International Journal of System Design and Computing (IJSDC)
Volume 02, Issue 02, July - December (2024)

RESEARCH ARTICLE

NETWORK INTRUSION DETECTION SYSTEM WITH AN EDGE BASED HYBRID FEATURE SELECTION APPROACH

A. Biju [1,*], and S. Wilfred Franklin [2]

[1] Department of Computer Science and Engineering, Maria College of Engineering and Technology, Attoor, Thiruvattaru, Tamil Nadu 629177 India

[2] Department of Electronics and Communication Engineering, C.S.I Institute of Technology, Thovalai, Kanyakumari, Tamil Nadu 629302 India.

[*]*Corresponding e-mail: bijubijua@gmail.com*

Abstract – An intrusion detection system looks through network data to find both legitimate and malicious activity. This study can detect new attacks, which is especially useful in IoT situations. Deep Learning (DL) has demonstrated its superiority in solving challenging real-world issues such as NIDS. This method, however, necessitates more processing resources and takes a lengthy time. During a classification process, feature selection is critical in selecting the best attributes that best describe the goal concept. A novel Network intrusion detection (NEST) technique has been proposed to develop an improved edge-based hybrid feature selection approach, which is a deep learning method for detecting malicious traffic. The Enhanced BPSO technique overcomes the difficulty of BPSO feature selection by combining Binary Particle Swarm Optimization (BPSO) and correlation– based (CFS) traditional statistical feature selection. Three intrusion detection module having three classifiers make up the proposed system. The Signature Detection Module (SDM) examines threats and classifies it as unknown, normal, or intruder based on matching signatures utilizing the Generalized Suffix Tree (GST) algorithm. The Anomaly Detection Module (ADM) employs deep Q-learning to detect unknown attacks. The Hybrid Detection Module (HDM) employs the Meta-AdaboostM1 algorithm. Results of simulation show that the proposed protocol increases Detection Rate, computation time, and False Alarm Rate when compared to the existing XGBOOST-DNN, HNGEA, and HDLNIDS methods. The detection rate of the proposed NEST method is 4.55%, 6.47%, and 10.32% higher than the existing XGBOOST-DNN, HNGEA, and HDLNIDS, techniques respectively.

Keywords *– Network Intrusion Detection System, Deep Learning, Generalized Suffix Tree, Q-learning, Meta-AdaboostM1.*

1. INTRODUCTION

Network intrusion detection systems (NIDS) have really been successfully advanced in academia and business in response to the increasing number of cyber-attack on government and commercial enterprises around the world [1]. The annual cost of cybercrime is gradually rising. The most dangerous cyber-crimes include malicious insiders, dos attack, and web- based attacks [2]. If harmful malware infiltrates a system, it may result in the loss of intellectual property, as well as the disruption of a country's critical national infrastructure [3]. Businesses utilise antivirus software, and IDS to safeguard computer systems from unauthorized access (NIDS) [4].

As a product of the expanding quantity of data and the increased need edge computing is becoming popular for processing data a critical crossroads in history [5]. Edge computing increases service stability and delivers artificial intelligence services for terminal devices and data that are rapidly developing Smart terminals and other edge computing devices are near to the source of data [6]. It responsible for processing data at the network's edge. Near-end service. and also, proximity and location awareness, can benefit users [7]. It is fast, better real-time, and much more secure in terms of information processing [8]. It could also aid in the reduction of expenditures and alleviation of network bandwidth congestion caused by cloud computing's high energy usage. Manufacturing, energy, smart homes, and transportation are just a few of the industries that use edge computing [9,10].

Any device, software, or hardware capable of detecting suspicious behaviour or preset threats and then implementing appropriate countermeasures will be considered an intrusion detection system (IDS) [11]. These devices have developed into crucial tools for detecting and protecting a network in the previous stages. Every day, new intrusions are detected, causing increasing amounts of damage and potentially affecting a company's information system's operation [12]. Today's packet filtering systems look for signals of malicious activity or unauthorized access in packets sent over the network. In packets transferred over the network, IDS systems seek for evidence of malicious activity or unauthorized and undetected access [13]. As computer attacks get more sophisticated and identification of breach has become more complex, this problem has emerged as one of the most important challenges in the world of computer

security. To overcome these issues, a novel Network intrusion detection (NEST) technique has been proposed. The following are the primary contributions.

