the ratio of total messages transmitted to total messages received at the destination. The mABC-KELM algorithm achieves high PDR compared than existing FNBN-EM, FSIDS and EAACK. Packet is delivered via reliable nodes through stable link. Successfully all the packets are delivered to the destination.

Table 1

<table>
<thead>
<tr>
<th>simulation parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Nodes</td>
</tr>
<tr>
<td>Area Size</td>
</tr>
<tr>
<td>Mac</td>
</tr>
<tr>
<td>Radio Range</td>
</tr>
<tr>
<td>Simulation Time</td>
</tr>
<tr>
<td>Traffic Source</td>
</tr>
<tr>
<td>Packet Size</td>
</tr>
<tr>
<td>Mobility Model</td>
</tr>
<tr>
<td>Protocol</td>
</tr>
</tbody>
</table>

![Figure 2: Detection Rate Comparison](image1.png)

![Figure 3: time vs. overhead](image2.png)
End to end delay

Fig. 5 shows that the graphical representation of end to end delay comparison between proposed mABC-KELM and existing methods. The mABC-KELM has less delay than existing systems. End to end delay should be kept minimum in order to satisfy QoS. The proposed system reduces delay by means of cluster based routing. Network partitioning will be reduced by integrating this routing in all networks.

Packet Integrity Rate

Fig. 6 shows that the graphical representation of Packet Integrity Rate comparison between proposed mABC-KELM and existing methods. The mABC-KELM has high packet integrity rate than previous systems. The proposed system increases packet integrity rate by adding encryption and decryption mechanism.

Network Lifetime

Fig. 7 shows that the graphical representation of Network Lifetime comparison between proposed mABC-KELM and existing methods. The mABC-KELM has more lifetime than existing methods. The proposed system increases network lifetime by adding link stability rate.

5. CONCLUSION

Intrusion detection is very important for the computer security. In this research, a performance enhancement model is proposed for intrusion detection system based on an optimal feature subset selection using several
bacteria principal components. The feature selection has been accomplished using the techniques of PCA and GA. The selected features subsets are presented to modified ABC optimized KELM. The innovation of this work lies in the development and implementation of the PCA and mABC-KELM in the intrusion detection for the first time. Experimental tests have been carried out to calculate the performance of the new method. The test result has showed satisfactory and effective intrusion detection performance of the proposed ABC-KELM method. In addition, when compared with PCA-ABC-KELM, it proves that the performance of the proposed PCA-mABC-KELM method is superior to its rivals in terms of both detection accuracy and training speed. Thus, the Proposed PCA-mABC-KELM method shows promising applications in the domain of intrusion detection.

REFERENCES


