Adaptive Voting Mechanism with Artificial Butterfly Algorithm based Feature Selection for IDS in MANET

Conference Paper · February 2023

DOI: 10.1109/ICICACS57338.2023.10099861

CITATIONS
16

READS
151

4 authors, including:

Parameshachari B D
Nitte Meenakshi Institute of Technology
216 PUBLICATIONS **5,460** CITATIONS

SEE PROFILE

Achyutha Prasad N
East West Institute of Technology
59 PUBLICATIONS **663** CITATIONS

SEE PROFILE

2023 IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS)

Adaptive Voting Mechanism with Artificial Butterfly Algorithm based Feature Selection for IDS in MANET

1st Parameshachari B.D.
Department of Electronics and Communication Engineering
Nitte Meenakshi Institute of Technology
Bengaluru, India
paramesh@nmit.ac.in

2nd Achyutha Prasad N
Department of Computer Science and Engineering
East West Institute of Technology
Bengaluru, India
achyuth001@gmail.com

3rd Dhanraj
Department of Computer Science and Engineering
East West Institute of Technology
Bengaluru, India
draj148@gmail.com

4th Manjunath T N
Department of Computer Science and Engineering
East West Institute of Technology
Bengaluru, India
manju.ssit@gmail.com

***Abstract*—Mobile ad hoc networks (MANETs) have gained more interest from consumers and academics than ever before thanks to the proliferation of wireless networks and the expansion of the benefits and uses of communication networks in general. MANETs are useful in a wide variety of settings since they don't rely on a centralised server or other hardware to relay messages or process data packets. It's one of the primary justifications for implementing MANET in many different domains. However, there are also numerous difficulties that have arisen as a result of these networks' rising popularity, with network security being one of the most crucial. There have been challenges with data transmission and reception due to MANETs' weak regulatory and security frameworks; network infiltration has been identified as one of the most pressing concerns. In MANETs, wireless nodes serve as relays and routers, connecting the source and sink nodes. Accordingly, it is now possible for rogue nodes to penetrate networks and destroy data packets. In order to cope with this issue, modern intrusion detection systems (IDSs) are utilised for remote monitoring of the functioning and actions of nodes present in wireless sensor networks. As well as being able to identify hostile nodes in the network, IDSs can often predict how such nodes will act in the future. In this research work, NSL-KDD dataset is used as an input data. SMOTE and Z-score method are used during pre-processing to remove the irrelevant features and normalize the data. The optimal features are carried out by Artificial Butterfly algorithm and then, finally, ensemble classifiers s. Multilayer Perceptron (MLP), Boosted Regression Trees (BRT) and finally, the adaptive voting mechanism is used to select the best classifier. The results proves that the proposed ensemble model achieved 97.16% of accuracy, where the existing models achieved nearly 95% to 96% of accuracy.**

Keywords—Intrusion Detection Systems; Artificial Butterfly algorithm; Adaptive voting mechanism; Mobile ad hoc networks; Boosted Regression Trees.

I. INTRODUCTION

Mobile Adhoc Network (MANET) has been seen as a significant agency in the recent field of computer science. The mobile adhoc network has been used to reach different network resources where the mobile node cannot explicitly access the network services but can only be reached with the assistance of other intermediary nodes. When accessing such resources with the assistance of intermediate nodes, they face a number of challenges. Each node in MANET acts as a router or an intermediary system due to a sudden shift in the network topology; it can impact the whole network layout, and the data can miss it while the topology shifts [1-2]. Nodes that submit alerts can increase the system load latency; it's why we need to make sure there are no loops. In addition, every network is designed to include a collection of resources for some given purpose. To accomplish this aim, a series of services would be given to users of the network or to users of any device operating on the network. Users registered with the service are authorized to use the service in compliance with the protocol listed above. In fact, there are malicious users or adversaries present within or outside the network, which may also be registered users [3]. Their aim is to degrade or weaken the network infrastructure offered. And the involvement of hostile nodes in the way of data transfer will entail creating various network risks. Due to resource constraints, the presence of permanent security monitoring nodes in the network is almost impossible, and remote control of node activity in the network and establishing security demands in MANET are thus required [4, 5].

A (NIDS) is a tool used to keep tabs on all the goings-on in a network. The primary function of NIDSs is to identify potentially harmful nodes and to foresee network assaults [6]. In the event that a malicious node is found in the network, an alert is produced for further action. It is important to note that the effectiveness of an IDS is dependent on the sort of technology used to identify assaults by NIDS, and that many different strategies have been suggested for doing so [7]. Choosing relevant characteristics from the primary dataset is an important part of NIDS performance [8]. Optimizing IDS performance [9] often involves reducing the amount of characteristics included in the data collection (such as the behaviour of nodes and network traffic) without sacrificing classification accuracy. Defending networks and devices against more sophisticated assaults like denial of service [10-11] is challenging using today's intrusion detection and prevention tools like firewalls, access control protocols, and authentication. However, most systems that rely on such methods have trouble keeping up with the constantly changing harmful practises they're supposed to

2023 IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS) | 979-8-3503-9846-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICICACS57338.2023.10099861

protect against due to their high rates. Thus, Machine Learning (ML) enables speedy data processing and visualisation, with the goal of facilitating the discovery of device vulnerabilities and defects by security specialists. To improve detection rates and flexibility, many ML means have been used to the problem of intrusion detection. These approaches are often used to establish the robustness and depth of the attack's current knowledge base [12].