- Initially the collected data are preprocessed using data conversion and data normalization technique.
- The study introduces an improved edge-based hybrid feature selection approach that combines BPSO and CFS for the improved selection of relevant features for detecting malicious traffic, optimizing the trade-off between processing resources and classification accuracy.
- The IDS is structured into three distinct modules: SDM, ADM, and HDM. Each module specializes in different aspects of intrusion detection, such as signature matching, anomaly detection using deep Q-learning, and hybrid classification using Meta-AdaBoostM1, respectively.
- Extensive simulations using the NSL-KDD dataset authorize the effectiveness of the proposed method.

The explanation that follows concerns the next half of this study: The literature is consulted in Section II to assess the research. In Section III, the proposed system is explained in full. The conclusion is found in Section V, whereas the result and discussion are found in Section IV.

2. LITERATURE SURVEY

In 2019, Papamartzivanos, et al. [14] suggested an innovative approach that delivers a climbable, self-adaptive, and independent abuse by uniting the advantages of self-taught erudition with MAPE-K frameworks. IDS. The experimental findings show that our approach can revitalize the IDS and eliminate the requirement for manual training set refreshes. The suggested approach is assessed using a number of classification metrics, and the results show that under crucial circumstances where a statically trained IDS is rendered useless, the ADR of the IDS can rise to 73.37%.

In 2020, Devan et al., [15] suggested XGBoost–DNN method uses a deep neural network to classify network invasions after applying the XGBoost algorithm for feature selection. The suggested framework is authorized by cross-validation, and its performance is related with well-known shallow ML methods such as SVM, naïve, and Bayes logistic regression. A deep learning model consistently outperforms prior models in terms of classification accuracy, as evidenced by the reported findings.

In 2020, Venkatraman, and B. Surendiran [16] suggested adaptive hybrid IDS using a controller technique for timed automata. The experimental findings demonstrate the suitability of suggested method, for smart city applications. It also demonstrates its accuracy (99.06%) in identifying various types of attacks in IoT environments, including replay, zero-day, and DoS attacks.

In 2020, Elhefnawy, et al. [17] proposed a framework for Hybrid Nested Genetic-Fuzzy Algorithms (HNGFA) to provide security experts with highly optimized outputs for the classification of major and small danger categories. The findings demonstrate that, in various configurations on complicated datasets, the HNGFA performs better than alternative methods in terms of detection, investigation, and dynamic regulations for all minor attack types with excellent precision.

In 2021, Seo, and Pak, [18] suggested a two-level intrusion detection method that has a high finding accuracy. The level 1 classifier first extracts a limited set of features from the packet to facilitate rapid organization and real-time threat detection. Since the level 2 classifier only handles flows that the level 1 classifier was unable to classify, the traffic is manageable by a labor-intensive machine learning-based classifier.

In 2023, Qazi, et al. [19] suggested a system HDLNIDS for identifying network intrusions. Experiments are conducted assessing the suggested method's efficacy using publicly available benchmark CICIDS-2018 data. The study's conclusions show that with an average accuracy of 98.90%, HDLNIDS performs better than existing intrusion detection techniques at identifying malicious attempts.

In 2023, Hnamte, et al. [20] suggested a cutting-edge, two-stage deep learning method for attack detection that combines AE with LSTM. It can analyze network activities and makes use of a very effective framework. The desired LSTM-AE's ideal network parameters are found using the CICIDS2017 and CSE-CICDIS2018 datasets. The outcomes of the experiments prove the efficiency of the suggested hybrid model and its applicability in identifying attacks in contemporary settings.

3. NETWORK INTRUSION DETECTION (NEST)

In this section, a novel Network intrusion detection (NEST) technique has been proposed for effective intrusion detection in the network environment. Initially methods for data conversion and data normalization are used to preprocess the gathered data. In order to optimize the trade-off between processing resources and classification accuracy, the paper presents an enhanced edge-based hybrid feature selection strategy that combines BPSO and CFS for the improved selection of relevant features for identifying hostile traffic. SDM, ADM, and HDM are the three main modules that make up the IDS. Aspects of intrusion detection that each module focuses on differently include signature matching, hybrid classification using Meta-AdaBoostM1, and anomaly detection using deep Q-learning. The efficacy of the suggested strategy is authorized by extensive simulations with the NSL-KDD dataset. Figure 1 shows the framework of proposed methodology.

