Wireless Personal Communications (2022) 127:1763–1785
https://doi.org/10.1007/s11277-021-08721-8

Biological Feature Selection and Classification Techniques for Intrusion Detection on BAT

Satheesh Narayanasami[1] · Sudhakar Sengan[2] · Saira Khurram[3] · Farrukh Arslan[4] · Suresh Kumar Murugaiyan[5] · Regin Rajan[6] · Vijayakumar Peroumal[7] · Anil Kumar Dubey[8] · Sujatha Srinivasan[9] · Dilip Kumar Sharma[10]

Accepted: 29 June 2021 / Published online: 9 July 2021

Abstract
Privacy is a significant problem in communications networks. As a factor, trustworthy knowledge sharing in computer networks is essential. Intrusion Detection Systems consist of security tools frequently used in communication networks to monitor, detect, and effectively respond to abnormal network activity. We integrate current technologies in this paper to develop an anomaly-based Intrusion Detection System. Machine Learning methods have progressively featured to enhance intelligent Anomaly Detection Systems capable of identifying new attacks. Thus, this evidence demonstrates a novel approach for intrusion detection introduced by training an artificial neural network with an optimized Bat algorithm. An essential task of an Intrusion Detection System is to maintain the highest quality and eliminate irrelevant characteristics from the attack. The recommended BAT algorithm is used to select the 41 best features to address this problem. Machine Learning based SVM classifier is used for identifying the False Detection Rate. The design is being verified using the KDD99 dataset benchmark. Our solution optimizes the standard SVM classifier. We attain optimal measures for abnormal behavior, including 97.2 %, the attack detection rate is 97.40 %, and a false-positive rate of 0.029 %.

Keywords Intrusion detection system · Dataset · Bat algorithm · Optimal features · SVM

1 Introduction

One of the significant challenges is network security because of the tremendous evolution of information technology. Big data is continuously receiving attackers and is therefore susceptible to external network intrusion. When an intruder sends malicious packets to the host machine or requires a vulnerable network to access or manipulate sensitive data, this is known as an intrusion. Protection protocols may be applied on a network to minimize the number of intruders. Unauthorized people cannot access this service through these devices. Attackers use various approaches to find the weakness in a network's security, and the

✉ Sudhakar Sengan
sudhasengan@gmail.com

Extended author information available on the last page of the article

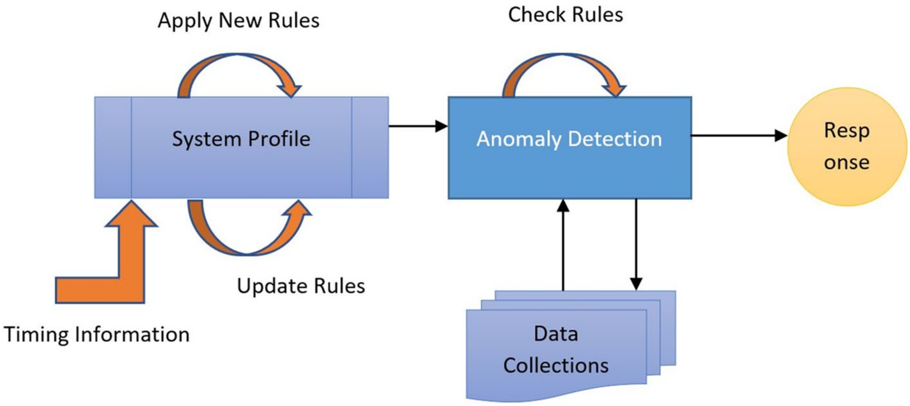


Fig. 1 Intrusion detection system setup

method used to identify and track this malicious behavior in a network is intrusion detection [1]. It is quite challenging to detect the network manually. The IDS system was therefore designed to carry out the work automatically, observe network and device operations to identify fraudulent behaviors. It can be a Network-based Intrusion Detection System (NIDS) and Host-based Intrusion Detection Systems (HIDS) [2]. HIDS detects abnormalities in a computer system. NIDS would be used to identify network system abnormalities. Network-based IDS are classified into two types: signature-based (or) misuse-based NIDS and anomaly-based NIDS. Signature-based NIDS identifies a threat by comparing a signature that has already been saved in the signature database with the received data packet. A signature is described as the established attack pattern or rule. However, unidentified assaults can't be detected. On either end, anomaly-based NIDS detects a new assault by identifying the customer's usual system behavior.

A slight difference between the observed occurrence and the normal activity is invasive. The drawback of anomaly-based NIDS is that normal behavior is complicated to construct due to the diversity of internet traffic. The Intrusion Detection System (IDS), as shown in Fig. 1, has become a key component of security architecture. The proposed system describes intrusion detection and identification. The intrusion framework usually handles a massive amount of data; one of the critical tasks of IDS is to preserve the highest value of features that display the entire data and delete redundant information. Feature selection decreases the number of features from the noisy dataset. This relevant function subset increases the detection rate. Selection features are classified into (a) Filter, (b) Wrapper, and (c) Hybrid approach [3].

Feature selection aims to reduce classification time and improve accuracy rate. They classified the datasets using the current scheme, which incorporates all of the dataset's attributes. For complex problems, the Bat algorithm is used. Sigmoid and tan hyperbolic functions were commonly used to solve non-continuous issues. The current algorithm selects features using the Bat Algorithm. That is the motion of the Bat in a d-dimensional binary space. As a result, a bat's location is described as a vector of binary coordinates, and the bat may traverse the hyper cube's corners. In each iteration step, the transformed values for the attribute subsets will be changed to ensure that the bat continues to travel to the appropriate location. They use knowledge to pick features in the current framework and the bat algorithm to update the SVM [4] regularization and kernel parameters. If the bat cannot find a better performance value within a pre-defined number of samples, use the Bat algorithm's original global solution. However, it slightly improves the algorithm's execution. SVM is assigned the selected attributes to review for classification accuracy.

The filter method requires correlation to classify its characteristics and doesn't rely on the classifier. Wrapper methods, on the other hand, are entirely reliant on the classifier. The actual application is to analyze and monitor system vulnerabilities. This will be more effective in identifying abnormality activities and user tracking policy. IDS system ensures secured web services along with file integrity. Our framework employed the wrapper approach. In this paper, the author uses a comparatively recent hybrid method, the Bat Algorithm, to improve the SVM classifier, providing significant improvement. Consequently, we propose a new adaptive method that incorporates Bat Algorithm and demonstrates experimentally that this outperforms the conventional BAT when combined with SVM.

2 Related Works

Intrusions may be identified as explicit or implicit. Secondary intrusions are triggered by authorized or unauthorized persons from outside the network into the network's surface. Primary intrusions are conducted out by authorized individuals within the network and the internal network. Attackers commonly compromise computer systems through software defects, password cracking, traffic flows collisions, and performance issues in networks, utilities, or network devices [5]. The application of a distinguished supervised learning algorithm fetches information that is curious to design an IDS. This paves the way for an easy and competent intrusion detection method susceptible to quick acclimatization by anyone. Many current Machine Learning methods like Decision Tree, Neural Network, Back-Propagation NN, Naïve Bayesian, and Bayesian Network for network data classification are used to conduct the comparative study.

IDS' research area focuses on the development of machine learning algorithms. Various machine learning algorithms have been developed over the past years to deal with noisy data and detect new attacks with a low false-positive rate, including neural networks, genetic algorithms, and decision trees [6]. According to filter-based feature selection, it can manage data features that are sequential and nonlinear. SVM classifier is used for sampling.

