A Novel Intrusion Detection System for RPL Based IoT Networks with Bio-Inspired Feature Selection and Ensemble Classifier

Jayaprakash Pokala (pokalajayaprakash@gmail.com)
JNTUA CEA: Jawaharlal Nehru Technological University Anantapur College of Engineering Ananthapuramu https://orcid.org/0000-0002-2131-9454
B. Lalitha
JNTUA: Jawaharlal Nehru Technological University Anantapur

Research Article

Keywords: Internet of Things, Simulated annealing, Salp swarm algorithm, Voting ensemble classifier, Intrusion detection system.

Posted Date: April 28th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-442429/v1

A Novel Intrusion Detection System for RPL Based IoT Networks with Bio-Inspired Feature Selection and Ensemble Classifier

P. Jaya Prakash[1], Dr B. Lalitha[2,*]
[1]Research Scholar, Dept. of CSE, JNTUA University Ananthapuramu
[2]Assistant Professor, JNTUA University College of Engineering, JNTUA Ananthapuramu
E-mail: pokalajayaprakash@gmail.com, lalitha.cse@jntua.ac.in

Abstract. Internet of Things (IoT) is the powerful latest trend that allows communications and networking of many sources over the internet. Routing protocol for low power and lossy networks (RPL) based IoT networks may be exposed to many routing attacks due to resource-constrained and open nature of the IoT nodes. Hence, there is a need for network intrusion detection system (NIDS) to protect RPL based IoT networks from routing attacks. The existing techniques for anomaly-based NIDS (ANIDS) subjects to high false alarm rate (FAR). Therefore, a novel bio-inspired voting ensemble classifier with feature selection technique is proposed in this paper to improve the performance of ANIDS for RPL based IoT networks. The proposed voting ensemble classifier combines the results of various base classifiers such as logistic Regression, support vector machine, decision tree, bidirectional long short-term memory and K-nearest neighbor to detect the attacks accurately based on majority voting rule. The optimized weights of base classifiers are obtained by using the feature selection method called simulated annealing based improved salp swarm algorithm (SA-ISSA), which is the hybridization of particle swarm optimization, opposition based learning and salp swarm algorithm. The experiments are performed with RPL-NIDDS17 dataset that contains seven types of attack instances. The performance of the proposed model is evaluated and compared with existing feature selection and classification techniques in terms of accuracy, attack detection rate (ADR), FAR and so on. The proposed ensemble classifier shows better performance with higher accuracy (96.4%), ADR (97.7%) and reduced FAR (3.6%).

Keywords: Internet of Things, Simulated annealing, Salp swarm algorithm, Voting ensemble classifier, Intrusion detection system.

1. Introduction

The massive development of IoT increases the physical device connections on internet. The long-time operation of huge number of devices requires low power network consumption [1]. The Ipv6 based low-power wireless personal area network (6LoWPAN) is a small IoT network that can enable the devices of IoT to operate on a low power [2]. Conventional routing protocols of IoT are not suitable for 6LoWPAN networks due to the lossy and low-power nature of 6LoWPAN networks. Therefore, RPL networks are introduced to provide efficient routing in 6LoWPAN networks [3, 4]. The main advantage of these control messages is that it follows particular patterns in case of repair and creation of IoT networks. Though RPL based IoT networks give more merits in routing, it is exposed to several routing attacks due to mobility and limited battery life. These attacks include selective forwarding, sybil, blackhole, sinkhole, hello flooding, clone ID and local repair attacks [1]. Apart from these, attackers may compromise the privacy and security of users through eavesdropping and economic losses to get access to their personal data. As a result, the solution to protect IoT from various routing attacks is network intrusion detection systems [5].

NIDSs are categorized into two namely anomaly-based NIDS and signature-based NIDS (SNIDS) [4]. SNIDS matches the network traffic with stored attack signatures to identify and detect the attacks [6]. ANIDS takes the normal traffic behavior and network traffic deviation as the baseline to detect any attacks [7]. Though SNIDS brings higher accuracy and less FAR, it is unable to detect novel attacks in existing methodologies. On the other hand, ANIDS can detect novel attacks but with high FAR. The performance of ANIDS mainly depends on the effectiveness of the analysis model i.e., classifier and quality of training dataset [8]. Thus, an effective ANIDS model built using machine learning (ML) algorithms are required to detect novel attacks in IoT. These algorithms train a classifier with normal and anomaly data for attack detection in IoT network. Many literature works on IDSs are focused on different classification techniques like ML classifier [8, 9], deep learning classifiers [10, 11], or ensemble learning [12, 13]. Of this, ensemble learning combines multiple classifiers to make better classification with reduced FPR compared to individual classifiers. The widely used algorithms in ensemble learning are majority voting, bagging and AdaBoost [14]. In this paper, a novel bioinspired voting ensemble classifier is proposed. Bio-inspired algorithms are effective in finding the best solution for

classification and feature selection [15]. A recent bio-inspired algorithm inspired from the swarm behavior of salp commonly termed as SSA is adopted in this research for both feature selection and classification. The advantages of SSA are few parameters requirement, low computational cost and simple implementation [16]. The convergence speed of traditional SSA is enhanced by integrating opposition based learning (OBL) strategy at the initialization stage and named as Improved SSA (ISSA) [17]. In this paper, a hybridized approach incorporated from both ISSA and particle swarm optimization (PSO) [18] is employed to optimize the weights of the voting classifier.

The FAR and accuracy of the classifier is dependent on the quality of the dataset used for training [8]. For training purpose, publicly available datasets such as KDD99, UNSW-NB15 and NSL-KDD cup 99 are utilized by many researchers in NIDS evaluation. However, NIDS evaluation for RPL based IoT networks with existing datasets such as NSL-KDD cup 99 and KDD99 are found to unfit and obsolete [4]. Hence, RPL-NIDDS17 dataset [19] is utilized for training the proposed NIDS for RPL based IoT networks. Though this dataset is imbalanced, the classification task will misclassify the minority classes. And so, it is necessary to balance the minority and majority classes by oversampling the minority classes. Besides, many sampling methods are available to balance the datasets. Synthetic minority over-sampling technique (SMOTE) is the most effective sampling method to balance the datasets [20]. If we include all the features for training the classifier, it will affect the performance of classification in terms of complex computation. Therefore, feature selection techniques are widely used by many researchers to lessen the computational complexity in NIDSs [9, 14]. In this paper, the ISSA [17] is applied for feature selection before classification. Moreover, simulated annealing approach [21] is integrated with ISSA to enhance the performance of feature selection in addition to search space exploitation [22].

In this paper, a novel feature selection and voting ensemble classifier-based NIDS is proposed for security against seven types of attacks in RPL based IoT networks. At first, the dataset is preprocessed in three steps i.e., cleaning, encoding and normalization. Though the dataset is imbalanced, a common method called SMOTE is applied for dataset balancing. Then feature selection is performed with SA-ISSA to minimize the size of balanced dataset by considering the best features from the dataset. The proposed voting classifier is the ensemble of ML-based classifiers namely decision tree (DT), logistic regression (LR), K-nearest neighbor (KNN), support vector machine (SVM) and deep learning-based classifiers namely bidirectional long short-term memory (Bi-LSTM). The weights of all the classifiers are optimized using PSO-ISSA technique to achieve higher attack detection rate (ADR). Finally, the performance of the proposed feature selection and classification approaches is evaluated and compared with existing methods. The major contribution of the proposed work is summarized as follows.

- A bio-inspired voting ensemble classifier based on SA-ISSA technique is proposed to detect seven types of attacks in the RPL based IoT networks.
- A bio-inspired feature selection technique based on PSO-ISSA technique is introduced to minimize the dimensionality of dataset and to minimize the FAR.
- The performance of the proposed feature selection algorithm is compared with existing feature selection algorithms such as PSO, GA, GWO, original SSA and improved SSA in terms of best fitness, average fitness, average error, standard error and worst fitness.
- The performance of the proposed bio-inspired voting-based NIDS model is compared with existing bio-inspired voting-based classifiers in terms of accuracy, precision, ADR, specificity, F-measure and FAR.

The remainder of this paper is arranged as follows. Section 2 details the existing works for RPL based NIDS, bio-inspired feature selection and ensemble-based classification. Section 3 explains the background of methods used in the proposed NIDS. A brief overview of the proposed NIDS is given in Section 4. The experimental results are discussed in Section 5. Finally, Section 6 concludes this paper.

2. Related Work

This section explores the existing NIDS for RPL based IoT including feature selection and ensemble classification processes.

2.1 Ensemble Classifier

There are many literatures that utilized ensemble classifier as IDS. Ranga and Verma [8] investigated the ML algorithms to detect DoS attacks in IoT. Datasets utilized for training the classifier are NSL-KDD, UNSW-NB15 and CIDDS-001. They have used various ensemble classifiers like random forest (RF), gradient boosted machine, AdaBoost, extremely randomized trees, extreme gradient boosting and single classifiers like multi-layer perceptron and classification and regression trees (CART). The performance is evaluated in terms of different measures like accuracy, sensitivity, specificity, FPR and AUC. Similarly, Al-Abassl et al. [23] utilized an ensemble deep learning-based IDS with deep neural network and DT classifiers for IoT in an industrial control system (ICS). Shahraki et al. [24] compared different versions of boosting algorithms such as modest AdaBoost, gentle AdaBoost and real AdaBoost for NIDS evaluation. Kasongo and Sun [25] analyzed the performance of ML-based IDS using a feature selection method called XGBoost algorithm. This approach utilized several classifiers like LR, KNN, SVM, ANN and DT to analyze the UNSW-NB15 dataset.