In this study, we use a synthetic Butterfly procedure to choose features, with the help of two pre-processing methods. In order to classify IDS data, we use four distinct ML methods and then utilise an adaptive voting mechanism to choose the most effective of the four. The remaining pieces of this work are laid out as shadows. In this second section, we will talk about security threats in MANET. The MANET intrusion detection system designs are then outlined in Section 3. Results and investigation of the experiments are presented in Section 4. In the fifth part, we sum up our findings.

II. Related works

Stacked auto-encoder based strategy for lowering correlation, and it was first proposed by Meddeb et al. [13]. It employs numerous processing layers in an effort to model important aspects at a high level and to get a suitable illustration feature from Data Correlation. The goal of the stacked AE-IDS technique is to provide an output with lower correlation, where the input's dimensionality is identical to that of the output. There are two stages to our proposed Deep Learning-based IDS. Deep Neural network (DNN) classifiers take their input from the encoder's output (DNN-IDS). It leverages the most likely assaults to disrupt routing services in Mobile Network and zeroes down on DoS attacks within labelled datasets available for intrusion detection.

The accuracy of a IDS relies heavily on the careful selection of protocol elements and the creation of fuzzy rules, as shown by Makani and Reddy [14]. (FIDS). In this article, we present a set of fuzzy rules to prevent blackhole attacks on networks. The RREQ rate, RREP rate, and Sequence number value are the three most important attributes used in the creation of these rules. The suggested FIDS were then tested in the ns2 simulator, where they were found to be effective in locating the malicious node and cutting it off from the network. The network throughput has improved when FIDS was implemented.

In this study, we present a new kind of Fuzzy Extreme learning machine (PCA-FELM) that is based on Principal Component Analysis, as originally described by Sing and Vigila [15]. Analysis is used to extract the features, and then the Fuzzy Extreme Learning Machine is used to categorise the features. A MAT LAB simulator is used to test the proposed PCA-FELM. When compared to other approaches, is shown to be more accurate (99.08%). The new PCA-FELM model has been exposed to outperform current methods in experimental evaluations on the Cup99 dataset.

Prabaharan, and Venkateswaran [16] In order to differentiate between assaults on MANETs, this article suggests using a neural Deep learning wireless intrusion detection scheme. Security implementation in MANET is challenging because of the network's inherent weaknesses. The suggested framework incorporates a hybrid conspiracy that brings together the resolve and abnormality-based approaches, providing additional security for such systems thanks to deep learning. The identification rate in MANETs is increased when the partial IDS is carried out using neural Deep learning. The suggested strategy makes use of a hybrid neural system and deep neural networks. In doing so, it proves that Recurrent Neural Networks may effectively enhance identification while simultaneously reducing the occurrence of false positives and negatives.

In the network layer, the "Malicious Packet Dropping Attack" is investigated by Vijayalakshmi et al [17]. In order to prevent this kind of assault, a new approach is developed that makes use of game theory for security. Secure communication among nodes that talk to one another to route traffic from source to destination is made possible thanks to the suggested system, which monitors the behaviour of neighbouring nodes and overcomes the drawbacks, such false positives, of conventional IDS. The suggested approach has increased the packet delivery ratio by 42% despite the presence of hostile nodes.

Protecting against blackhole, grayhole, and selfishness attacks, Sbai and Elboukhari [18] present a MANETs-IDS based on a machine learning algorithm using the ns-3 simulation platform the optimised link state routing (OLSR) protocol (RFC 3626).When running our simulation, we accounted for both the network's density and a model of random mobility for nodes. Observed experimental findings demonstrate the suggested detection method achieved encouraging results.

III. Proposed Model

Fig. 1 offerings the working flow of projected typical.

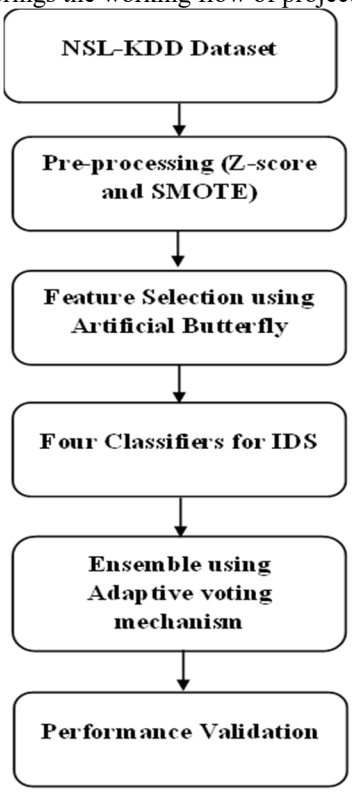

Fig. 1 Process of proposed model

A. Dataset Description

In previous work, we took into account the NSL-KDD data set, which is a popular resource for leak detection. All four common forms of cyberattacks—Probe, U2R, R2L, and DoS—are represented in the dataset alongside "normal" data. Each incursion record has 42-dimensional information, broken down into three groups: a 3-dimensional symbol feature, a traffic type designation, and 38-dimensional digital information. Table I provides a summary of the data.