An evolutionary algorithm is used to select features. They created Particle Swarm Optimization (PSO) [7] for selecting features and performed classification using ensembles of tree-based classifiers. PSO is a technique proposed for detecting intrusion. They pick features using a genetic algorithm and use Adaptive Mutation to achieve gradual convergence. A hybrid algorithm that integrates modified Artificial Bee Colony with Enhanced Particle Swarm Optimization is used to achieve the best result. The method of tenfold cross-validation can be used for classification. They use the KDDCup'99 [8] benchmark dataset to assess the efficiency of this work.

The author invented a new detection model. They pick features using Binary Particle Swarm Optimization and validate the results using SVM and C4.5 classifiers. In comparison with PSO, BPSO achieves superior performance. The author suggested an ensemble classifier as a hybrid of SVM and K-Nearest Neighbors [9]. PSO searching returns a subset of features, and an ensemble classifier identifies the assault.

The Dynamic Membrane-driven Bat Algorithm (DMBA) method aims to enhance resident diversity through tradeoff where the static membrane methods in DMBA would be dynamically involved by integration and separation rules that help maintain the diversity of the population. This method is used to cover the classification depending on the SVM classifier used in many areas like disease diagnostics, face recognition, text recognition, plant disease identification, sentiment analysis, and IDS for network security applications.

Additionally, they use the KDDCup'99 benchmark dataset. For selecting features, the Binary Bat Algorithm is suggested. The bat's location is described using a vector of binary coordinates. The bat has traveled across d-dimensional binary space. They validate their method using two datasets: cancer and iris [10].

Techniques for Feature Selection in High-Dimensional Data. The primary issue is to improve the optimization efficiency to solve various optimization issues and advantages of various dynamic membranes computing structures. To extend technologies and develop more reliable data crowned, there is a need to process many data sets. The approaches for selecting the features describe how the features are incorporated during the evaluation process, namely feature subset-based and feature k-means, according to the machine learning algorithm used, namely wrapper, embedded, hybrid, and filter [11]. It is demonstrated that feature ranking-based methods are more effective in system memory and highly computational than subset-based methods and that k-means methodologies do not reduce redundancy.

3 Proposed Model

The proposed framework selects features using the bat algorithm. To identify malicious activity and enhance classification accuracy, specific characteristics would be integrated into the SVM classification algorithms [12]. Even in high-dimensional noisy datasets, SVM provides excellent generalization and avoidance of local minima, as well as good precision. Figure 2 depicts our system framework focused on the Bat algorithm, while Fig. 2 describes the system selection process. The Feature selection function, in this case, seeks to extract the most critical data from a sample set of features. Given that this task can be

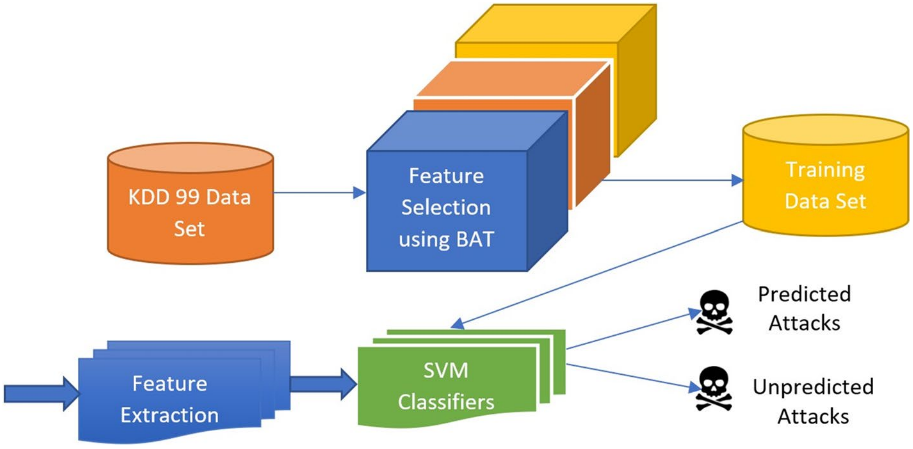


Fig. 2 Proposed architecture

viewed as an optimization problem, the combinational growth of potential solutions can make an efficient detection impractical [13].

3.1 Bat Algorithm

Yang was influenced by microbats that use echolocation. Bats emit a loud sound pulse to locate prey or obstacles. As a result, he created the Bat Algorithm in 2010 [14]. The Bat flies randomly in search of its game. The complete algorithm is shown in the flow chart in Fig. 3. Three classification principles have been identified to evaluate an intelligent bat algorithm:

Step 1 Directional microphones are used by all bats to measure the contrast between a threat and victims and to detect range.

Step 2 Bats move randomly, and their action is identified by their spatial position (x_i) and speed (v_i). These values are measured to search for threats using a changing wavelength (), frequency (freqmin), and loudness (A_0). Consequently, bats can modify the frequency of their emitted pulses and the speed during which they transmit vibrations (r [0,1]) to the distance of their objective [15].

Step 3 The loudness can change in many different ways to presume its ranges between a considerable value (A_0) and a consistent minimum value (A_{min}).

3.2 Mathematical Model of Bat

Step 1 The formula determines the hyperplane.

$$f_{bat}(X) = sf(x) + Matrix_b \tag{1}$$

The formula determines the hyperplane.
whereby *sf* denotes the scaling factor and $Matrix_b$ represents the bias matrix.
Step 2 By transforming the dataset into a higher-dimensional feature vector, we can change the nonlinear SVM to a linear problem through kernel functions. We are using SVM with Radial Basis Function (RBF) for testing prototype, with the appropriate RBF kernel [16]:

$$Kernal(X_{i,}X) = Exp\left[-\frac{1}{2\sigma^2}(X_i - X^2)\right] \tag{2}$$

Step 3 Bat algorithm is a particle swarm algorithm that encompasses a network of sensors to conduct searches. Bat algorithm can find the best C and σ based on the SVM's reliability while determining SVM threshold limits. Each agent has a current position,

$$X_i = (X_{i,1}, X_{i,2}, ..., X_{i,Dim})T, A \tag{3}$$

Step 4 Current flying velocity,

$$Velocity_i = (Velocity_{i,1}, Velocity_{i,2}, ..., Velocity_{i,Dim})Time \tag{4}$$

where Dim is the problem dimension. Each agent alerts its direction and speed according to the given equation to determine the best position:

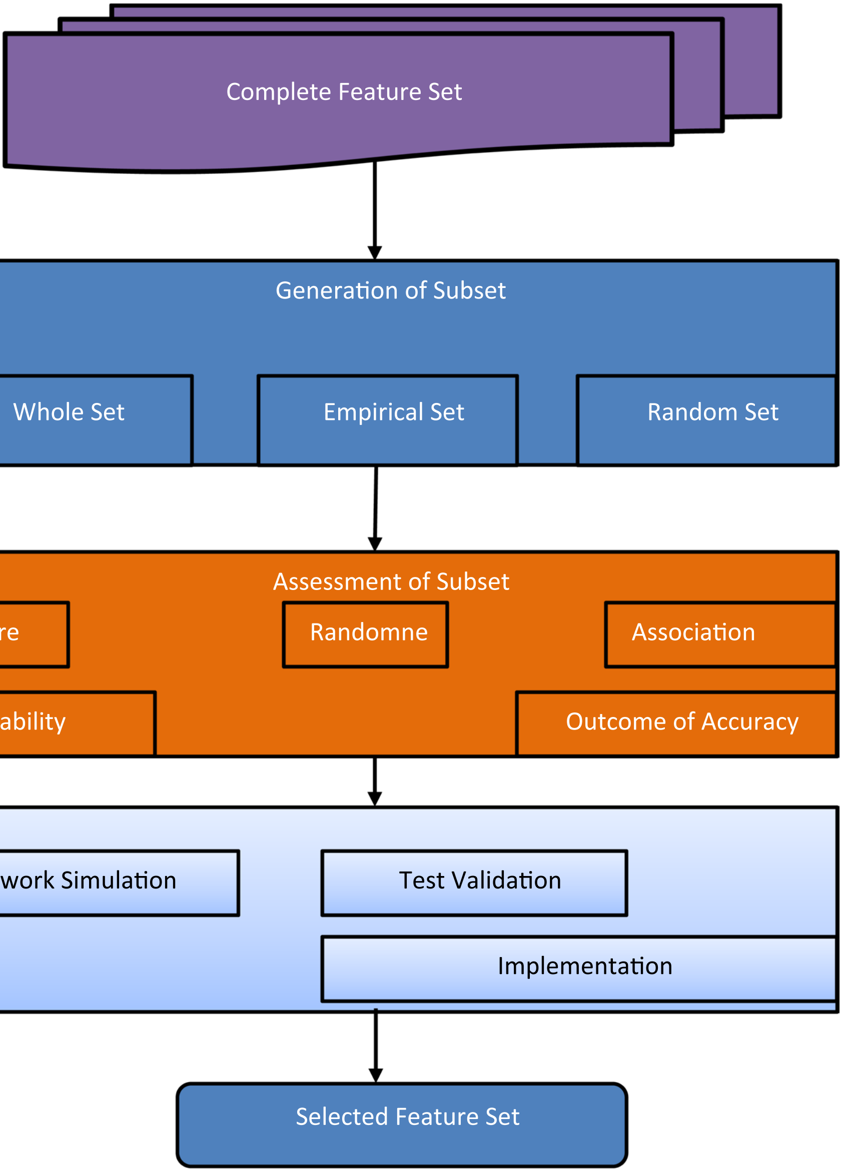


Fig. 3 The proposed framework of the feature selection process

Step 5

$$Freq_i = Freq_{Minimum} + (Freq_{Maximun} - Freq_{Minimum}).\beta \tag{5}$$

Step 6

$$Velocity_{i,j}^{T} = Velocity_{i,j}^{T-1} + (X_{i,j}^{T-1} - XBest_j).Freq_i \tag{6}$$

Step 7

$$X_{i,j}^{T} = X_{i,j}^{T} + Velocity_{i,j}^{T} \tag{7}$$

where βϵ[0,1] is a randomly generated vector. XBest is the group's optimal solution. Only the fitness function decides the solution's consistency. The fitness function used in this model is the precision of the SVM after it is trained on the dataset described by the bat's location. When a bat approaches a point, its Loudness (Li) decreases, and its pulse emission (PEi) rate increases [17].

Step 8

$$L_i^{T+1} = \alpha.L_i^{T} \tag{8}$$

Step 9

$$PE_i^{T+1} = PE_i^{0}.[1 - e^{-1\gamma.T}] \tag{9}$$

where α (0 < α < 1) and γ (γ > 0) are constant values. Yang implements uniform random walks to enhance the exploration dimension in S space:

Step 10

$$X_{fresh} = X_{\Pr evious} + \delta.PE_{Time}^{*} \tag{10}$$

Where δ ϵ [-1,1], random number PE*Time; average loudness of all bats.

In some instances, BA is similar to the prominent PSO. The particle's position characterizes the method; each swarm member has its velocity and function and based on their fitness value, they have been updated in real-time. Some other notable differences exist since BA utilizes the approach through using random walks and adjusting the loudness and pulse rate. For PSO, extraction is regulated by regional and specific best methods, while discovery is managed using two learning criteria [18].

3.3 Bat Feature Selection

To increase the accuracy of the classifier, the feature selection of Bat attempted to enhance the feature subset from each step. The essential elements in this optimization algorithm are [19] [20]:

- Each bat has a location that signifies a subclass of all its features. The bat practices and examines the SVM classifier using this element.
- After examining certain bats, the swarm's optimal solutions fitness value is determined.

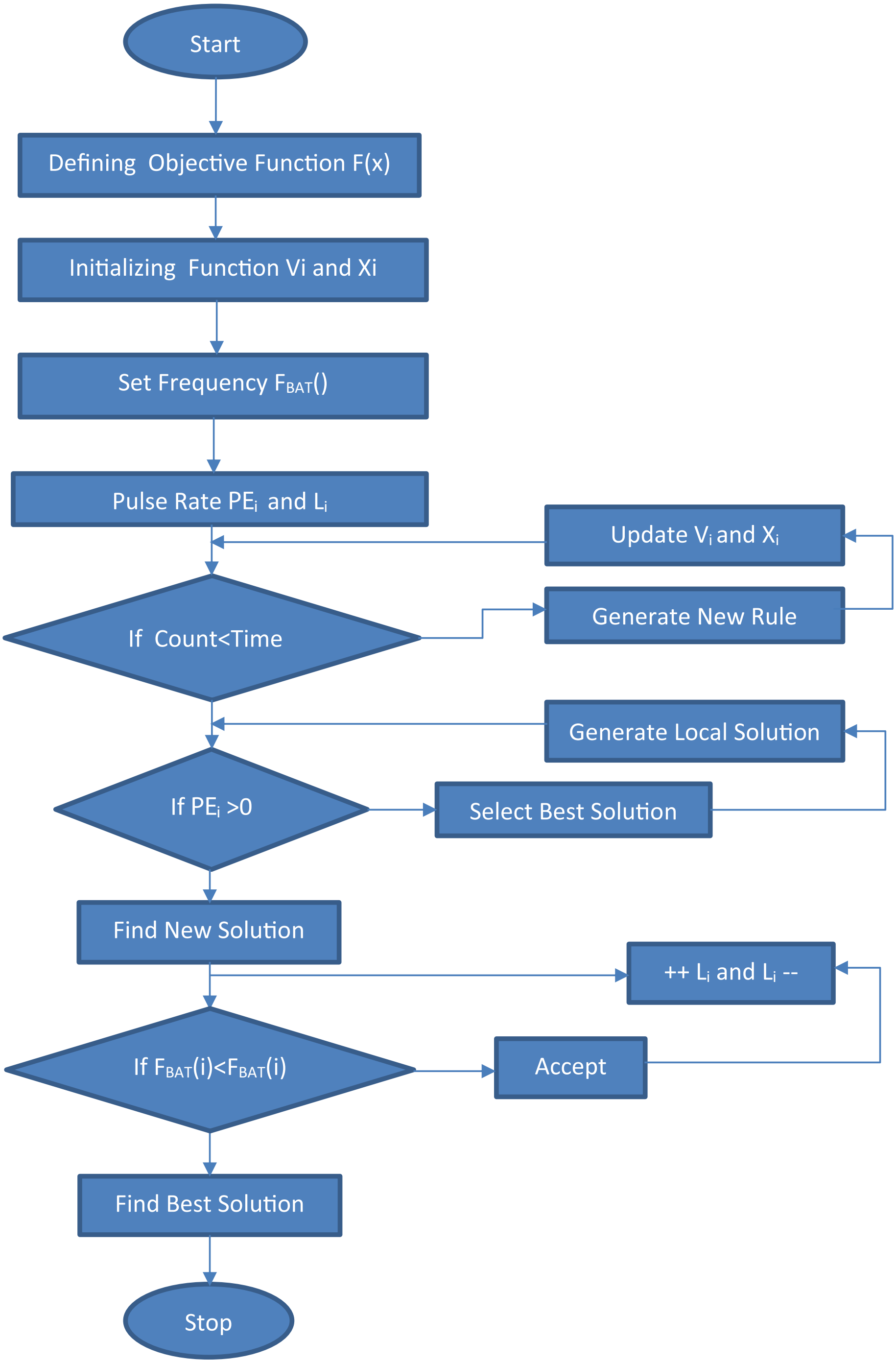


Fig. 4 Flowchart of bat algorithm

- While reaching the optimization algorithm, each bat improves its location and pulse rate, and frequency.
- The solution is randomized by the bat using Levy flights such that it performs the computation, and thus S(X_{ij}) will be the sigmoid function.
- The frequency and velocity of the bat will be updated in case there is no significant improvement in the fitness value.
- Finally, an increase in pulse rate and decrease in loudness of the bat has been experienced in case the new fitness value outperforms the global best. There is a modification in the global optimum.