Yang et al. [12] introduced a paralleled quadratic ensemble learning based on gradient boosting decision tree (GBDT) for IDS. The detection accuracy is higher for CICIDS17 dataset against attacks like distributed DoS (DDoS), port scan, benign, web attack traffic and infiltration. Bhati et al. [13] presented an IDS based on majority voting based ensemble of discriminant classifiers. KDDcup99 dataset is utilized for evaluation. This technique detects all types of attacks with higher accuracy. The above-mentioned literatures utilized various ensemble classifiers in IDS. The ensemble-based classifier for RPL based IoT networks are introduced by Verma and Ranga [1]. They utilized an ensemble learning-based NIDS (ELNIDS) for detecting seven types of routing attacks in RPL based IoT. It contains four different classifiers namely, RUSBoosted trees, bagged trees, boosted trees and subspace discriminant boosted trees. Different measures such as accuracy, area under curve and ROC curve are evaluated for performance assessment. RPL-NIDDS17 dataset is utilized to train the classifiers. However, they did not utilized classifier tuning method and feature selection technique.

2.2 Hybrid Feature Selection and Voting Ensemble Classifier

Some literatures combined the merits of both feature selection and voting based ensemble learning. Zhou et al. [26] proposed new IDS with ensemble learning and feature selection techniques. A hybrid approach of correlation-based feature selection and bat algorithm (CFS-BA) is employed to reduce the data dimension. Then a voting ensemble methodology is introduced with the combination of RF, c4.5 and forest by penalizing (Forest PA) algorithms. CIC-IDS2017, AWID and NSL-KDD datasets are utilized in the experimental analysis to show the effectiveness of CFS-BA ensemble method. The feature selection technique reduced the model building time of this model compared to those models with all the features. Moreover, it exhibits better performance than other approaches. Asadi et al. [27] utilized voting classifier and feature selection techniques to detect the botnet attacks. PSO algorithm is utilized in feature selection process to select effective features from the dataset. SVM, DT C4.5 and deep neural network algorithms are utilized in the voting system to detect the botnet attack. The datasets utilized in this proposed work are Bot-IoT and ISOT. The experimental results revealed the low accuracy of this approach in both datasets.

Tama et al. [28] designed a two-stage ensemble for ANIDS. The feature selection process utilized a hybrid of three optimization algorithms i.e., genetic algorithm (GA), PSO and ant colony algorithm. The datasets utilized for training the classifier are UNSW-NB15 and NSL-KDD. The two-stage ensemble consists of a Meta classifier with another Meta classifier as the base classifier. The training time of proposed model is reduced with the optimal feature selection method. The accuracy, sensitivity and precision of the proposed approach is higher than existing works. However, the FPR is higher than existing works. Kumar et al. [29] proposed a cyber-attack detection framework for internet of medical things (IoMT) based on fog-cloud architecture and ensemble learning. The ensemble learning includes NB, RF and DT classifiers. The outputs of classifiers are directed to XGBoost model in order to detect the attacks. It produced higher ADR, accuracy and reduced the FAR up to 5.59%. The above mentioned works utilized an ensemble learning-based classifier with feature selection technique for IDS. Nevertheless, there are no other works for RPL based IoT with bio-inspired hybrid feature selection and ensemble classification.

2.2 IDS in RPL based IoT

Though many works proposed various IDS for IoT networks, only few literatures proposed IDS for RPL based IoT networks. Cakir et al. [10] presented a deep-leaning-based gated recurrent unit (GRU) to detect hello flooding (HF) attacks with high accuracy rate in the RPL based IoT networks. They have compared the performance measures of proposed model with SVM and LR classifiers. The performance of this approach is measured in terms of mean square error (MSE), accuracy, root mean square error (RMSE), mean absolute error (MAE), delay, energy consumption and packet delivery rate (PDR). Though GRU based deep learning model shows higher performance, it can detect only one attack. Pu [30] designed a Gini Index-based countermeasure (GINI) to protect the RPL based networks from Sybil attack. The performance of proposed GINI countermeasure is compared with two existing algorithms like two-step detection and SecRPL. This approach showed improved performance interns detection latency and detection rate. However, it cannot detect other types of attacks.

Murali and Jamalipour [31] introduced a bio-inspired lightweight IDS based on artificial bee colony (ABC) to protect mobile RPL from Sybil attack. The performance of this model is analyzed for three types of Sybil attacks in terms of specificity, control traffic overhead, accuracy, packet delivery ratio, sensitivity and energy consumption. Though this approach gives better results, it can detect only one kind of attack. Gothawal and Nagaraj [32] proposed game models-based anomaly intrusion detection system (GAIDS) for the protection of RPL based networks. It has two interrelated formulations like evolutionary game for confirmation of attack and stochastic game for detection of attack. The detected attackers are isolated by GAIDS in order to maintain the performance of GAIDS. This method detects many RPL attacks such as local pair, rank, neighbor and DIS attack. Though the proposed approach can detect the RPL attack, it was unable to detect new types of attacks.

3. BACKGROUND

3.1 Salp Swarm Algorithm

SSA is a novel optimization algorithm that mimics the behavior of salp (kind of marine tunicate). The Salp's are planktonic tunicate with barrel shape and belong to the Salpidae's family. They have a unique swarm behavior called salp chain that helps them to do better movements and foraging. The salp chain behavior can be mathematically modeled for optimization problems commonly called as salp swarm algorithm. Initially, the population is divided into two groups called follower and leader. The leader group is the front of salp chain and followers are other salps. The direction of movement of leader is followed by the follower salps. The position of the salp is determined for n-dimension, which is the search space of given problem. The target of salp swarm is food source searching. The position of the leader salp is updated using Equation (1).

$$x_m^1 = \begin{cases} F_m + k_1\left((ub_m - lb_m)k_2 + lb_m\right) & k_3 \geq 0.5 \\ F_m - k_1\left((ub_m - lb_m)k_2 + lb_m\right) & k_3 < 0.5 \end{cases} \quad (1)$$

Where, the position of leader in m^{th} dimension is represented by x_m^1. The food source for m^{th} dimension is F_m. ub_m and lb_m represents the upper and lower bound value of m^{th} dimension. Random values are represented by k_2, and k_3. The balance between exploration and exploitation of SSA is maintained by a significant controlling parameter k_1. The k_1 parameter can be calculated by using Equation (2).

$$k_1 = 2e^{-\left(\frac{4l}{L}\right)^2} \quad (2)$$

Here, the current iteration is represented by k_1 and maximum number of repetitions of the algorithm is represented by L. The random values k_2 and k_3 are in the interval [0,1]. The position of followers can be updated using Equation (3).

$$x_m^i = \frac{1}{2}(x_m^i + x_m^{i-1}) \quad (3)$$

Here, the value of $i \geq 2$ and the position of i^{th} follower in m^{th} dimension is represented by x_m^i. The initialization of SSA starts by randomly generating the position of populations. Then the fitness value is evaluated. The best fitness value is considered as F_m, which is the goal for the followers. In each iteration, k_1 value, the position of

leader and follower is updated using Equations (1), (2) and (3). All these steps except initialization will be continued until reaching the maximum number of iterations..

3.2 Opposition Based Learning

The OBL technique is utilized among many researches as an optimization technique to improve the quality of initial population through diversifying the population. This strategy searches the search space in both directions. The original solution and opposite solution are included in the directions. From both the solutions, the worst solutions are taken and the opposition is applied on the worst solution. The opposite position $\tilde{x}$ of original position $x \in [a, b]$ in the j-th dimension will be calculated using the Equations (4).

$$\tilde{x}_j = a_j + b_j - x_j \tag{4}$$

Here, D is the problem dimension. j= 1, 2, 3, …, D. The original position x and opposite position $\tilde{x}$ will be denoted by Equations (5) and (6).

$$x = [x_1, x_2, x_3, \ldots, x_D] \tag{5}$$

$$\tilde{x} = [\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \ldots, \tilde{x}_D] \tag{6}$$

If the fitness value of opposite population f ($\tilde{x}$) is better than original position f(x), then $x=\tilde{x}$, else x=x. Thus, optimization is performed using opposite population.

3.3 Simulated Annealing

SA is inspired from the physical annealing process of metalwork. It is widely used in many optimization problems to obtain a best neighboring solution. The initial stage of SA randomly generates initial solutions (R) taken as best solutions (R_{best}). New neighbor solution (R*) can be generated from the current solution. Then the fitness function is calculated for new solution (R*) and compared with the best solution(R_{best}). The difference in the fitness function of both solutions is calculated using following Equation (7).

$$\theta = f(R^*) - f(R) \tag{7}$$

If ($\theta > 0$), then the new neighbor (R*) is considered as the best solution (R_{best}) in the next iteration. Otherwise, ($\theta < 0$)) the worse solutions are accepted with a probability as given in Equation (8).

$$P = e^{-\theta/T} \tag{8}$$

Here, θ is difference in the fitness function of both solutions. Here, f represents the fitness function, R* is a new neighbor solution and R is the current best neighbor, T is a control parameter called absolute temperature.