TABLE. I. DESCRIPTION OF DATASET

Category	Train	Test
DoS	11656	7458
Probe	45927	2421
R2L	995	2756
Normal	67343	9711
U2R	52	200
Total	125973	22544

B. Data Pre-processing:

Your data may be normalised using the Z-score approach and the minority group can be oversampled with the SMOTE [19] algorithm that is utilised in the data preprocessing step. During pre-processing, we provide the training and testing cases indicated in Table I as input to the normalisation and SMOTE procedures.

C. Z-Score Normalization:

The first stage of processing data is normalising the data using a Z-score. Prior to their usage, all categorical characteristics in the data must be encoded using a label encoder to be converted to numerical ones. As can be seen here, normalisation is accomplished by assigning a value of x i norm, the normalised version of the original x i, to each data point in the collection is given in (1).

$$x_{norm} = \frac{x_i - \mu}{\sigma} \quad (1)$$

The mean of the feature is denoted by while the standard deviation is represented by. Z-score data is normalised, which is significant since Fuzzy-based approaches work best with normalised datasets..

D. SMOTE Technique

Secondly, minority class oversampling is accomplished via the use of the SMOTE method. By artificially generating additional examples of the underrepresented minority class, this method helps to get a more even distribution of classes for the ML classification model's output [20]. Minority class oversampling in execution improves training model performance [20], particularly for datasets that include network traffic and are prone to this issue.

On the basis of the examination of the original examples, the SMOTE algorithm generates new cases of the minority class. In this way, all instances of the minority class are collected into a single set, denoted here as X minority. In order to produce a new synthetic instance X new for each instance X inst included in X minority, we use the following expression (2).

$$X_{new} = X_{inst} + rand(0,1) * \left(X_j - X_{inst}\right), j = 1,2,..,k \quad (2)$$

The collection of X inst's k closest neighbours (X 1,X 2,...,X k) is randomly sampled using a random integer in the range [0,1]. When used as an oversampling method, SMOTE generates brand-new, high-quality examples that statistically match samples from the minority class. The following step, feature selection, involves picking the best features from the NSL-KDD data that has been normalised and made balanced by the SMOTE procedure.

E. Feature Selection

To comprehend model detection implementation and temporal complexity, this study compares two feature selection strategies: one based on information gain and the other on correlation. This is of utmost importance for constructing hypothetical models for complex, large-scale systems that generate high-dimensional data.

1) Artificial Butterfly Algorithm: Specifically examining how life history optimization influences lifetime investment in aggressive behaviour [21] may be done with great success using butterfly territoriality as a model system. The unique patterns and textures of spotted woods may also serve as a fantastic jumping off point for developing a whole new optimization technique. We devised the (ABO) algorithm, which is based on the method through which speckled trees locate mates. As shown in Table II, the ABO algorithm's pseudo code may be found online. In the ABO algorithm, some guidelines are set up to idealise the butterfly's methods of mating.

a) All male butterflies try to travel toward a better position called a sunspot in the hopes of increasing their chances of meeting female butterflies.

b) Each sunspot butterfly is always trying to relocate to its neighbor's sunspot, since the latter is invariably a better location.

c) All of the butterflies in the canopy are constantly racing toward the sunspot butterflies in an effort to claim it.

TABLE. II. PSEUDO CODE OF ABO ALGORITHM

```
1. Initialize the locations of butterfly population
2. Evaluate the fitness of every butterfly
3. While not meet the terminal condition
4. Sort all butterflies by their fitness
5. Select some butterflies with better fitness to form sunspot l
form canopy butterflies
6. For each sunspot butterfly fly to one new location accordin
Evaluate the fitness of the new sunspot
apply greedy selection on the original location and the new or
End for
7. For each canopy butterfly
Fly to one randomly selected sunspot butterfly according to c
Evaluate the fitness
8. If better fitness
Apply greedy selection on the original location and the new or
9. else
Fly to new location according to free flight mode
End if
```

Table II shows how the initial butterfly population was stratified into two groups based on fitness. Each group is given its own unique flight plan. There's a parallel here with niching strategies. To efficiently locate several optimum solutions, it is common practise to change the behaviour of a classical

algorithm using a niching approach [22] to preserve different groups in the selected population component. The sunspot flying mode, the canopy flight mode, and the free flight mode are the three possible ways to fly. Various flying techniques are available for these modes. That is to say, when each of the three flight modes is assigned a different flying strategy, ABO may generate a brand-new algorithm.