3.4 The General Bat Algorithm for Feature Selection

The below-mentioned algorithm indicates the Feature Set selection by using the Bat system (Fig. 4).

```
Step 1    Input: PS (Network Size), Maximum Threshold.
Step 2    Output: $f_{Bat}(X_{Best})$ solution.

.
    Define pulse frequency, rates, and loudness.
    Initialize the $f_{Bat}(x_i)$ and position $x_i$.

Step 3    While Time < Maximum Threshold Do.
Step 4    For Each $X_{Bat}$ i to PSO Do.

.
    Generate New and Optimized Solution.

Step 5    If rand > $L_{i,}$ Then.

.
    Determine the $X_{Best}$ solution.
    Estimate local search.

Step 6    End If.
Step 7    If L &&< $L_i$ &&$f_{Bat}(X_i)$ < f(X_Fresh) Then.

.
    Accept new solution –$L_i$ && ++ $L_i$.

Step 8    End If.
Step 9    End For.

.
    Time = Time + 1.
```

Table 1 Set of Features

No	List of the feature	Feature configuration	Type
1.	**TIME LIMIT**	Link Interval	Constant
2.	**FORM OF PROTOCOL**	TCP and UDP	Disconnected
3.	**SERVICE**	TELNET and FTP	Disconnected
4.	**FLAG**	Link Status FLAG	Disconnected
5.	**SOURCE NODE DATA**	Bits of Data Transferred from the SRC-TO-DES	Constant
6.	**TARGET NODE DATA**	Bytes Sent to SRC from DES	Constant
7.	**LAND**	‘1; If Server◇Linked Else Data Broadcast	Disconnected
8.	**WRONG_FRAGMENT**	Number of FALSE Bits	Constant
9.	**URGENT**	Size of Priority Packets	Constant
10.	**HOT**	High Proportion of “HOT” Procedures	Constant
11.	**ATTEMPT_FAILED_LOGINS**	Numbers of Failed User Authentication	Constant
12.	**LOG_INNPUT**	If user = successful, 1; otherwise, 0.	Disconnected
13.	**NUMBER_COOPERATED**	SET of “vulnerable” Factors	Constant
14.	**ORIGIN_NODE**	If Root Node Attained, 1 Returned; Otherwise, 0 Returned.	Constant
15.	**SU_ROOT_ENDEAVORED**	If SU ROOT = Attempted, 1 Returned; Otherwise, 0 Produced	Constant
16.	**ROOT_NUMBER**	% of “ROOT” Access Rights	Constant
17.	**FILE_CREATION_LOG**	Size of File	Constant
18.	**ITRATION_SHELLS_LOG**	Command Triggers	Constant
19.	**ACCESS_FILES_ITRATION**	Operations Performed on User Access Data	Constant
20.	**OUTBOUND_CMDS_ITRATION**	Range of Outbound Signals, FTP REQ	Constant
21.	**IS_THREAT_LOGIN**	If Login HOT list, assign = 1 ; Else, Assign = 0	Disconnected
22.	**IS_VISITOR NODE_LOGIN**	If Login HOST, 1 Returned; Else, 0 Restored.	Discrete
23.	**COUNTER_0**	End of 2 s., Limited Range of Links to the Identical Network as TCP	Constant
24.	**LINK_COUNTER**	NO. Links to the same Provider as the Connection Type in the Prior 2 s.	Constant
25.	**ERROR_PROPORTION**	% Links with Spoofed Failures	Constant
26.	**ERVICE_ERROR_LOG**	% SYN_ERROR Links	Constant
27.	**REJECTION_ERROR_LOG**	% REJECT_ERROR Links	Constant

Table 1 (continued)

No	List of the feature	Feature configuration	Type
28.	**SERVICE_ERROR_PROPORTION**	% of "REJECT_ERROR" Links	Constant
29.	**SERVICE PROVIDER_PROPORTION**	% Links to the Identical Provider	Constant
30.	**DIFFERENT_SERVICE_PROPORTION**	% Links to Different Services	Constant
31.	**MULTI_HOP_PROPORTION**	% Links to Multi-Hop Networks	Constant
32.	**DESTITAION_HOP_COUNTER**	% Links with the Identical Destination Node	Constant
33.	**DESTINATION _HOST_SERVICE_COUNTER**	Group of Links; Identical Network State and Use the Identical Service	Constant
34.	**DESTINATION _HOST_IDENTICAL_ SERVICE_RATE**	% Links that have the Identical Destination Address using the Identical Service	Constant
35.	**DESTINATION _ MULTI-HOP_SERVICE_RATE**	% Changed Services on the Unique Host	Constant
36.	**DESTITAION_HOST_IDENTICAL_SOURCE PORT**	% Interconnection to the Domain Controller that uses the Root Terminal	Constant
37.	**DESTITAION_HOST_ SERVICE_MULTI-HOST**	% Links to the Identical Provider from Multi-Hop Networks	Constant
38.	**DESTITAION_HOST_SERVICE_ERROR_RATE**	% Links to the Domain Controller with an S0 Failure	Constant
39.	**DESTITAION_HOST_SERVICE_ERROR_RATE**	% Links to the Novel Host and Selected Platform that Includes anS0 Fault	Constant
40.	**DESTITAION_HOST_REQ_ERROR_RATE**	% Links to the Main Host with an RST failure	Constant
41.	**DESTITAION_HOST_ SERVICE _REQ_ERROR_RATE**	% RST Failures on Links to the Original Host and Designated Service	Constant

```
Step 100 Return $X_{Best}$ =f($X_{Best}$ Solution).
Step 110 End.
```

.

4 Results and Discussions

4.1 Test Feature Set

The KDD'99 data set is the most commonly utilized set of data for evaluating IDS. This model obtains the data collection during the DARPA'98 IDS assessment program. Approximately 8 GB of compressed raw (binary) TCP dump data are found in DARPA'98 from seven weeks of network traffic, which is liable to a transformation of about 5 million link records containing 110 *Bytes* approximately. Almost 2 million communication records are included in the 14 days of testing results. The KDD training dataset has about 4,950,000 single link vectors, of which most of them contain 41 features and are marked as either normal or an assault with a distinction in each type of attack.

4.2 Support Vector Machine

The Support Vector Machine is a two-classifier algorithm. Table 1, which was used for SVM feature selection, indicates the feature collection. It has a hyper-optimal that is split into two groups. The algorithm evaluates the supporting vector to represent the hyperplane, which achieves the highest accuracy. The kernel classifier and classifier design processes for network anomaly detection problems were applied. They test the kernel type bang and the limit values on the precision of the intrusion classification by an SVM. Classification accuracy was shown to vary with kernel type and parameter values. SVMs could detect intrusions with improved efficiency and reduce the number of false alarms when the input parameters are appropriately selected. Kernel function has few advantages, such as less convenient parameters and nonlinear solid forecasting.

4.3 Data Set

To train and test the proposed model KDD'99 data set is used. It is a benchmark data set that was offered by the designer of the intrusion detection system. The data set contains the 41 feature that is grouped into four types:

a. *Basic Features* Extracted without payload from the packet header.
b. *Content Features* The payload from the initial TCP packet can be evaluated using domain information.
c. *Time-based Traffic Features* The same communication features in the last two seconds between origin and the endpoint.
d. *Host-based Traffic Characteristics* Calculate the expected window over time rather than connection numbers.