3.4 Particle Swarm Optimization

PSO is inspired from the swarm behavior of both bird flocking and fish schooling. The swarm is initialized by generating the velocity (v_i) and positions (p_i) in j-th dimension. PSO initializes its main loop to evaluate all particles via fitness function determination. The fitness solution is evaluated with its global best and best value. The position as well as velocity of particles can be updated using Equations (9) and (10) respectively. These steps except initialization are repeated until reaching the maximum number of iterations.

$$x_{ij}^{(t+1)} = x_{ij}^{(t)} + v_{ij}^{(t+1)} \tag{9}$$

$$v_{ij}^{(t+1)} = w\, v_{ij}^{(t)} + c_1\, r_1 (x_{ij}^{p(t)} - x_{ij}^{(t)}) + c_2\, r_2\, (x_j^{g(t)} - x_{ij}^{(t)}) \tag{10}$$

In the j-th dimension, x_{ij} is the position of i-th particle and v_{ij} is the i-th velocity. The current iteration is represented by t and inertia weight is represented by w. c_1 and c_2 represents the coefficients of acceleration. In j-th dimension, the best previous i-th particle position is represented by $x_{ij}^{p(t)}$. In j-th dimension, the position of global best is represented by $x_j^{g(t)}$. The random variables r_1 and r_2 lies in the range between 0 and 1.

3.5 Classification Methods

SVM is one of the most popular supervised learning algorithms that perform both regression and classification. However, it is widely used for classification in machine learning models. It can handle simple and complex datasets with higher accuracy compared to other algorithms. In classification, SVM transforms the data points and find a hyperplane with maximum margin from multiple decision boundaries to classify the data points in n-dimensional space using kernel trick concepts. The Gaussian RBF, polynomial or linear kernel can be utilized to minimize the computational complexity related to the prediction of new data points. The data vectors closer to the hyperplane called support vectors determine the position of the hyperplane.

DT is also a supervised learning technique that can be utilized to perform both regression and classification. It utilizes tree structure to classify the data based on given conditions in which root node represents the whole training dataset. Decision rules such as Boolean function are represented as branches and label of output class is represented by each leaf node. The DT algorithm starts with root node containing the whole dataset. The best attributes are selected using attribute selection measure. The decision node is then created with best attributes. This process is repeated until finding the leaf node for all branches.

KNN is also one of the simplest supervised learning techniques which are utilized for both regression and classification. This algorithm assumes the similarity between training data and new data to classify a new data. The similarity is calculated between the training data and the testing data using Euclidean distance. During classification, the data is assigned to the category for which the similarity is maximum. The number of nearest neighbors is calculated for testing data. The category of new data is selected for those data having a large number of nearest neighbors.

LR is also a supervised learning classification technique mainly utilized to predict the probability of target variable. The target variable can take only discrete values for given set of features in a classification problem. It can predict the output of a categorical dependent variable. It can classify the new dataset using discrete and continuous datasets.

Bi-LSTM is the advanced process of conventional LSTM. LSTM is one of the widely used recurrent neural networks. It is a sequence processing model that can solve the long-term dependencies. Bi-LSTM consists of two independent LSTMs. The input is given in forward direction in one LSTM and in backward direction in another LSTM. Thus, Bi-LSTM increases the amount of information available to the network by connecting the forward and backward information about the input data at every time step.

4. Proposed Methodology

In the proposed work, a novel bio-inspired feature selection algorithm and voting-based classifier is introduced for attack detection in RPL based IoT networks. Initially, encoding, scaling and cleaning methods are applied to preprocess the dataset. After preprocessing, the dataset is balanced with SMOTE technique. The essential features are selected from balanced dataset by using novel feature selection technique (SA-improved SSA). The selected features are then divided into training and testing data. Finally, the proposed voting-based classification algorithm (PSO-improved SSA) is applied to classify the routing attacks. Figure 1 displays the complete framework of the proposed NIDS.

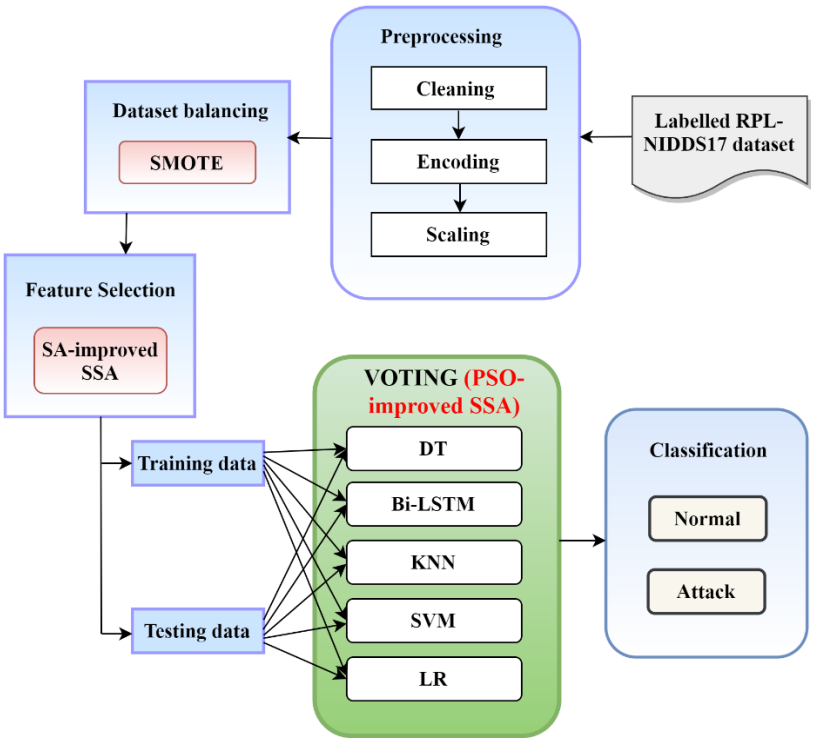

Figure 1. Block diagram of proposed methodology for NIDS

4.1 Dataset Preprocessing and Balancing

The first step in data preprocessing is cleaning process to improve the quality of dataset. This step performs removing duplicates, handling missing values and encoding. Machines can read only numeric data but the dataset consists of both numeric and nominal data. Thus, encoding is utilized to convert the characters in the dataset to numeric values. The last step in preprocessing is data scaling performed to speed up the process. The features in the dataset highly vary in range with magnitude and units. It is necessary to keep all the data in one format and hence scaling can normalize the data within the range between 0 and 1.

The RPL-NIDDS17 dataset is consisted of large number of normal instances compared to the number of attack instances. That is, the routing attack becomes minority class and the normal traffic becomes majority class. This type of data is known as imbalanced data. This makes the classifier to be dominated with normal class which in turn reduces the possibilities of attack detection. Classification with these kinds of imbalanced data will bias in favor of the normal class. Hence, the classification accuracy will be poor for attack class compared to normal class. To overcome this issue, many techniques are suggested to balance the dataset. One of the recent algorithms is SMOTE, which is widely utilized by many researchers. It can over-sample the minority classes by duplicating randomly selected data in order to balance the dataset. A subset of data is taken as an example from the attack class where SMOTE finds the k- nearest neighbors. New instances of attack class are synthesized between the nearest neighbor and instances of attack class. These synthetic instances are then added with the original dataset. The new oversampled dataset is used to train the classification models.

4.2 Feature selection

The presence of unwanted and redundant data consumes higher computation time and degrades the performance of classifiers. To overcome these issues, optimization algorithms pick only the best features from all kind of features with less computational effort in a reasonable time. This section details the proposed SA-improved SSA employed for feature selection. The selected features are divided to be used for training and testing. The training data is directed as input to the voting classifier.

A) Improved SSA (ISSA)

The improved SSA is the advanced method of traditional SSA. A subset of population with lowest fitness is selected from population of SSA to apply opposition instead of opposing all the population. This is given in Equation (11).

$$\widetilde{x_m^i} = lb_m + ub_m - x_m^i \quad (11)$$

Where $x_m^i \in [lb_m, ub_m]$, j=1, 2, …, D. The dimension of the problem is represented by D. The subset of population is chosen based on the fitness value. The solution with worst fitness value in SSA means they are far away from the original solution. Hence, opposition is applied for the worst solutions to search in the opposite direction. The opposite solutions ($\widetilde{x_m^i}$) are added with the original solutions (x_m^i). Then the fitness value is determined with KNN classifier for each solution in ($x_m^i \cup \widetilde{x_m^i}$). In fitness calculation, the preprocessed dataset is divided into training and testing sets using k-fold cross-validation. KNN classifier is trained with training data and the classification is performed using testing data. Accuracy is calculated for the KNN classifier to obtain error rate of the classifier. This error rate is saved as the cost function of fitness calculation. Based on this fitness value, the position is updated with best solutions from ($x_m^i \cup \widetilde{x_m^i}$). The best fitness value is saved as $f_j^{position}$ which act as the target of followers. The value of k_1 is updated with Equation (2). Moreover, the position of leader and follower is updated using Equations (1) and (3). These processes except initialization and opposition of population are continued until the maximum iteration condition is reached. The final best solutions ($x_j^{fitness}$) are taken as the selected features. This strategy can improve the performance of traditional SSA.

B) SA-improved SSA

The final solutions ($x_j^{fitness}$) of ISSA are applied to SA algorithm to enhance the solutions of ISSA. In this manner, the SA approach acts as an internal local search agent for ISSA algorithm. Initially, SA algorithm sets the final solution ($x_j^{fitness}$) of ISSA algorithm as the current best solution (x_j^{best}). Then, mutation is employed in SA to generate a new neighborhood solution (x_m^i*) for current best solution (x_j^{best}). The fitness values are calculated for current best solution (x_j^{best}) and new neighborhood solution (x_m^i*). If the fitness of new neighborhood solution (x_m^i*) is less than the current best solution (x_j^{best}), then the best solution (x_j^{best}) will be updated with the new solution (x_m^i*). Otherwise, the final solutions are accepted with the acceptance probability calculation. The crossover operator is utilized in SA to accept the final solutions of ISSA ($x_j^{fitness}$) along with the new neighborhood solutions (x_m^i*). Thus, the hybridization of SA with ISSA can improve the exploitation ability of ISSA. Algorithm 1 explains the feature selection procedure using SA-improved SSA.