We provide three possible butterfly flight paths. In this study, the position of a digital butterfly is represented by a vector of dimension D.

1) First, each butterfly flies in the direction of a neighbour at random using Eq (3). For either the sunspot or canopy flying modes in the ABO algorithm, this tactic is used is given in (3).

$$X_{i,j}^{t+1} = X_{i,j}^{t} + \left(X_{i,j}^{t} - X_{k,j}^{t}\right).rand() \quad (3)$$

2) And I am the butterfly in this metaphor. Here, j is a dimension index chosen at random between [1,D], t is the sum of repeats, rand() produces a random integer between [-1,1], and k is a butterfly chosen at random. At this point, k is distinct from i.

3) If you follow the equation in Eq., each butterfly will fly towards a neighbour that is chosen at random (4). The ABO algorithm's sunspot flying mode and canopy flight mode both make advantage of this tactic given in (4).

$$X_i^{t+1} = X_i^t + \frac{x_k^t - x_i^t}{\|x_k^t - x_i^t\|}.(Ub - Lb).step.rand() \quad (4)$$

4) where I is the butterfly being simulated, t is the sum of iterations, X i(t+1) is the butterfly's novel position, step is the flight distance, rand() produces a random value between 0 and 1, and k is a randomly chosen butterfly. At this point, k is distinct from i. Where Lb is the lower boundary value of the ith virtual butterfly's flight range and Ub is the upper boundary value of the ith virtual butterfly's flight range. There is an issue for which Lb and Ub are useful.

5) According to the formula in Eq (5). A similar strategy has been used during the exploration phase to look for a new position [23]. This method is implemented in the ABO algorithm's free-flight mode is given in (5).

$$X_i^{t+1} = X_k^t - 2.a.rand(\,) - a.D \quad (5)$$

6) where an is gradually dropped from 2 to 0 throughout the length of the iteration, X i(t+1) is the new position of the ith virtual butterfly, and rand() creates a random value between 0 and 1. (0,1). A butterfly with the identifier k was chosen at random. Following (6).

$$D = |2.rand(\,).X_k^t - X_i^t| \quad (6)$$

7) If I represents the butterfly in question, k represents a butterfly chosen at random, and rand() produces a random integer between 1 and k. (0,1).

8) It is not possible to predetermine the value of the step parameter in (4). As shown in (7), we use a diminishing method. With increasing iterations, step reduces from 1 to step e. Greater worldwide searching capacity and variety may be achieved in the first stages with a greater step value. Using a reduced step value in the latter stages prevents excessive leaping and improves convergence. The method might help the suggested algorithm strike a good balance between its exploratory and exploitative capabilities..

$$step = 1 - (1 - step_e).\frac{E}{\max E} \quad (7)$$

where E is the current sum of evaluations and maxE is the supreme possible sum of evaluations.

F. Classification for IDS

1) Support Vector Machine (SVM): In [24], the SVM, is presented with a collection that have been used for the problems of function determination. The kernel function of the mathematical system has been employed for data transformation in the SVM model. When real-world SVM datasets are transformed into high-dimensional feature space, a hyper-plane is produced using the training datasets. The optimal linear hyper-plane has been used for differentiation of the true output space. It was also used to divide information into two groups, such as "non-flood" and "flood" in the case of flood susceptibility (0, 1). The effectiveness of a (SVM) model is highly sensitive to the selection of appropriate kernel functions, such as the kernel (LN). The RBF is chosen as a benchmark kernel function because, according to several studies, it is the most effective of the many kernel functions considered for flood susceptibility models. Radial basis kernel has been widely used for modelling attack susceptibility because of its versatility in handling datasets of varying dimensionality and superior generalisation capabilities..

2) Logistic Regression: Logistic regression (LR) is employed because it gives a straightforward equation for dividing issues into binary or many classes; this is especially useful when categorising attacks on the basis of a large feature set with a binary output (attack or no attack). In order to get the highest accuracies from the LR model, we tuned its hyperparameters to achieve optimal performance across all datasets. The logistic regression hypothesis function may be described in mathematical terms as follows.:

$$h_\theta(X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X)}} \quad (8)$$

The goal of logistic regression is to minimise the cost function, which is accomplished by transforming the output to a function. How to Work Out the Cost Function

$$Cost(h_\theta(x), y) = \begin{cases} \log(h_\theta(x)), & y = 1 \\ -\log\left(1 - h_\theta(x)\right), & y = 0 \end{cases} \quad (9)$$

3) Multilayer Perceptron: One kind of artificial neural network is the multilayer. Although it is possible to use only three layers in an MLP, we have found that experimenting with a larger number of layers and more finely adjusted parameters yields the most accurate predictions. Following is a function representation [31] of a simple multi-layered perceptron model with one hidden layer:

$$f(x) = g(b^{(2)} + W^{(2)}(s(b^{(1)} + W^{(1)}x))) \quad (10)$$

The activation functions are denoted by, the bias vectors by b((1)) and b((2)), and the weight matrices by W((1)) and W((2)). Our model has three hidden layers and uses the ReLU activation function and Adam solver..