4.3.1 In the KDD cup, all of the attacks are categorized as follows:

a. Denial of Service (DOS): Intruder denies customer service valid.
b. Probes: Intruder searches the overall computer networks to collect details or identify vulnerabilities for possible threats.
c. Remote to Local (r2l): The attacker attempts to get permission from remote access to the local user.
d. User to Root (u2r): The standard user is getting access to the root account.

4.4 Performance Evaluation

The feature selection is measured using different metrics, including accuracy, precision, recall, and F1-Score. Similar specifications are described in terms of True Positive (T_p), False Positive (F_p), False Negative (F_n), and True Negative (T_n).

- True Positive (T_p): To identify the packets, it is attempting to attacks.
- False Positive (F_p): To accept regular transmissions as an intruders.
- True Negative (T_n): Zero analysis of malicious nodes.
- False Negative (F_n): As its routine, the attack packets are observed.

Accuracy parameter is being used to compute the number of cases correctly recognized as normal or attack, as defined in the following equation:

$$Accuracy = \frac{T_p+T_n}{T_p+F_p+F_n+T_n}$$

Precision is used to evaluate true-positive and false-positive cases, as illustrated in the following formula:

$$\Pr ecision = \frac{T_p}{T_p+F_p}$$

The recall is being used to evaluate true-positive and false-negative cases. The following formula represents recall analytically.

$$\operatorname{Re} call = \frac{T_p}{T_p+F_n}$$

The F1-score is measured as the aggregate of recall and precision. This can be defined as follows:

$$F1 - Score = \frac{2x \Pr ecisionx \operatorname{Re} call}{\Pr ecision+\operatorname{Re} call}.$$

At present, performance reviews can lack accuracy and recall. If a process has a limited recall but a high accuracy, an additional criterion is needed. As a rule, the F1 score must address this issue.

Table 2 Test results

System	No. of features	Detection rate (%)	False alarm rate	Accuracy (%)
SVM	41	91.25	0.0721	86.45
SVM	30	87.54	0.0845	85.34
SVM	25	94.47	0.0484	94.16

Fig. 5 Performance analysis of different feature set with SVM approach

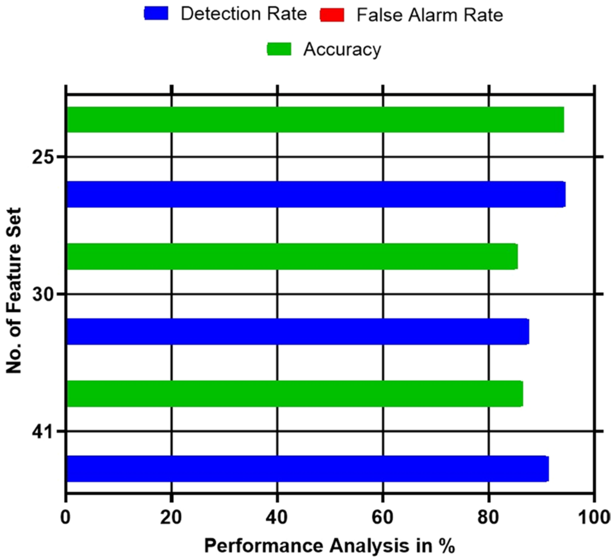
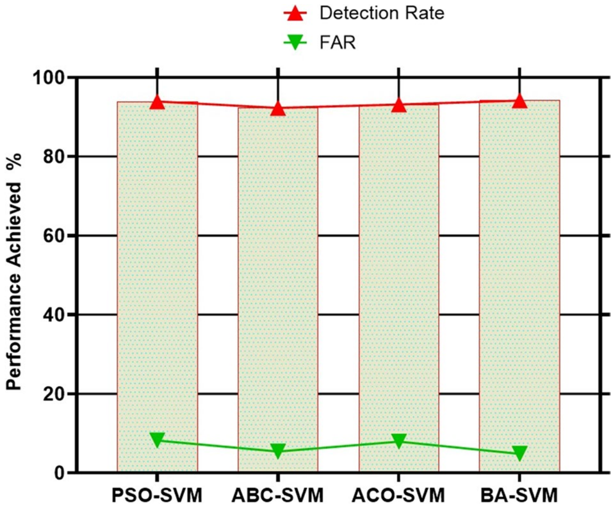
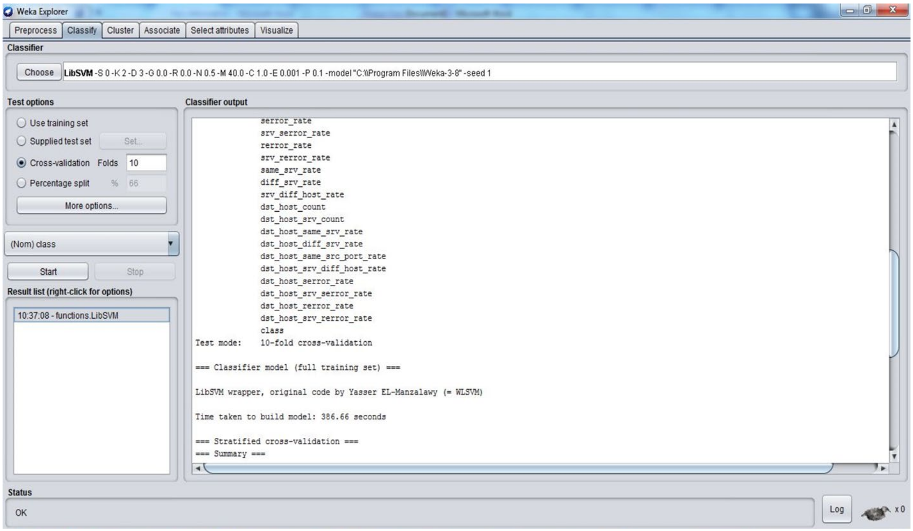


Table 3 Comparison with other IDS

System	Dataset	Detection rate (%)	FAR (%)
PSO-SVM	KDD99	93.92	8.2
ABC-SVM	KDD99	92.25	5.4
ACO-SVM	NSL-KDD	93.14	7.9
BA-SVM	KDD99	94.12	4.8

Fig. 6 Performance analysis of dataset with different SVM approach





```
LibSVM -S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -model "C:\\Program Files\\Weka-3-8" -seed 1
```

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

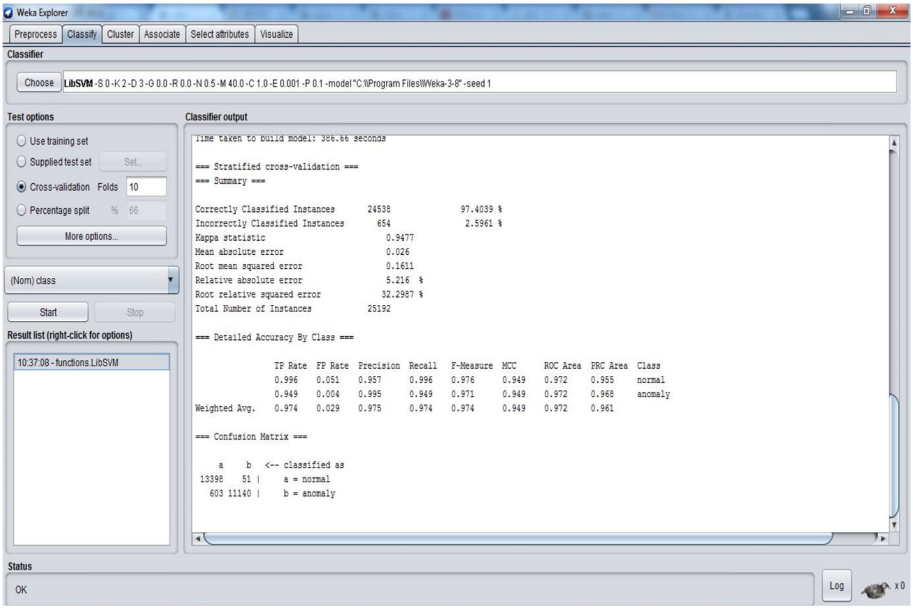
(Nom) class

10:37:08 - functions.LibSVM

```
             serror_rate
             srv_serror_rate
             rerror_rate
             srv_rerror_rate
             same_srv_rate
             diff_srv_rate
             srv_diff_host_rate
             dst_host_count
             dst_host_srv_count
             dst_host_same_srv_rate
             dst_host_diff_srv_rate
             dst_host_same_src_port_rate
             dst_host_srv_diff_host_rate
             dst_host_serror_rate
             dst_host_srv_serror_rate
             dst_host_rerror_rate
             dst_host_srv_rerror_rate
             class
Test mode:   10-fold cross-validation

=== Classifier model (full training set) ===

LibSVM wrapper, original code by Yasser EL-Manzalawy (= WLSVM)

Time taken to build model: 386.66 seconds

=== Stratified cross-validation ===
=== Summary ===
```