Algorithm 1. Feature selection using SA-Improved SSA

```
Initialize the positions (x_m^i)
Calculate the opposite positions (x~_m^i) using Equation (9)      //OBL strategy
x_m^i ← (x_m^i ∪ x~_m^i)
Calculate the fitness for each x_m^i
    while (t<max iterations)
    x_j^fitness ← best solution, f_j^position ←best fitness
    Update the value of k_1 using Equation (2)
        for every (x_m^i ∪ x~_m^i)
            if (x_m^i ∪ x~_m^i) is leader then
                Update the leader position using Equation (1)
            else
                Update the follower position using Equation (3)
            end if
        end for
    t=t+1
    end while
return x_j^fitness
```

while (t < max iterations) **// Simulated Annealing**

x_m^i *← Mutate ($x_j^{fitness}$)

Calculate fitness for $x_j^{fitness}$ and x_m^i *

if fitness (x_m^i *) < fitness ($x_j^{fitness}$)

then $x_j^{best} \leftarrow x_m^i$ *

else

Calculate the acceptance probability (P) using Equations (7), (8)

Generate a random number in [0,1]

if (Rand<=P)

x_m^i+ = crossover ($x_j^{fitness}$, x_m^i *)

$x_j^{best} \leftarrow x_m^i$+ (Accept the worst solution)

end if

end if

end while

return x_j^{best}

4.3 Voting Classifier based on PSO-improved SSA

After feature selection, the data partitioned for training purpose is applied to train the classifier. In the proposed work, a voting classifier based on PSO-improved SSA is utilized to classify the network traffic. The proposed model is the combination of five different ML classifiers like SVM, KNN, LR, DT and Bi-LSTM. The main goal of integrating PSO-improved SSA with voting classifier is to optimize the weights of each classifier.

After the initialization of ISSA, the fitness value is measured for its initial solutions. During fitness calculation, the weights of the base classifier are utilized to predict the outputs of test data. The error rate is calculated from the prediction accuracy which is set as objective function in fitness value calculation and weight optimization. The best solutions ($x_j^{fitness}$) are selected based on the fitness values. The value of k_1 is updated using Equation (2). Based on the velocity of PSO as shown in Equation (10), the velocity of follower is calculated as given in Equation (12).

$$v_j^{(t+1)} = w\, v_j^{(t)} + k_1 (x_j^{fitness} - f_j^{position}) \quad (12)$$

Here, $x_j^{fitness}$ is the best solutions selected based on the fitness value. $f_j^{position}$ represents the best fitness value which act as the target for followers. Then the leader position is updated using Equation (1). Based on Equations (9) and (3), the position of follower is updated as shown in Equation (13).

$$x_m^i = \frac{1}{3}(x_m^i + x_m^{i-1} + v_j^{(t+1)}) \quad (13)$$

These processes except initialization and opposition are continued until the maximum iteration condition is obtained. Thus, the weights of classifiers can be effectively optimized by PSO-improved SSA technique. The Algorithm 2 is the pseudocode for PSO-ISSA based weight optimization for voting classifier.

Algorithm 2. PSO-improved SSA for voting

Initialize the positions (x_m^i)

Calculate the opposite positions ($\widetilde{x_m^i}$) using Equation (9) **//OBL strategy**

$x_m^i \leftarrow (x_m^i \cup \widetilde{x_m^i})$

Calculate the fitness for each x_m^i

while (t<max iterations)

$x_j^{fitness} \leftarrow$ best solution, $f_j^{position} \leftarrow$best fitness

Update the value of k_1 using Equation (2)

for every x_m^i

Calculate the velocity of the follower using Equation (12) **//PSO**

if x_m^i is leader **then**
Update the leader position using Equation (1)
else
Update the follower position using Equation (13)
end if
end for
t=t+1
end while
return $x_j^{fitness}$

The optimized weights of the five classifiers are utilized by the voting classifier to predict the output of test data. The accuracy of each classifier is calculated with the predicted labels and text labels of each classifier. Then the predicted labels and the accuracy of each classifier are given to the voting classifier to generate final predicted labels. The voting classifier generates predicted labels according to the maximum accuracy of all five classifiers.

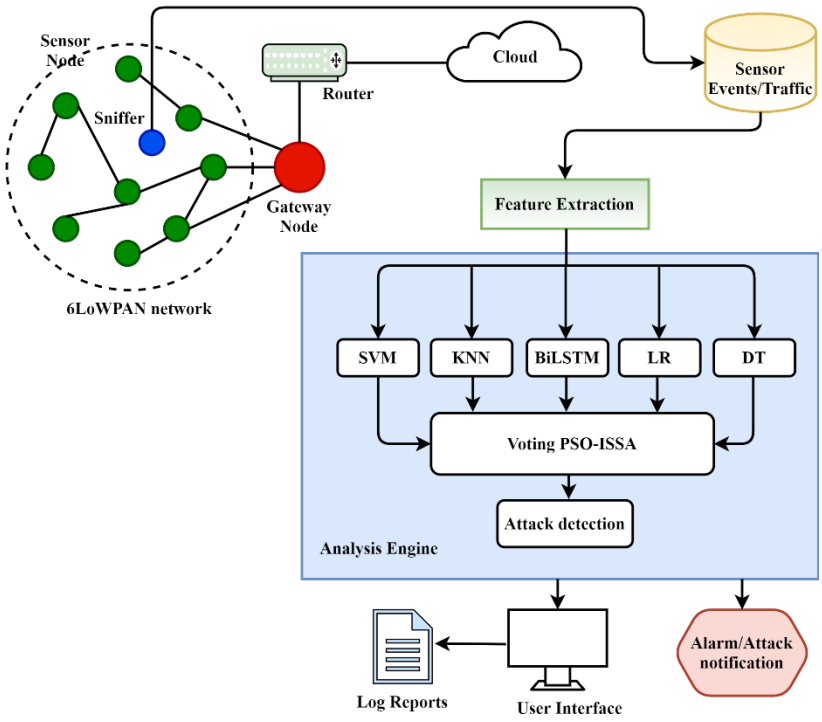

Figure 2. Architecture of proposed NIDS

The architecture of the proposed NIDS is shown in Figure 2. The architecture of the proposed work contains three units i.e., data collection unit, analysis unit and detection unit. The data collection unit consists of sensor events/traffic repository and sniffer. The sniffer is utilized to monitor all the packet transmissions within the 6LoWPAN network. It is directly connected to sensor events/ traffic collection repository that stores all the packet transmissions and sniffed sensor events in the form of packet traces. From the collected packet traces, the useful features can be extracted by using feature extraction process. The analysis unit is also named as analysis engine, which is the main part of the architecture. It has trained ensemble models that classify the traffic instances. A voting system utilizes the predictions to classify traffic into normal or attack based on the majority voting. The detection unit consists of alarm/attack notification module which receives the commands from analysis unit for raising alarm if an attack is detected. Besides, the analysis engine monitors the traffic regularly and sends information to the user interface to store all the information in the form of log reports. This paper is mainly focused on the performance improvement of analysis engine.

5. Experiment Results and Discussion

The experiments are performed on an HP laptop with Windows 10 operating system, Intel Core i3 processor having 2.3 GHz frequency, 4GB of RAM. The software used for the implementation and evaluation of the proposed framework is MATLAB R2020a. In this section, the performance of the proposed feature selection and classifier techniques is evaluated and compared with existing algorithms.

5.1 Dataset

RPL-NIDDS17 dataset is utilized to train the ensemble classifier in the proposed work. NetSim tool is used to create this synthetic dataset. NetSim is widely utilized for different network environment simulations i.e., FANET, MANET, IoT and VANET. The IoT network includes gateway, sensor nodes, wired node and router to create the dataset. All the information is saved in a separate CSV file for every attack. Finally, all the CSV files are combined to form single dataset. This dataset consists of 20 attributes with features of time, basic and flow type and two additional attributes for labelling. Moreover, it is comprised of one normal traffic pattern with seven routing attack patterns such as Sybil, blackhole, sinkhole, clone ID, local repair attacks, hello flooding and selective forwarding. In this dataset, the number of routing attack instances is 33,337 and the number of normal instances is 431,981. Thus, the dataset is imbalanced.

5.2 Dataset Balancing using SMOTE

Table 1. Dataset balancing using SMOTE

Category	Total instances in dataset	Utilized instances	With SMOTE
Attack	33,337	33,337	1,33,348
Normal	4,31,981	1,33,348	1,33,348

The full description of the RPL-NIDDS17 dataset is shown in Table 1. It shows that the number of normal instances utilized for classification is 1, 33,348. The number of attack instances utilized for classification is 33,337. This demonstrates that the number of instances in attack class is very less compared to the normal class. Thus, SMOTE based oversampling is done in the attack class to balance the dataset. The nearest neighbor parameter is set as $k = 4$ on the SMOTE algorithm to oversample the number of instances in the attack class equal to the normal class. After SMOTE based oversampling, the number of attack instances is increased to 1,33,348. That is, 40% of features are increased in the attack class (normal class = 4 * attack class). Table 2 shows the performance metrics of the proposed NIDS with and without SMOTE algorithm. In terms of accuracy, the proposed model shows higher accuracy of 96.4%, which is very less for the model without SMOTE. Moreover, precision, recall, F-measure and specificity of the proposed model with SMOTE algorithm are higher than the model without SMOTE. The error rate is very less for the model with SMOTE. Thus, the performance of the proposed NIDS is improved significantly with the integration of SMOTE.