4) Boosted Regression Trees (BRT): As an ensemble method, the BRT model takes use of the best features of both regression trees (models that employ the recessive binary splits to provide answers) and boosting algorithms. The learning rate, the bagging rate, the complexity of the tree, the minimum number of observations at the end nodes, and the number of trees are some factors that play important roles in the BRT fitting. The BRT has some benefits over other predictive methods, including I the ability to manage different types of predictor variables, (ii) the enhancement of missing or lost data, (iii) the elimination of the necessity to convert or delete outlier data.

5) Adaptive Voting Algorithm: Here, we present an adaptive voting method that may combine the best features of existing algorithms. Algorithm 2 demonstrates how to train many classifiers (weak classifiers) using the same training data and then combine them into a single, more robust classifier. The method's central idea is to compute the weight w ij of a classifier algorithm for a class of data, where w ij reflects the likelihood (probability) of getting the detection result under these conditions. Various data sets have different characteristics, thus the voting weight must be adjusted manually by referring to the weight value. Table III represents the pseudo code of adaptive voting algorithm

TABLE. III. ADAPTIVE VOTING

Algorithm 2 Adaptive voting
Input
$\boldsymbol{T} = \{(\boldsymbol{x_1}, \boldsymbol{y_1}), (\boldsymbol{x_2}, \boldsymbol{y_2}), \dots, (\boldsymbol{x_n}, \boldsymbol{y_n})\}, \boldsymbol{x_i} \in \boldsymbol{X_i}, \boldsymbol{y_i} \in \boldsymbol{Y}$
$\boldsymbol{Y} = \{\boldsymbol{1}, \dots, \boldsymbol{c}\}\ \boldsymbol{F} = \{\boldsymbol{f_1}, \boldsymbol{f_2}, \dots., \boldsymbol{f_m}\}$
Output: $\boldsymbol{H(x)}$
Initialize the class weights;
$\boldsymbol{w} = \begin{bmatrix} \boldsymbol{w_{11}} & \cdots & \boldsymbol{w_{1c}} \\ \vdots & \cdots & \vdots \\ \boldsymbol{w_{m1}} & \cdots & \boldsymbol{w_{mc}} \end{bmatrix}$
For $\boldsymbol{i}$ **from 1 to m**
Fit a classifier $\boldsymbol{f_i(x)}$ **to the data**
Calculate possibility for $\boldsymbol{f_i(x|c=j)}, \boldsymbol{w_{ij}} = \frac{\sum_0^c (f_i(x)==\hat{y})\&\&(y==j)}{N}$
classifier.preict(Test_x),
Compute option for one record to belong class c, $\boldsymbol{p_i} = \sum_1^m \boldsymbol{w_{ij}}(\boldsymbol{f_i(x)} == \boldsymbol{c})$
Output:
$\boldsymbol{H(x)} = \boldsymbol{argmax}(\sum_{i=1}^{i=c} \boldsymbol{p_i}(\boldsymbol{f_i(x)} == \hat{\boldsymbol{y}})$

G. Algorithmic description:

1) Train and assess the algorithms in F after optimization using test and training data.

The weight cardinality w ij is then determined by (2) calculating the training accuracy of each method against each attack type.

3) The [0, 4] type predicted results of each classifier are computed for each test record.

Fourth, choose the group that received the most votes overall and use it as your final record's anticipate outcome.

The fifth and last category should be the whole exam results.

To demonstrate how the weighted voting method really works, an example is provided.

Table IV shows that Classifier 1's detection effect is best for Type 3, whereas Classifier 3's detection is best. Table V displays the outcomes of a representative vote.

TABLE. IV. THE THREE CLASSIFIER CLASS-WEIGHTS

Class weight	Classifier 1	Classifier 2	Classifier 3
Class 1	W-11=08	W-12=0.6	W13=0.7
Class 2	W-21=0.7	W-22=0.8	W23=0.9

TABLE. V. EXAMPLES OF VOTING RESULTS.