Status: OK

Fig. 7 SVM classification with 41 feature



```
LibSVM -S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -model "C:\\Program Files\\Weka-3-8" -seed 1
```

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) class

10:37:08 - functions.LibSVM

```
Time taken to build model: 386.66 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       24538               97.4039 %
Incorrectly Classified Instances       654                2.5961 %
Kappa statistic                          0.9477
Mean absolute error                      0.026
Root mean squared error                  0.1611
Relative absolute error                  5.216  %
Root relative squared error             32.2987 %
Total Number of Instances            25192

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.996    0.051    0.957      0.996    0.976      0.949    0.972     0.955     normal
                 0.949    0.004    0.995      0.949    0.971      0.949    0.972     0.968     anomaly
Weighted Avg.    0.974    0.029    0.975      0.974    0.974      0.949    0.972     0.961

=== Confusion Matrix ===

     a     b   <-- classified as
 13398    51 |     a = normal
   603 11140 |     b = anomaly
```

Status: OK

Fig. 8 True positive and false positive rate for 41 features

Fig. 9 Comparison chart with various IDS

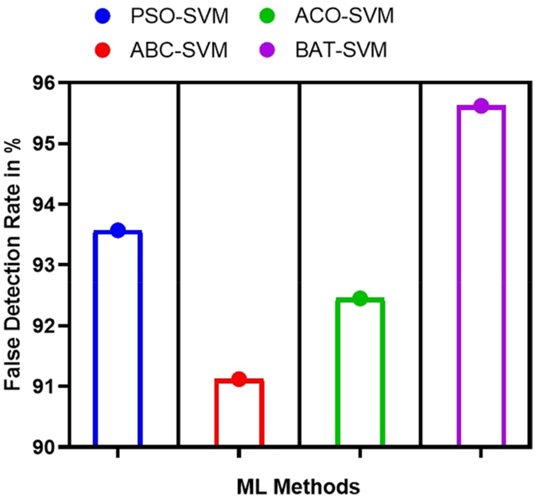
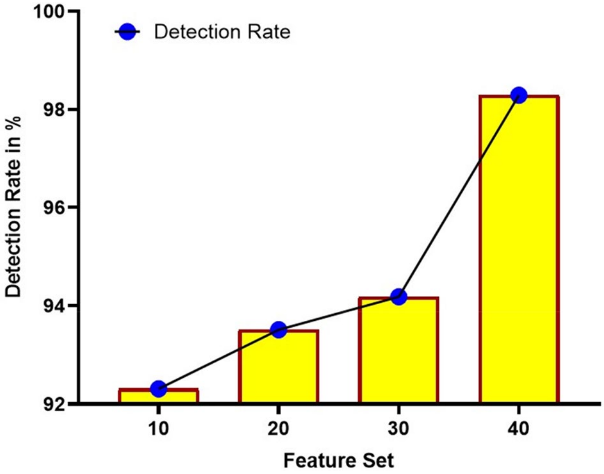


Fig. 10 Feature size Vs. detection ratio



4.5 Results and Analysis

An Intel Core ™ *i*7 processor with 8 GB of RAM under windows 10 was used to complete all tests. We have used input metrics for Bat algorithm:

- Maximum loudness A0 = 10.
- Minimum pulse rate r0 = 0.9.
- $\mathrm{Freq}_{\mathrm{min}} = 0.8$ and $\mathrm{Freq}_{\mathrm{max}} = 1.0$.
- $\alpha = 0.9$ and $\gamma = 0.1$.

.

SVM gives 86.45 % precision for a maximum of 41 features. The choice of feature selection employing the bat algorithm led to greater accuracy. The results of the tests and output of the feature set are indicated in Table 2; Fig. 5. The suggested pattern

Fig. 11 Feature size Vs. search time

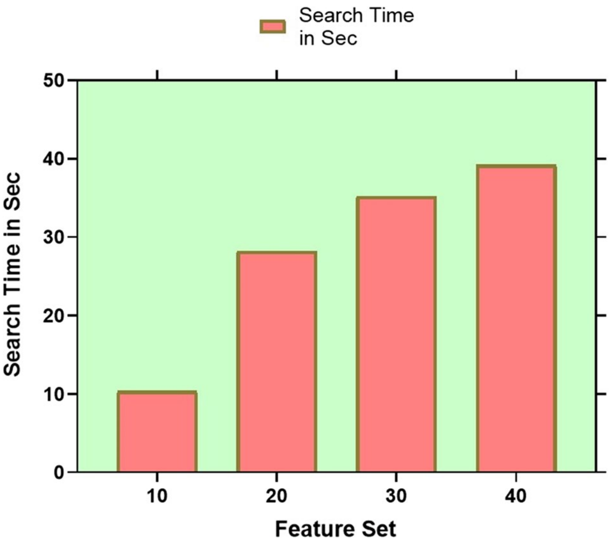
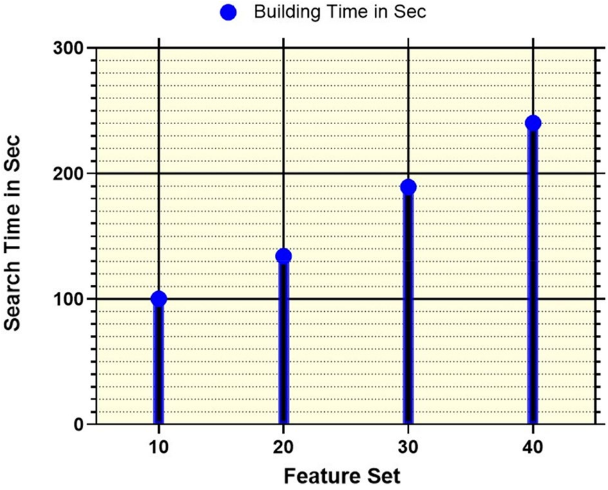


Fig. 12 Feature size Vs. building time



was linked to other feature selection methods, such as PSO, Ant Colony Optimization (ACO), Artificial Bee Colony (Table 3; Fig. 6). The following table results indicate that the suggested framework generates significant improvements in detection rate and FAR. Performance Achieved.

4.6 Experimental Results

The test was conducted to find the time needed to construct a model concerning the feature size. For development and research, the detecting ratio of the model was evaluated using the same dataset. The result shows that the Detection Ratio (almost the same) changes slightly when the characteristic size decreases; instead, the layout time and the model building duration vary greatly. Figure 7 shows the classification characteristics, and Fig. 8 shows the True Positive Rate and False Positive Rate. Figures 9, 10 and 11, and Fig. 12 demonstrate the similarities of the suggested technique with other techniq ues.