Table 2. Performance of proposed NIDS with and without SMOTE

Parameters	With SMOTE	Without SMOTE
Accuracy	0.964	0.8884
Precision	0.9526	0.6929
ADR	0.9767	0.7939
F-measure	0.9645	0.7399
Specificity	0.9514	0.912
FAR	0.0486	0.088

5.3 Feature Selection

The dataset balanced through SMOTE technique is then applied for feature selection process via SA-ISSA.

Table 3. Parameter settings for feature selection algorithm

Algorithm	Parameters	Values
PSO	Inertia Weight W_{max}, W_{min}	[0.9, 0.6]
	Acceleration constants c_1, c_2	[2,2]
GA	Crossover rate	0.8
	Mutation rate	0.1
GWO	a	2 to 0
Proposed SA-ISSA	c_2, c_3	[1,0]
	current iteration, l	2
	Max iteration, I_{max}	50
	Initial temperature, T	0.1
	Number of search agents	20

The proposed algorithm for feature selection (SA-ISSA) is compared with other feature selection algorithms such as traditional SSA, OBL-SSA (ISSA), PSO, GA and GWO to show its performance in NIDS. Table 3 shows the values of optimization parameters assigned in the proposed and other optimization algorithms for feature selection experiments. The constant value is set for number of search (s= 20), maximum iterations 50 and current iteration (l=2) for the proposed and existing algorithms.

Table 4. Feature selection using SA-ISSA

Optimizer	PSO	GWO	GA	SSA	ISSA	SA- ISSA
Average error	0.1974	0.1689	0.1891	0.1553	0.1381	0.1356
Average Fitness	0.2556	0.2289	0.2159	0.2175	0.2013	0.2005
Best Fitness	0.1987	0.1322	0.1962	0.1378	0.1031	0.1014
Worst Fitness	0.2784	0.2473	0.2639	0.2047	0.2016	0.2009
Standard Fitness	0.0886	0.0299	0.0277	0.0283	0.0236	0.0228

Table 4 displays the performance evaluation of the proposed feature selection method with other feature selection methods. Other feature selection algorithms employed in the comparison are GA, PSO, GWO, traditional SSA and OBL-SSA (ISSA). Compared with other optimization algorithms, the proposed SA-improved SSA algorithm achieved minimum error in feature selection. This proves that the proposed algorithm selects proper features from the full dataset. It also shows that the lowest fitness value can be achieved by the proposed algorithm which is lower than other algorithms. Moreover, it can be observed that the best fitness value can be found by the proposed feature selection algorithm. Though the proposed algorithm has lowest standard deviation, it cannot find the worst fitness. KNN classifier is utilized in the fitness calculation with value of k=5.

5.4 Evaluation Metrics for Classification

The performance results of the proposed work have been measured with four basic classification metrics i.e., TN (true negatives), FN (false negatives), TP (true positives) and FP (false positives). The performance measures utilized in this paper include accuracy, precision, detection rate, specificity, F-measure, FPR, FNR and FAR.

- **TP:** It is the count of correctly detected attack instances.
- **TN:** It is the count of correctly detected normal instances.
- **FP:** It is the count of incorrectly detected attack instances.
- **FN:** It is the count of incorrectly detected normal instances.
- **Accuracy:** It measures the capability of the model to predict all the instances correctly as denoted in Equation (14). It is the count of correctly detected instances over the total detected in the test data.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{14}$$

- **Precision:** It is the count of correctly detected attack instances over the total attack instances in test data and it is computed as denoted by Equation (15).

$$\text{Precision} = \frac{TP}{TP+FP} \tag{15}$$

- **Attack Detection rate (ADR):** It is also called as sensitivity or recall. It calculates the capability of the attack detection as denoted in Equation (16). It is the number of correctly detected attack instances over classified attack instances.

$$\text{ADR} = \frac{TP}{TP+FN} \tag{16}$$

- **Specificity**: It is called as specificity or selectivity. It is the number of correctly detected normal instances over classified normal instances. It calculates the capability of the normal instance detection as denoted in Equation (17).

$$\text{Specificity} = \frac{TN}{TN+FP} \tag{17}$$

- **F-measure:** It is defined as the harmonic mean of ADR and Precision. It is also known as F-score and it is calculated as shown in Equation (18).

$$\text{F-measure} = 2 \times \frac{\text{Precision} \times \text{ADR}}{\text{Precision}+\text{ADR}} \tag{18}$$

- **FAR:** It is the average of false negative rate (FNR) and false positive rate (FPR) computed as denoted in Equation (19).

$$\text{FAR} = \frac{FPR+FNR}{2} \tag{19}$$

Here, FPR and FNR are calculated using Equations (20) and (21) respectively.

$$\text{FPR} = \frac{FP}{TN+FP} \tag{20}$$

$$\text{FNR} = \frac{FN}{FN+TP} \tag{21}$$

- **MCC:** Mathew correlation coefficient is calculated as given in Equation (22).

$$\text{MCC} = \frac{TP.TN - FP.FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{22}$$

- **Kappa:** It can be measured as given in Equation (23).

$$\text{Kappa} = \frac{\text{Accuracy}-\text{RA}}{1-\text{RA}} \tag{23}$$

Here, RA is random accuracy can be calculated as denoted in Equation (24).

$$\text{RA} = \frac{(\text{TN}+\text{FP})\times(\text{TN}+\text{FN})+(\text{FN}+\text{TP})\times(\text{FP}+\text{TP})}{\text{Total}\times\text{Total}} \tag{24}$$

5.5 Classification with PSO-ISSA

The proposed voting classifier is based on PSO-ISSA utilized to optimize the weights of the classifiers. After weight optimization, the predicted outputs are taken by the voting classifier. Based on maximum voting, it classifies the attacks in the RPL based IDS. To show the effectiveness of the proposed PSO- ISSA based voting classifier, comparison is performed with various optimization techniques. Comparison results are taken with various algorithms such as PSO, GA, GWO, traditional SSA and ISSA.

Table 5. Parameter settings for classification algorithms

Algorithm	Parameters	Values
PSO	Inertia Weight W_{max}, W_{min}	[0.9, 0.2]
	Acceleration constants c_1, c_2	[2,2]
GA	Crossover rate	0.8
	Mutation rate	0.1
GWO	a	2 to 0
Proposed PSO-ISSA	Maximum velocity, v_{max}	6
	Inertia Weight W_{max}, W_{min}	[0.9, 0.2]

The performance of the proposed voting classifier algorithm (PSO-ISSA) is compared with other voting classifier algorithms such as traditional SSA, OBL-SSA (ISSA), PSO, GA and GWO to prove the effectiveness of proposed voting classifier. Table 5 shows the values of optimization parameters assigned in the proposed and other optimization algorithms for feature selection experiments. The constant value is set for the proposed and other algorithms such as the number of search s= 20, maximum iterations = 20, current iteration l=2.

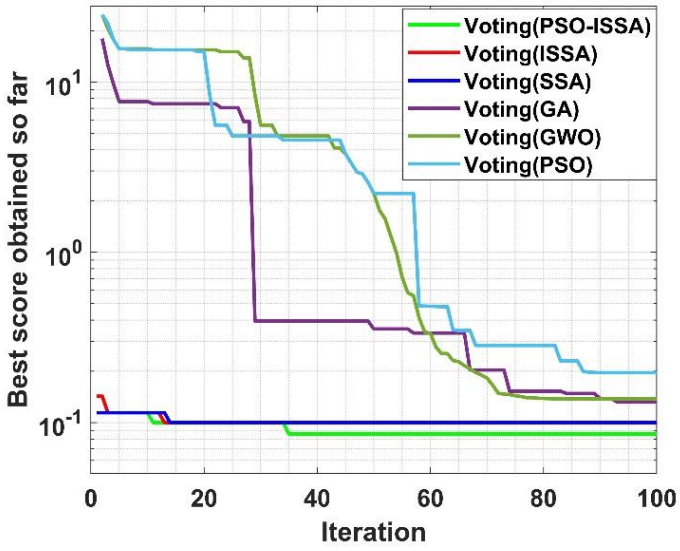

Figure 3. Convergence curve comparison for voting classifier algorithms

Figure 3 shows the convergence curve comparison of proposed voting classifier algorithm (PSO-ISSA) with other voting algorithms. From this, it can be observed that the convergence speed of proposed PSO-ISSA

algorithm is higher than other algorithms such as PSO, GA, GWO, traditional SSA and OBL-SSA (ISSA). The convergence rate of original SSA is improved with the integration of PSO and OBL strategy. The proposed PSO-ISSA based voting classifier is faster than other voting classifiers and gives better-optimized weights in a lesser time. Hence, the convergence graph proves that the proposed PSO-ISSA algorithm is more suitable for classification in NIDS.