Predict	Classifier 1	Classifier 2	Classifier 3	Voting	Result
Record -1st	Class 2 nd	Class-2nd	Class 1 st	0.7+0.8>0.7	Class 2
Record -2nd	Class 1st	Class-1 st	Class 1st	0.8+0.5+0.7	Class 1
Record –3rd	Class 3rd	Class-2 nd	Class 2 nd	0.9<0.8+0.9	Class 2
Record -4th	Class 3rd	Class-2 nd	Class 1st	0.7<0.8<0.9	Class 3

IV. RESULTS AND DISCUSSION

A. Evaluation Metrics

Efficiency may be evaluated using a number of different indices, including the F1 score, Accuracy, (AUC). All of these metrics should add up to 1 for the ideal model. This technique is used to compute a extensive variety of metrics:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN,} \quad (11)$$

$$Sensitivity = \frac{TP}{TP+FN,} \quad (12)$$

$$Specificity = \frac{TN}{TN+FP,} \quad (13)$$

$$Precision = \frac{TP}{TP+FP,} \quad (14)$$

$$Recall = \frac{TP}{TP+FN,} \quad (15)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (16)$$

where TP is the attack that has been accurately identified. TN is the non-attack that has been accurately identified. FP is the attack, which is wrongly identified as non-attack. An incorrectly identified attack goes by the designation of FN. Table VI presents the experimental analysis of proposed model with existing techniques in terms of various metrics. The existing techniques mainly focused on deep learning model, where no techniques focused on feature selection model.

Therefore, the existing techniques are implemented with ABO and results are averaged.

TABLE. VI. COMPARATIVE ANALYSIS OF PROPOSED MODEL WITH EXISTING TECHNIQUES

Model Name	Sensitivity	Specificity	F1	Accuracy	AUC
DT	0.8259	0.9841	0.8295	0.9203	0.9870
XGBoost	0.8063	0.9866	0.8286	0.9308	0.9871
RF	0.8196	0.9848	0.8286	0.9503	0.9870
SVM	0.8115	0.9860	0.8290	0.9517	0.9873
LR	0.8118	0.9858	0.8287	0.9606	0.9871
MLP	0.8192	0.9850	0.8293	0.9675	0.9870
BRT	0.8320	0.9828	0.8304	0.9701	0.9873
Ensemble Model	0.8570	0.9970	0.8821	0.9716	0.9980

In the analysis of accuracy, proposed ensemble model achieved 97%, DT and XGBoost achieved 93%, RF and SVM achieved 95%, LR and MLP achieved 96%. The existing techniques such as DT, XGBoost, RF, SVM, LR and MLP achieved 81% to 82% of sensitivity, 98% of specificity, 82% of F1 and 98% of AUC. But, BRT achieved 83% of sensitivity, 98% of specificity, 83% of F1 and 98% of AUC, where the proposed ensemble model achieved 85% of sensitivity, 99% of specificity, 88% of F1 and 99% of AUC. The reason for better performance is that the adaptive voting mechanism is used and ABO is used as feature selection. Moreover, comparing with all techniques, BRT shows better performance for attack detection. Fig. 2 to Fig. 6 presents the graphical comparison of proposed model with existing techniques.