5 Conclusions

This article's significant aspect is to identify the appropriate IDS function. Since eliminating irrelevant attributes is one of the Intrusion Detection systems challenging works. This feature selection approaches reduced dataset attributes while enhancing the classifier's prediction performance, detection accuracy, and false alarm rate. To assess the suggested model's efficiency, the KDDcup99 IDS benchmark data collection has been used. And for different datasets, the presented algorithm achieves positive performance. The proposed Bat algorithm with SVM constructs a wrapper system for 41 feature selection and chooses the appropriate features. The KDDCup99 test showed attacker detection accuracy, and the false alarm rate is better than PSO and ABC in the suggested method.

Funding Not Applicable.

Declarations

Conflict of interest The author declares their is no conflict of interest.

References

1. Yang, X.-S. and He, X. (2013). Bat algorithm: Literature review and applications. International Journal of Bio-Inspired Computation, 5(3):141–14
2. Yang, X.-S. and Deb, S. (2009). Cuckoo search via l´evy flights. In Proceedings of the World Congress on Nature & Biologically Inspired Computing, pages 210– 214. IEEE
3. Wang, J., Li, T., and Ren, R. (2010). A real-time IDSs based on artificial bee colony-support vector machine algorithm. In Proceedings in the International Workshop on Advanced Computational Intelligence, pages 91–96. IEEE.
4. Shadi Aljawarneh, Monther Aldwairi, and Muneer Bani Yassein. (2018). "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model." Journal of Computational Science, 25, 152–160
5. Shi-Jinn Horng, Ming-Yang Su, Yuan-Hsin Chen, Tzong-Wann Kao, Rong-Jian Chen, Jui-Lin Lai, and Citra Dwi Perkasa. (2011). "A novel intrusion detection system based on hierarchical clustering and support vector machines." Expert systems with Applications, 38(1), 306–313.
6. X Zhu, Z Huang, H Zhoul, Design of a Multi-agent Based Intelligent Intrusion Detection System. IEEE International Symposium on Pervasive Computing and Applications (IEEE, Amsterdam, 2006), pp. 290–295
7. G Bourkache, M Mezghiche, K Tamine, A Distributed Intrusion Detection Model Based on a Society of Intelligent Mobile Agents for Ad Hoc Network, in the 2011 Sixth IEEE International Conference on Availability, Reliability, and Security, Vienna, August 2011 (IEEE, Amsterdam, 2011), pp. 569–572
8. C-h Fonk, GP Parr, PJ Morrow, Security schemes for Mobile Agent-based Network and System Management Framework. J. Networks Syst. Manag. Springer 19, 232–256 (2011)
9. L Zadeh, Role of soft computing and fuzzy logic in the conception, design, and development of information/intelligent systems, in Computational Intelligence: Soft Computing and Fuzzy-neuro Integration with Applications, ed. by O Kaynak, L Zadeh, B Turksen, I Rudas. Proceedings of the NATO

Advanced Study Institute on Soft Computing and its Applications held at Manavgat, Antalya, Turkey, 21–31 August 1996, volume 162 of NATO ASI Series (Springer, Berlin, 1998), pp. 1–9

10. M. S. Hoque, Md.A. Mukit, and Md.A. N. Bikas, "An implementation of intrusion detection system using genetic algorithm," International Journal of Network Security & Its Applications, vol. 4, no. 2, pp. 109–120, 2012.
11. T. T. H. Le, Y. Kim, and H. Kim, "Network intrusion detection based on novel feature selection model and various recurrent neural networks," Applied Science, vol. 9, no. 1392, pp. 1–29, 2019
12. S.-B. Cho and H.-J. Park, "Efficient anomaly detection by modeling privilege flows using hidden Markov model," Computers & Security, vol. 22, no. 1, pp. 45–55, 2003.
13. W. Wei and C. Guo, "A text semantic topic discovery method based on the conditional co-occurrence degree," Neurocomputing, vol. 368, pp. 11–24, 2019.
14. P. Sun, P. Liu, Q. Li et al., "DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system," Security and Communication Networks, vol. 2020, Article ID 8890306, 11 pages, 2020.
15. G. Farahani, "Feature selection based on cross-correlation for the intrusion detection system," Security & Communication Networks, vol. 2020, Article ID 8875404, 17 pages, 2020.
16. D. Smith, Q. Guan, and S. Fu, "An anomaly detection framework for autonomic management of compute cloud systems," in Proceedings of the 2010 IEEE 34th Annual Computer Software and Applications Conference Workshops, pp. 376–381, IEEE, Seoul, South Korea, July 2010
17. H. B. Nguyen, B. Xue, P. Andreae, et al., "Particle swarm optimization with genetic operators for feature selection," in Proceedings of the 17 IEEE Congress on Evolutionary Computation (CEC), pp. 286–293, IEEE, San Sebastian, Spain, June 2017.
18. M. H. Aghdam and P. Kabiri, "Feature selection for intrusion detection system using ant colony optimization," International Journal of Network Security, vol. 18, no. 3, pp. 420–432, 2016.
19. B. Alsalibi, L. Abualigah and A.T. Khader 2021 "A novel bat algorithm with dynamic membrane structure for optimization problems", Electronics, vol. 10, pp. 1992–2017.
20. D. Mustafa Abdullah and A. Mohsin Abdulazeez, "Machine Learning Applications based on SVM Classification A Review", Qubahan Academic Journal, vol. 1, no. 2, pp. 81–90, 2021.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Dr. N. Satheesh is working as a Professor, Department of Computer Science & Engineering in St. Martin's Engineering College, Dhulapally, Secunderabad. Since May 2019. He obtained B.E., in Electronics & Communication Engineering from Sri Balaji Chockalingam Engineering College, Arani, University of Madras in 2004, M.E., in Computer Science & Engineering from Faculty of Engineering & Technology, Annamalai University, Chidambaram in 2008 and Ph.D. in Computer Science & Engineering from Karpagam Academy of Higher Education, Karpagam University in 2018. His area of specialization is Wireless Security, Wireless Sensor Networks, Internet of Things. He has 12+ years of teaching experience and 2+ years of software experience. He has 13: International Journal Publications in SCI/SCOPUS/UGC CARE, 06: International Conference Publications, and 06: Patent Publications. He received Teaching and Research Excellence National Award in 2020.

Dr. Sudhakar Sengan received PhD degree in Information and Communication Engineering from Anna University, Chennai, Tamil Nadu, India. And received his ME degree in the Faculty of Computer Science and Engineering from Anna University, Chennai, Tamil Nadu, India in 2007. He is presently working as Professor and Director International Relation, Department of Computer Science and Engineering, PSN College of Engineering and Technology (Autonomous), Tirunelveli–627152, Tamil Nadu, India. He has 20 years of Experience in Teaching/Research/Industry. He has published papers in 85 International Journals, 20 International Conferences and 10 National Conferences. His research interest includes Network Security, Information Security and Mobile Ad Hoc Network. He has filled 15 Indian and 3 International Patents in various field of interest. He is a member of various professional bodies like MISTE, MIEEE, MIAENG, MIACSIT, MICST, MIE and MIEDRC. He guided more than 100 Projects for UG & PG students in engineering streams. He is the Recognized Research Supervisor in Anna University under the faculty of Information and Communication Engineering. He is a Member of Editorial Board in many reputed journals like Elsevier, Springer, IEEE, Inderscience, Wiley, etc., He is an Expert Member in Center of Excellence, Syllabus Committee in various reputed Universities. He received an award of Honorary Doctorate (Doctor of Letters-D.LITT.) from International Economics University; SAARC Countries in the field of Education and Students Empowerment in the month of April 2017. He is currently guiding many research scholars in various Universities. He delivered Guest Lectures at Various Autonomous Institutions and Universities. He is Doctoral Committee Member for many scholars in Anna University, Dr.MGR University, SRM University, Vels University, Bharathiyar University, Prist University. He has published 3 Text books for Anna University, Chennai Syllabus.