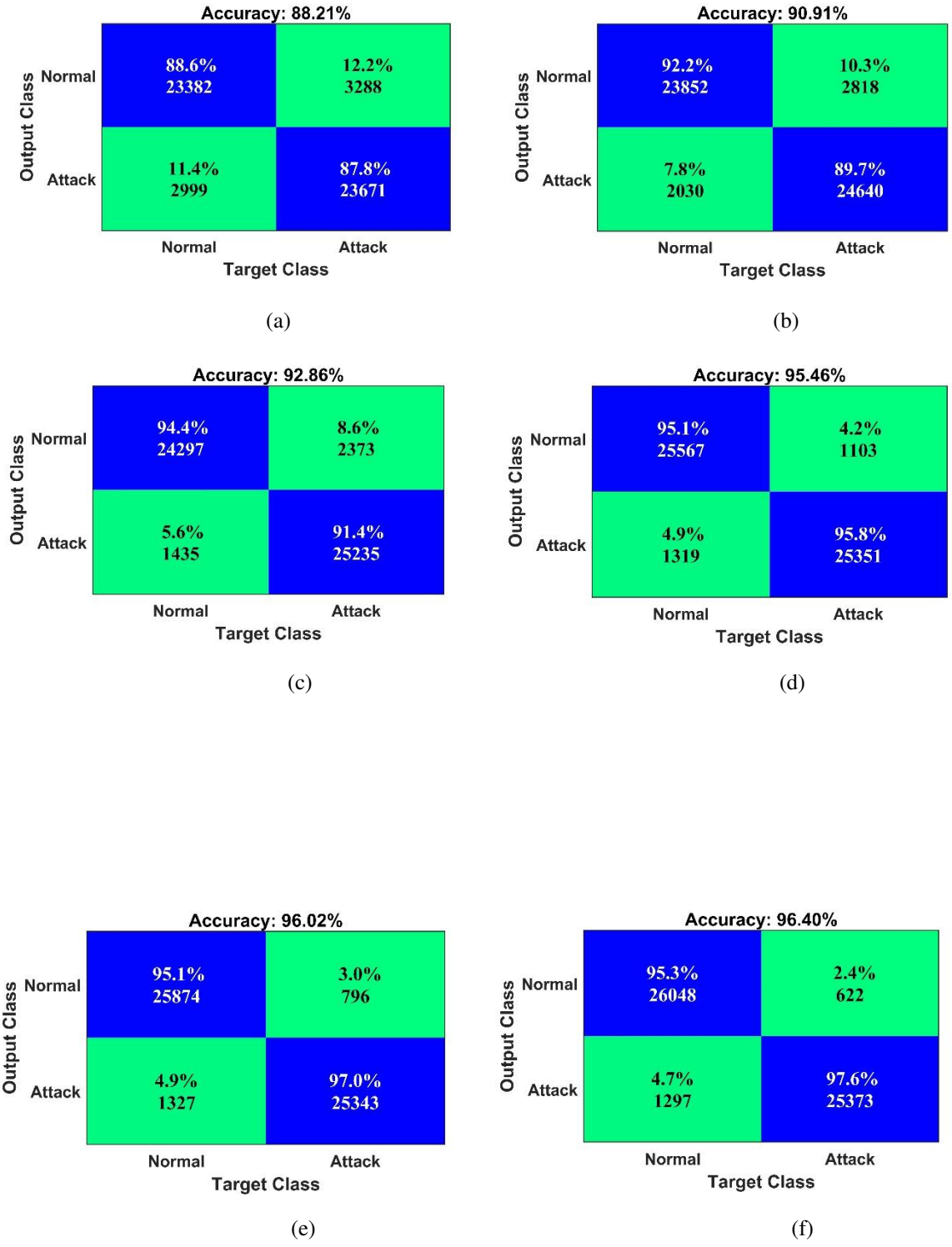

Figure 4. Confusion matrix for voting classifier of IDS (a) Voting-PSO, (b) Voting-GWO, (c) Voting-GA, (d) Voting-SSA, (e) Voting-ISSA, (f) Voting-PSO-ISSA

SVM, KNN, DT, LR and Bi-LSTM are the five ML classifiers utilized in the voting classifier system. Figure 4 (a), (b), (c), (d), (e) and (f) displays the confusion matrix of voting classifier with PSO, GWO, GA, SSA, ISSA and proposed PSO-ISSA algorithms respectively. From this, it can be observed that total 53340

instances are utilized for testing the voting classifier. All the classifiers except PSO based classifier accuracy is above 90%. In these classifiers, the proposed PSO-ISSA based voting classifier has higher accuracy of 96.4%. The voting classifier predicts the test data based on the maximum votes of ensemble classifiers so that it outperforms other classifiers.

Table 6. Performance comparison for voting classifier algorithms

Algorithm	Parameters						
	Precision	ADR	Specificity	F-measure	FAR	MCC	Kappa
PSO-ISSA	0.9526	0.9767	0.9514	0.9645	0.0359	0.9283	0.9280
ISSA	0.9512	0.9702	0.9502	0.9606	0.0398	0.9206	0.9204
SSA	0.9509	0.9586	0.9505	0.9548	0.0455	0.9092	0.9092
GA	0.9442	0.9110	0.9462	0.9273	0.0714	0.8577	0.8572
GWO	0.9216	0.8943	0.9239	0.9077	0.0909	0.8186	0.8182
PSO	0.8863	0.8767	0.8876	0.8815	0.1178	0.7643	0.7643

Table 6 shows the comparison results of voting classifier with various algorithms in terms of precision, ADR, specificity, F-measure, FAR, Mathew correlation coefficient (MCC) and Kappa. The precision rate, specificity and F-measure of all the voting classifiers except PSO based voting classifier is above 90%. However, the proposed PSO-ISSA based voting classifier achieved a higher value with 95.26% accuracy, 95.14% specificity and 96.45% F-measure. The attack detection rate of GWO and PSO based voting classifiers are less than 90%. The proposed classifier attains 97.67% attack detection rate which is higher than other voting classifiers. Moreover, MCC and Kappa values of all the classifiers show that the proposed classifier has better performance than other methods. Besides, the FAR of proposed IDS with PSO-ISSA based voting classifier is 3.6% which is very lesser than others. Thus, the PSO-ISSA based voting classifier outperforms other classifiers for the IDS in RPL based IoT networks.

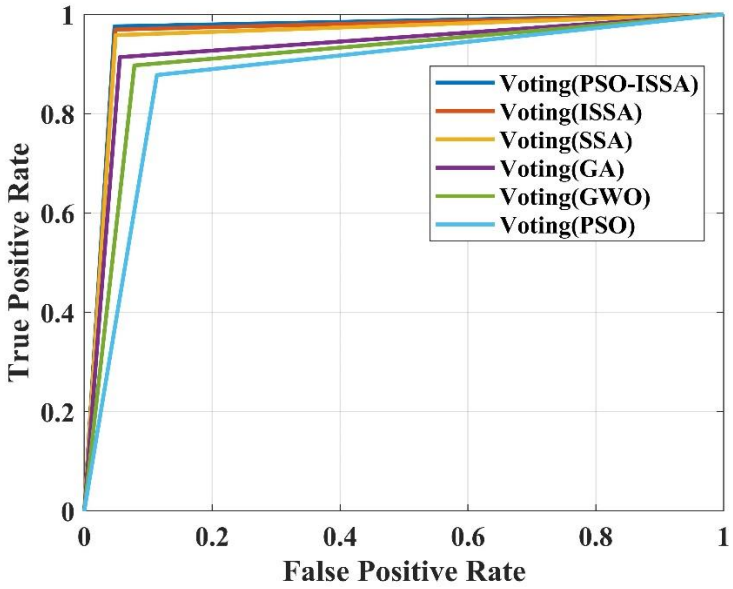

Figure 5. ROC curve comparison for Voting classifier in IDS

Figure 5 shows the comparison for voting classifier with proposed and other bio-inspired algorithms in terms of ROC curve. It shows that the voting system with proposed algorithm outperforms other algorithms. To prove the effectiveness of the proposed voting classifier, it is compared with other ensemble learning methods. Table 7 shows the comparison results of proposed voting classifier and other ensemble classifiers in terms of accuracy, ADR and FAR. The proposed voting classifier outperforms other ensemble techniques like majority voting, AdaBoost and bagging. From these results, it can be concluded that proposed NIDS is effective for seven types of attacks in the RPL based IoT networks.

Table 7. Performance comparison for ensemble learning techniques

Algorithm	Parameters		
	Accuracy	**ADR**	**FAR**
Proposed voting classifier (PSO-ISSA)	0.964	0.9767	0.0359
Majority voting	0.9525	0.9546	0.0497
AdaBoost	0.9263	0.911	0.0727
Bagging	0.8921	0.8667	0.1178

6. Conclusion

In this paper, a novel NIDS is proposed to combine the merits of both voting ensemble classifier and feature selection. The proposed NIDS can detect Sybil, blackhole, sinkhole, selective forwarding, local repair and hello flooding attacks. Two types of optimization are utilized in this paper. First, SA-ISSA technique is utilized to select the optimal best features to reduce the dimensionality and improve the computational complexity. Then, PSO-ISSA algorithm is utilized in the ensemble classifier to optimize the weights of base classifier. SVM, LR, DT, KNN and Bi-LSTM are the base classifiers utilized in the proposed voting ensemble classifier. RPL-NIDDS17 dataset is utilized to train the proposed NIDS model. The performance of the proposed approach is calculated and compared with existing algorithms for both feature selection and classification in terms of ADR, accuracy, F-measure, FAR and so on. From the experimental results, it can be known that the proposed voting ensemble classifier based on PSO-ISSA technique with SA-ISSA shows the best performance. Thus, the proposed NIDS is effective to detect the attacks in RPL based IoT networks.