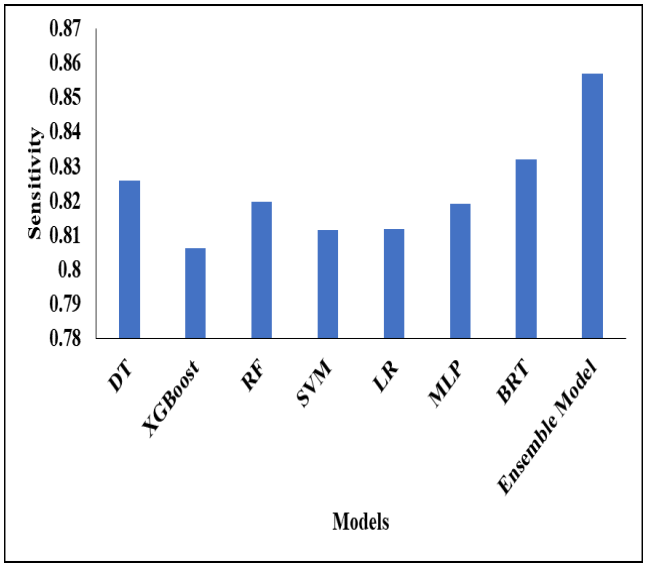

Fig. 2 Sensitivity Comparison

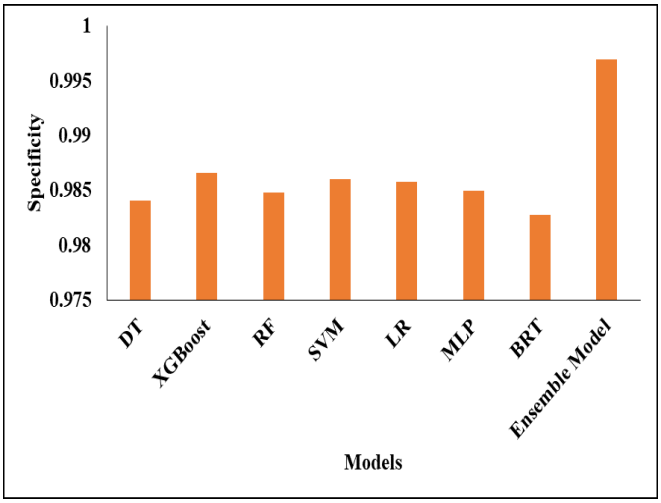

Fig. 3 Specificity Judgement

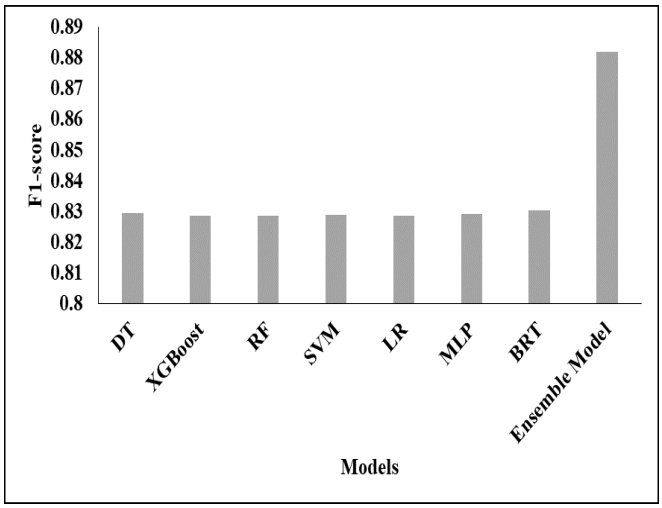

Fig. 4 F1-score judgment

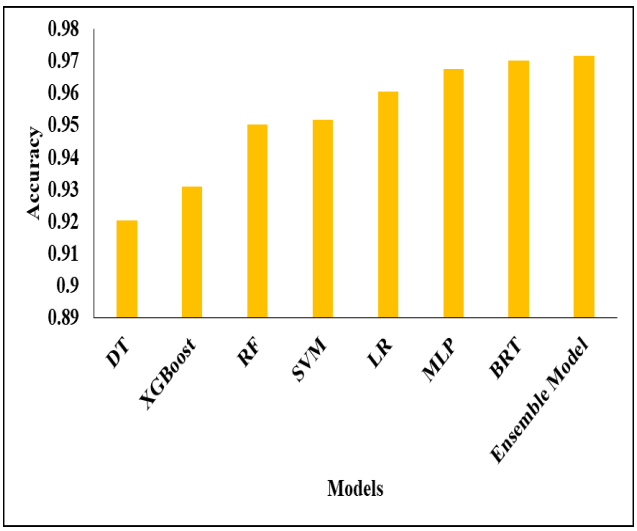

Fig. 5 Accuracy Judgement

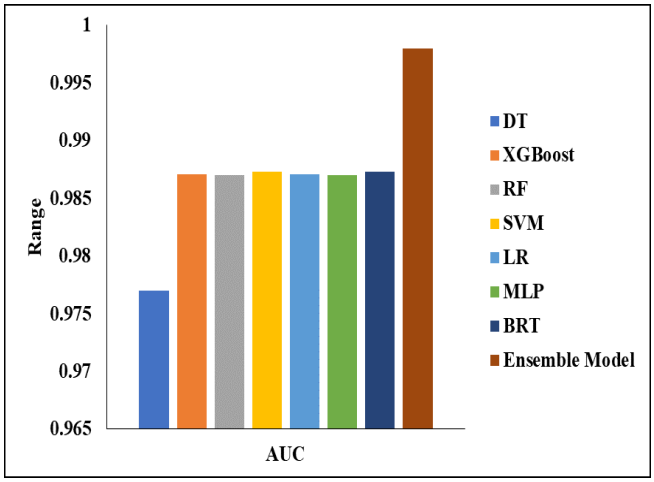

Fig. 6 AUC judgment

V. Conclusion

In this effort, we offer a unique ensemble classifier based on ABO for feature selection. The Z-score and SMOTE methods are first used to standardise the input data. After that, we employ four distinct classifiers to zero down on the most relevant attributes for categorization. Then, an adaptive voting method is used to choose the top classifier. This manner, detection rates may be boosted to better protect networks. MAT LAB is used to test the suggested model. Existing approaches are compared to the suggested model, and it is shown to have a greater accuracy of 97% than DT, XGBoost, RF, SVM, BRT, LR, and MLP. In experiments on the NSL-KDD dataset, the proposed ensemble model outdoes methods. The suggested attack detection approach will soon be enhanced by the addition of new characteristics, allowing the system to recognise a greater variety of network assaults. It's possible that future work may accommodate for different threats to boost network performance. Preventative techniques may be included to avoid not just network failure but also undesired compute cost in the face of these assaults.

References

[1] Laqtib, S., El Yassini, K. and Hasnaoui, M.L., 2020. A technical review and comparative analysis of machine learning techniques for intrusion detection systems in MANET. International Journal of Electrical and Computer Engineering, 10(3), p.2701.

[2] Hussain, M.S. and Khan, K.U.R., 2020. A survey of ids techniques in manets using machine-learning. In Proceedings of the Third International Conference on Computational Intelligence and Informatics (pp. 743-751). Springer, Singapore.