Saira Khurram is a senior Teacher dealing with all fields of Biology .In her 17 years teaching career .She had performed immense recorded performances in teaching methodology and curriculum development. she had contributed immensely to the enhancement of High School Education. She had conducted many workshops for range of audience regarding classroom management. & eye ball to eye ball concentration while applying teaching methodologies. She had helped many candidates of High School for their research during this whole educational tenure . Her students are not only Alumni but also attained Distinctions in Biology Nationaly & Internationaly. She is been serving as Internal Practical Supervisor for Course Code 9700 A level.

Farrukh Arslan did his BSc in Electrical Engineering from the University of Engineering and Technology, Lahore, Pakistan and MSc and PhD from Purdue University, USA. Currently he is working as an Assistant Professor in UET Lahore, Pakistan. He is experienced with working in information technology and data science in academia. His research interests include Machine Learning, Data Mining and Data Science.

M. Suresh Kumar has 19 years of experience in the field of Teaching and Research and is currently working as a Associate Professor, Information Technology Department, Sri Sai Ram Engineering College, Chennai. He received his Ph.D degree under Information and Communication Engineering, Anna University in Web Service composition. Also, his research area includes Web Service, Cloud Computing, IoT, Machine Learning, Semantic Web and Cyber Security. Around thirty numbers of peer reviewed International Journal and Conference publications are there in his credit. He is currently supervising two Ph.D. scholars.

R. Regin has received his B.E. (Computer Science and Engineering) degree in the year 2007 From CSI Institute of Technology, Nagercoil, and M.E. (Computer Science and Engineering) degree in the year 2011 from Annamalai University, Chidambaram. Currently he is pursuing his Ph.D. in Anna University, Chennai, in the field of WSN, VANET, and MANET. He has 9 years of teaching experience, and he is working as an Assistant Professor in the Department of Information Technology at Adhiyamaan College of Engineering, Hosur, Tamil Nadu.

Dr. P. Vijayakumar is currently working as Associate Professor in School, of Electronics Engineering at Vellore Institute of Technology, Chennai, Tamil Nadu, India and completed his Ph.D in Wireless Communication and Network Security at Pondicherry University, Pondicherry during 2015. He Completed his B.Tech. in Rajiv Gandhi College of Engineering & Technology, Puducherry during 2004 and M.Tech. in Pondicherry Engineering College, Pondicherry during 2006. He has totally 15 years of teaching and research experience and published more than 40 research papers in SCOPUS /SCI Indexed National/International Journals and Conferences. His area of specialization is Elliptic and Hyperelliptic Curve Cryptography, Blockchain technology, Cryptography and Network Security, Cryptographic Algorithms, DNA Steganography, Embedded System and IoT. He filed, published and got granted for more than 10 Patent.

Dr. Anil Kumar Dubey is a Senior Member of IEEE 10 years now. He has previously served as an Excom Member of IEEE Rajasthan Sub Section, Chair of Humanity & Technologies at IEEE Rajasthan Sub Section. He has graduated in 2008 from Uttar Pradesh Technical University Lucknow, INDIA. In 2010, he obtained Master's Degree from Rajasthan Technical University, Kota, INDIA. He received his PhD in Computer Science & Engineering from Career Point University Kota, INDIA in 2015. Dr. Dubey is currently working as an Assistant Professor (CSE) in ABES Engineering College, Ghaziabad, NCR, INDIA. Previously he was Associate Professor (CSE) in Poornima Institute of Engineering & Technology Jaipur, Rajasthan, INDIA and Faculty in the Department of CS & IT at Government Engineering College Ajmer, Rajasthan, INDIA. He has organized and participated in various international & national conferences/workshops/FDPs. He has published several articles in reputed journal and conferences as Springer, IEEE, ACM etc. He has delivered various Keynote addresses and won various awards. He serves as editorial board member and Committee Member of many international journals as well as international Conferences and reviewed papers. He is also a member of societies such as IEEE, ACM, ISTE, CSI, CSTA, etc.

Dr. S. Sujatha received the B.E. Degree in Telecommunication Engineering from MVJ College of Engineering, Bangalore, India, M.Tech. Degree in VLSI and embedded system design from the BMS College of Engineering Bangalore, India, and Ph.D. degree in Electronics and Communication Engineering from, Pondicherry Engineering College, Pondicherry, She is working as Assistant Professor in the department of Electronics and Communications Engineering, Christ (Deemed to be University, School of Engineering and technology Bangalore India since 2017 till date. Her research interests include Wireless Communication, Optical Communication, Optical Communication Network, Microcontroller, and MSP430. She is presented and published several papers in National and International journal.

Dr. Dilip Kumar Sharma did his M.Sc. (Mathematics) from Government PG College Guna (M.P.) in the year 1998 and M.Tech. (Future studies and planning) from School of Future studies and planning, Devi Ahilya University Indore (M.P.) in the year 2002. He did his Ph.D. from JUIT Wakanaghat, Solan (H.P.) in the year 2009. Presently he is working in Jaypee University of Engineering and Technology, Guna (M.P.). He has about 16 years teaching experience. He is member of IEEE, Bombay Section. He is life member of Forum for Interdisciplinary Mathematics (FIM) and Indian Science Congress. He has published many research papers in reputed international journals and presented research papers in Conferences. He has visited NUS Singapore and Concordia University, Montreal, Canada. He is also a member of editorial board of the JUET Research Journal of Science and Technology. He has supervised 3 Ph.D. scholars and one Post-Doctoral fellow sponsored by NBHM, DAE, Mumbai and he is currently supervising two Ph.D. scholars.

Authors and Affiliations

Satheesh Narayanasami[1] · Sudhakar Sengan[2] · Saira Khurram[3] · Farrukh Arslan[4] · Suresh Kumar Murugaiyan[5] · Regin Rajan[6] · Vijayakumar Peroumal[7] · Anil Kumar Dubey[8] · Sujatha Srinivasan[9] · Dilip Kumar Sharma[10]

Satheesh Narayanasami
nsatheesh1983@gmail.com

Saira Khurram
sarakhurram28@gmail.com

Farrukh Arslan
farrukh_arslan@uet.edu.pk

Suresh Kumar Murugaiyan
sureshkumar.it@sairam.edu.in

Regin Rajan
regin12006@yahoo.co.in

Vijayakumar Peroumal
vijayrgcet@gmail.com

Anil Kumar Dubey
anildudenish@gmail.com

Sujatha Srinivasan
sujatha.s@christuniversity.in

Dilip Kumar Sharma
dilipsharmajiet@gmail.com

[1] Department of Computer Science and Engineering, St. Martin's Engineering College, Hyderabad, Telangana 500100, India

[2] Department of Computer Science and Engineering, PSN College of Engineering and Technology, Tirunelveli, Tamil Nadu 627152, India

[3] Biology Teacher, Teaching Human Social Biology and Applied Sciences, Roots IVY International Schools, Faisalabad 38000, Punjab, Pakistan

[4] University of Engineering and Technology, Lahore 39161, Punjab, Pakistan

[5] Department of Information Technology, Sri Sai Ram Engineering College, Chennai, Tamil Nadu 600044, India

[6] Department of Computer Science and Engineering, Adhiyamaan College of Engineering, Hosur, Tamil Nadu 635109, India

[7] School of Electronics Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu 600048, India

[8] Department of Computer Science and Engineering, ABES Engineering College, Ghaziabad, Uttar Pradesh 201009, India

[9] Department of Electronics and Communications Engineering, School of Engineering and Technology, Christ (Deemed to be University), Bangalore, Karnataka 560029, India

[10] Department of Mathematics, Jaypee University of Engineering and Technology, Guna, Madhya Pradesh 473226, India