Conflicts of InterestStatement

Manuscript title: **A Novel Intrusion Detection System for RPL Based IoT Networks with Bio-Inspired Feature Selection and Ensemble Classifier**

The authors whose names are listed immediately below certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

Author names:

1. **Mr. P. JAYA PRAKASH**
2. **Dr. B. LALITHA**

References

1. Verma, A., & Ranga, V. (2019). ELNIDS: Ensemble Learning based Network Intrusion Detection System for RPL based Internet of Things. 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU). doi:10.1109/iot-siu.2019.8777504.
2. Verma, A., & Ranga, V. (2020). Security of RPL based 6LoWPAN Networks in the Internet of Things: A Review. IEEE Sensors Journal, 1–1. doi:10.1109/jsen.2020.2973677.
3. Winter, T. (2012), Rpl: Ipv6 routing protocol for low-power and lossy networks. https://tools.ietf.org/html/rfc6550.
4. Verma, A., Ranga, V. (2019). Evaluation of Network Intrusion Detection Systems for RPL Based 6LoWPAN Networks in IoT. Wireless Pers Commun 108, 1571–1594. https://doi.org/10.1007/s11277-019-06485-w.
5. Sun, M., & Chen, T. (2010). Inventors; Inventec Corp, assignee. Network intrusion detection system. United States patent application US 12/411,916. September 30, 2010.
6. Wu, H., Schwab, S., & Peckham, R. L. (2008). Inventors; McAfee LLC, assignee. Signature based network intrusion detection system and method. United States patent US 7,424,744. September 9, 2008.
7. Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, 28(1–2), 18–28.
8. Verma, A., Ranga, V. Machine Learning Based Intrusion Detection Systems for IoT Applications. Wireless Pers Commun 111, 2287–2310 (2020). https://doi.org/10.1007/s11277-019-06986-8.
9. Tama, B. A., Comuzzi, M., & Rhee, K.-H. (2019). TSE-IDS: A Two-Stage Classifier Ensemble for Intelligent Anomaly-based Intrusion Detection System. IEEE Access, 1–1. doi:10.1109/access.2019.2928048
10. Cakir, S., Toklu, S., & Yalcin, N. (2020). RPL Attack Detection and Prevention in the Internet of Things Networks Using a GRU Based Deep Learning. IEEE Access, 8, 183678-183689. doi: 10.1109/ACCESS.2020.3029191.
11. Yavuz, F. Y., Ünal, D., & Gül, E. (2018). Deep learning for detection of routing attacks in the Internet of Things, Int. J. Comput. Intell. Syst., 12(1), 39-58.
12. Yang, J., Sheng, Y., & Wang, J. (2020). A GBDT-Paralleled Quadratic Ensemble Learning for Intrusion Detection System. IEEE Access, 8, 175467–175482. doi:10.1109/access.2020.3026044.
13. Bhati, B. S., Rai, C. S., Balamurugan, B., & Al-Turjman, F. (2020). An intrusion detection scheme based on the ensemble of discriminant classifiers. Computers & Electrical Engineering, 86, 106742. doi: 10.1016/j.compeleceng.2020.106742.
14. El-kenawy, E.-S. M., Ibrahim, A., Mirjalili, S., Eid, M. M., & Hussein, S. E. (2020). Novel Feature Selection and Voting Classifier Algorithms for COVID-19 Classification in CT Images. IEEE Access, 1–1. doi:10.1109/access.2020.3028012
15. Davahli, A., Shamsi, M. & Abaei, G. Hybridizing genetic algorithm and grey wolf optimizer to advance an intelligent and lightweight intrusion detection system for IoT wireless networks. J Ambient Intell Human Comput 11, 5581–5609 (2020). https://doi.org/10.1007/s12652-020-01919-x
16. Zhao, X., Yang, F., Han, Y., & Cui, Y. (2020). An Opposition-based Chaotic Salp Swarm Algorithm for Global Optimization. IEEE Access, 1–1. doi:10.1109/access.2020.2976101
17. Tubishat, M., Idris, N., Shuib, L., Abushariah, M. A. M., & Mirjalili, S. (2019). nImproved salp swarm algorithm based on opposition based learning and novel local search algorithm for feature selection. Expert Systems with Applications, 113122. doi: 10.1016/j.eswa.2019.113122
18. Eberhart, R., & Kennedy, J. (n.d.). A new optimizer using particle swarm theory. MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science. doi:10.1109/mhs.1995.494215.
19. Verma, A., & Ranga, V. (2018). RPL-NIDDS17- A Data set for Intrusion Detection in RPL based 6LoWPAN Networks (Internet of Things). https://doi.org/10.5281/zenodo.1406034.
20. Feng, W., Dauphin, G., Huang, W., Quan, Y., Bao, W., Wu, M., & Li, Q. (2019). Dynamic Synthetic Minority Over-Sampling Technique-Based Rotation Forest for the Classification of Imbalanced Hyperspectral Data. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 1–11. doi:10.1109/jstars.2019.2922297
21. Kirkpatrick, S., Gelatt, C. D., Jr, & Vecchi, M. P. (1983). Optimization by simulated annealing, Sci., 220(4598), 671-680.
22. Jia, H., Li, J., Song, W., Peng, X., Lang, C., & Li, Y. (2019). Spotted Hyena Optimization Algorithm With Simulated Annealing for Feature Selection. IEEE Access, 7, 71943–71962. doi:10.1109/access.2019.2919991

23. Al-Abassi, A., Karimipour, H., Dehghantanha, A., & Parizi, R. M. (2020). An Ensemble Deep Learning-based Cyber-Attack Detection in Industrial Control System. IEEE Access, 1–1. doi:10.1109/access.2020.2992249
24. Shahraki, A., Abbasi, M., & Haugen, Ø. (2020). Boosting algorithms for network intrusion detection: A comparative evaluation of Real AdaBoost, Gentle AdaBoost and Modest AdaBoost. Engineering Applications of Artificial Intelligence, 94, 103770. doi:10.1016/j.engappai.2020.103770
25. Kasongo, S.M., Sun, Y. Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset. J Big Data 7, 105 (2020). https://doi.org/10.1186/s40537-020-00379-6.
26. Zhou, Y., Cheng, G., Jiang, S., & Dai, M. (2020). Building an Efficient Intrusion Detection System Based on Feature Selection and Ensemble Classifier. Computer Networks, 107247. doi: 10.1016/j.comnet.2020.107247.
27. Asadi, M., Jamali, M. A. J., Parsa, S., & Majidnezhad, V. (2020). Detecting botnet by using particle swarm optimization algorithm based on voting system. Future Generation Computer Systems. doi:10.1016/j.future.2020.01.055.
28. Tama, B. A., Comuzzi, M., & Rhee, K.-H. (2019). TSE-IDS: A Two-Stage Classifier Ensemble for Intelligent Anomaly-based Intrusion Detection System. IEEE Access, 1–1. doi:10.1109/access.2019.2928048
29. Kumar, P., Gupta, G. P., Tripathi, R. (2021). An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks, Computer Communications, 166, 110-124, https://doi.org/10.1016/j.comcom.2020.12.003.
30. Pu, C. (2020). Sybil Attack in RPL-Based Internet of Things: Analysis and Defenses. IEEE Internet of Things Journal, 1–1. doi:10.1109/jiot.2020.2971463
31. Murali, S., & Jamalipour, A. (2019). A Lightweight Intrusion Detection for Sybil Attack under Mobile RPL in the Internet of Things. IEEE Internet of Things Journal, 1–1. doi:10.1109/jiot.2019.2948149
32. Gothawal, D.B., Nagaraj, S.V. Anomaly-Based Intrusion Detection System in RPL by Applying Stochastic and Evolutionary Game Models over IoT Environment. Wireless Pers Commun 110, 1323–1344 (2020). https://doi.org/10.1007/s11277-019-06789-x