[3] V. S. Kumar, P. K. Pareek, V. H. Costa de Albuquerque, A. Khanna, D. Gupta and D. R. S, "Multimodal Sentiment Analysis using Speech Signals with Machine Learning Techniques," 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), 2022, pp. 1-8, doi: 10.1109/MysuruCon55714.2022.9972662.

[4] S. G. Gollagi, R. Srividya, G. Santhosh Kumar and P. K. Pareek, "A New Method of Secure Image Encryption by Using Enhanced RSA Algorithm," 2021 International Conference on Forensics, Analytics, Big Data, Security (FABS), 2021, pp. 1-5, doi: 10.1109/FABS52071.2021.9702550.

[5] Chandramma, P. Kumar Pareek, K. Swathi and P. Shetteppanavar, "An efficient machine translation model for Dravidian language," 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2017, pp. 2101-2105, doi: 10.1109/RTEICT.2017.8256970.

[6] V. G. Biradar, P. K. Pareek, V. K. S and N. P, "Lung Cancer Detection and Classification using 2D Convolutional Neural Network," 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), 2022, pp. 1-5, doi: 10.1109/MysuruCon55714.2022.9972595.

[7] C. C, P. K. Pareek, V. H. Costa de Albuquerque, A. Khanna and D. Gupta, "Deep Learning Technique Based Intrusion Detection in Cyber-Security Networks," 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), 2022, pp. 1-7, doi: 10.1109/MysuruCon55714.2022.9972350.

[8] P. Nandihal, V. Shetty S, T. Guha and P. K. Pareek, "Glioma Detection using Improved Artificial Neural Network in MRI Images," 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), 2022, pp. 1-9, doi: 10.1109/MysuruCon55714.2022.9972712.

[9] D. S M, R. N, S. K and P. K. Pareek, "Machine Learning based Education System with Sentiment Analysis for Students," 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), 2022, pp. 1-6, doi: 10.1109/MysuruCon55714.2022.9972555.

[10] Parameshachari, B. D., Rajashe Karappa, KM Sunjiv Soyjaudah, and Sumithra KA Devi. "Partial image encryption algorithm using pixel position manipulation technique: The smart copyback system." In 2014 4th international conference on artificial intelligence with applications in engineering and technology, pp. 177-181. IEEE, 2014.

[11] C. C, P. K. Pareek, V. H. Costa de Albuquerque, A. Khanna and D. Gupta, "Deep Learning Technique Based Intrusion Detection in Cyber-Security Networks," 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), 2022, pp. 1-7, doi: 10.1109/MysuruCon55714.2022.9972350

[12] Venkatasubramanian, S., 2021. Multistage Optimized Fuzzy Based Intrusion Detection protocol for NIDS in MANET. INTERNATION AL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY, 8(6), pp.301-311.

[13] Meddeb, R., Jemili, F., Triki, B. and Korbaa, O., 2022. A Deep Learning based Intrusion Detection Approach for MANET.

[14] Makani, R. and Reddy, B.V.R., 2022. Designing of Fuzzy Logic-Based Intrusion Detection System (FIDS) for Detection of Blackhole Attack in AODV for MANETs. In Cyber Security and Digital Forensics (pp. 113-128). Springer, Singapore.

[15] N. A. Prasad and C. D. Guruprakash, "An ephemeral investigation on energy proficiency mechanisms in WSN," 2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Tumkur, 2017, pp. 180-185.

[16] P. N and C. D. Guruprakash, "A Relay Node Scheme for Energy Redeemable and Network Lifespan Enhancement," 2018 4th International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Mangalore, India, 2018, pp. 266-274.

[17] Achyutha Prasad, N., Guruprakash, C.D., 2019. A relay node scheme of energy redeemable and network lifespan enhancement for wireless sensor networks and its analysis with standard channel models. International Journal of Innovative Technology and Exploring Engineering 8, 605–612.

[18] Achyutha Prasad, N., Guruprakash, C.D., 2019. A relay mote wheeze for energy saving and network longevity enhancement in WSN. International Journal of Recent Technology and Engineering 8, 8220–8227. doi:10.35940/ijrte.C6707.098319.

[19] Achyutha Prasad, N., Guruprakash, C.D., 2019. A two hop relay battery aware mote scheme for energy redeemable and network lifespan improvement in WSN. International Journal of Engineering and Advanced Technology 9, 4785–4791. doi:10.35940/ijeat.A2204.109119.

[20] Arora, S. and Singh, S., 2019. Butterfly optimization algorithm: a novel approach for global optimization. Soft Computing, 23(3), pp.715-734.

[21] E. L. Yu, P. N. Suganthan, Ensemble of niching algorithms. Inform. Sci. 180.15(2010):2815-2833

[22] Perveen, Nazil, Dinesh Singh, and C. Krishna Mohan. "Spontaneous facial expression recognition: A part based approach." In 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 819-824. IEEE, 2016.

[23] Tao, P., Sun, Z. and Sun, Z., 2018. An improved intrusion detection algorithm based on GA and SVM. Ieee Access, 6, pp.13624-13631.