Figures

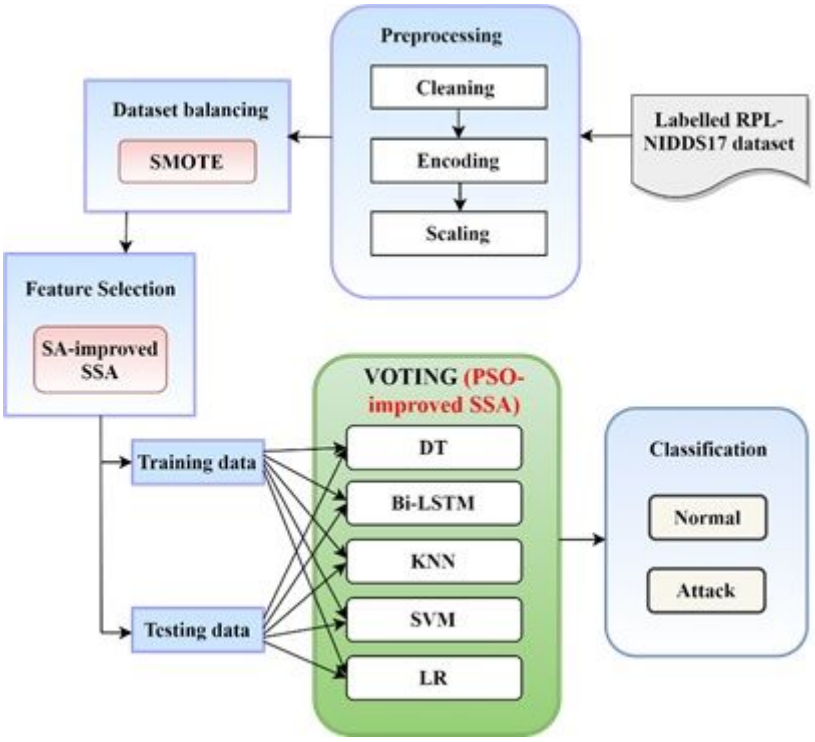

Figure 1

Block diagram of proposed methodology for NIDS

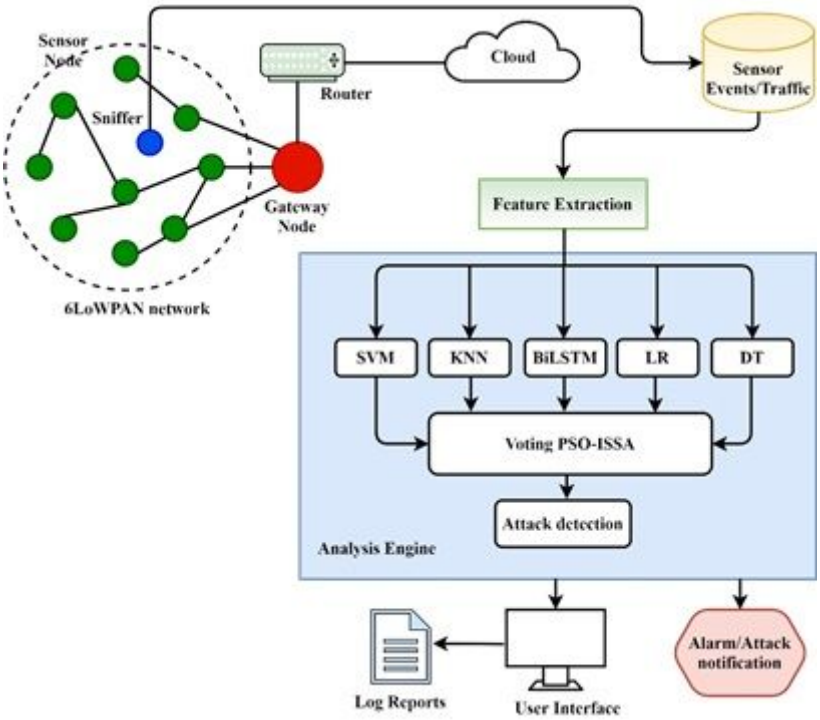

Figure 2

Architecture of proposed NIDS

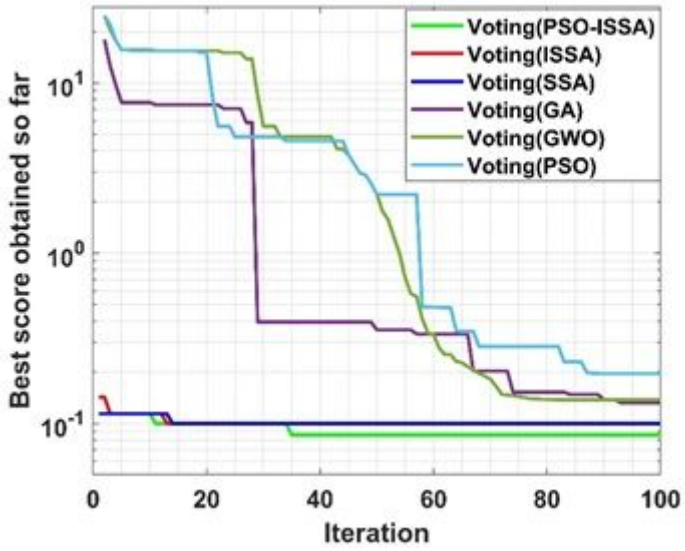

Figure 3

Convergence curve comparison for voting classifier algorithms

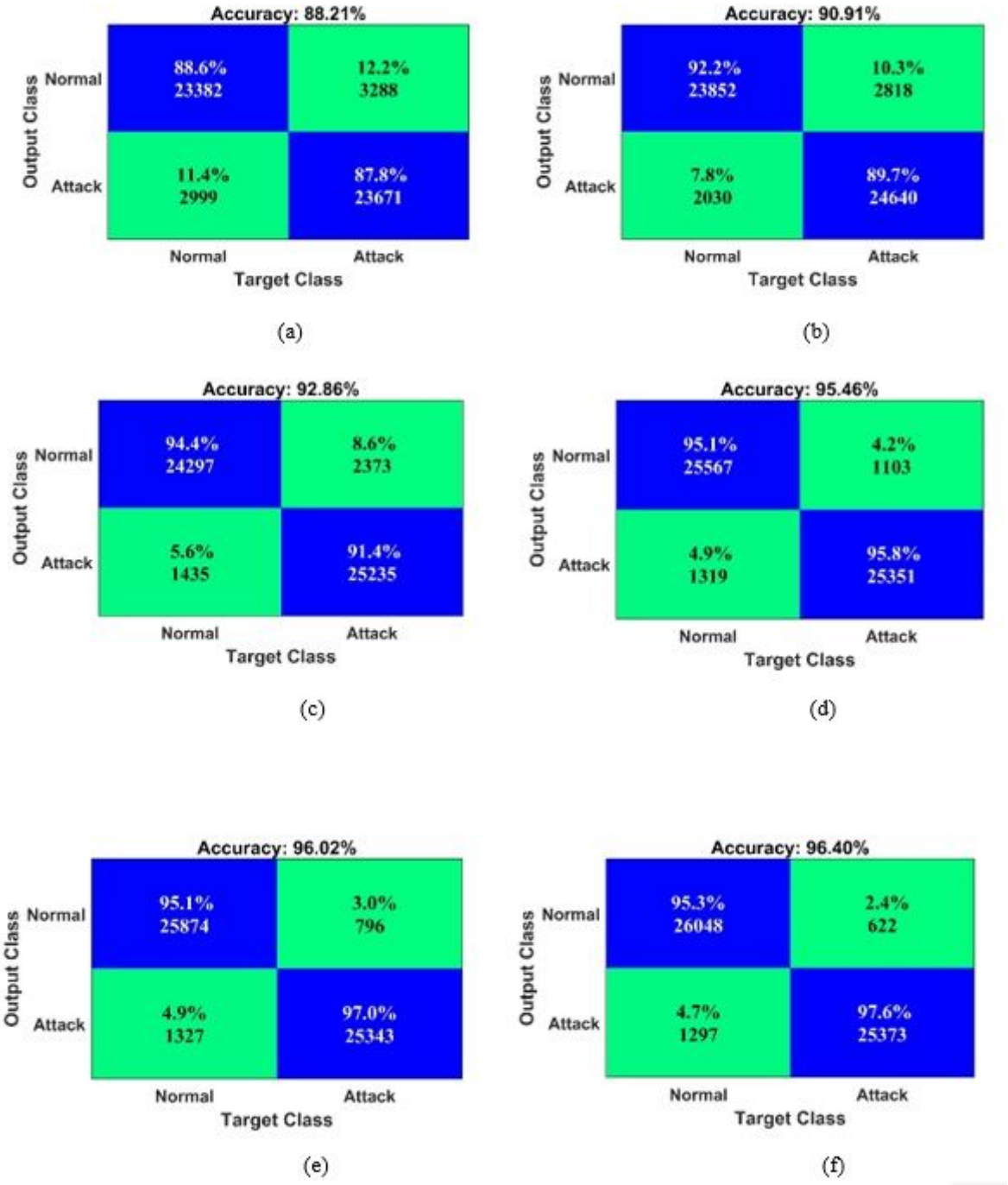

Figure 4

Confusion matrix for voting classifier of IDS (a) Voting-PSO, (b) Voting-GWO, (c) Voting-GA, (d) Voting-SSA, (e) Voting-ISSA, (f) Voting-PSO-ISSA

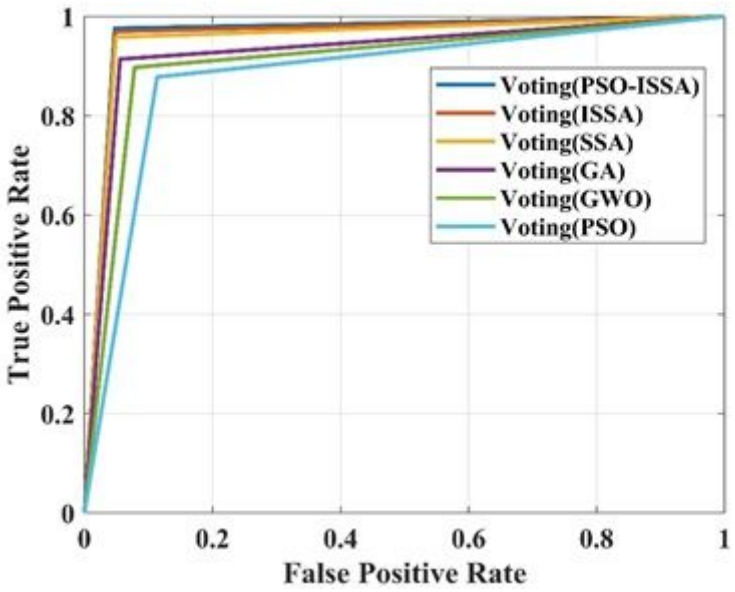

Figure 5

ROC curve comparison for Voting classifier in IDS