3.1. Data Collection

The dataset is exclusively responsible for testing, inspecting, and evaluating the way the discovery scheme behaves, and it plays a critical part in obtaining a better outcome. A high-performance one could deliver beneficial results not only for an offline device, but also in a real-world setting. The majority of the writers used the NSL-KDD datasets, which are a better version of the KDD CUP 99 database that eliminates duplicate data and selects articles based on their proportion. During pre-processing, it contains

148,517 document each with 41 attribute and a class mark. The five types are DoS, U2R, probing, R2L, and normal.

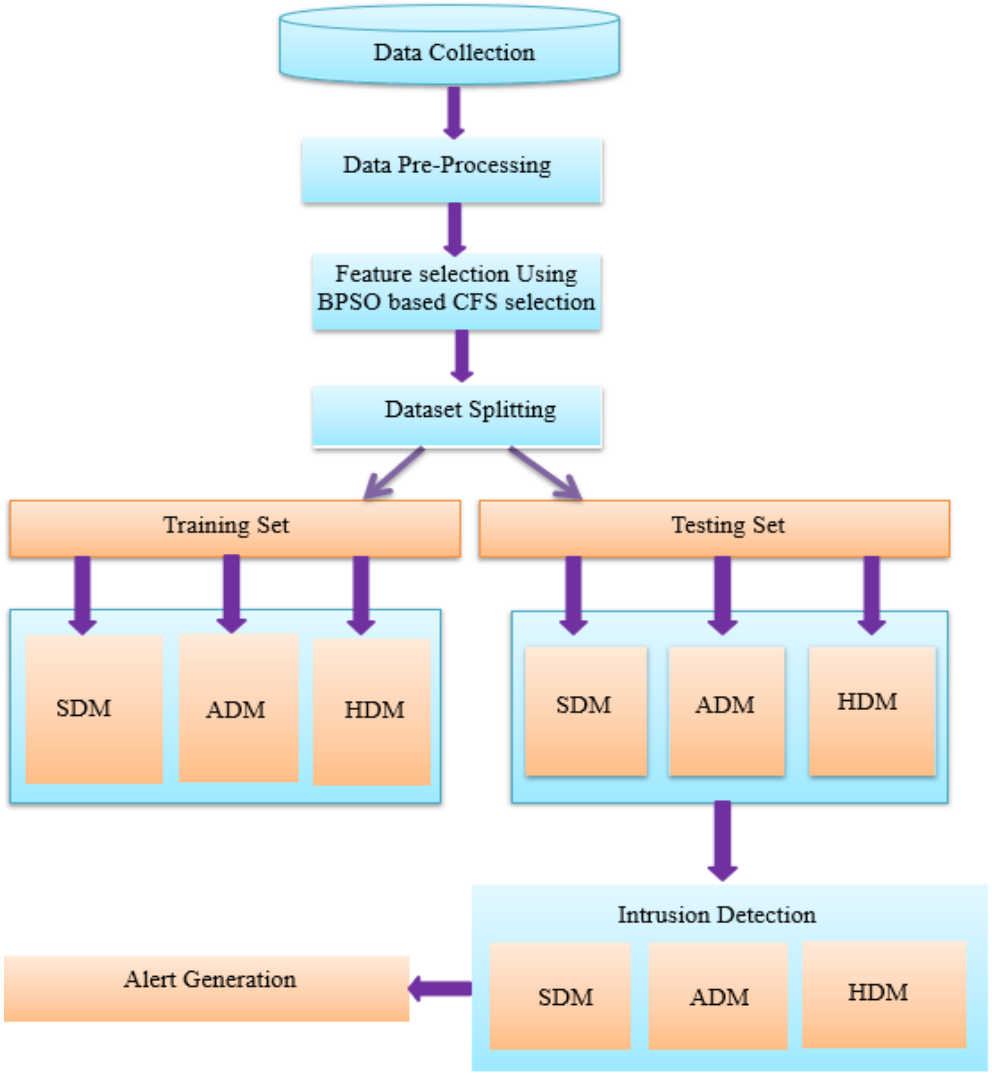

Figure 1. Architecture of proposed NEST methodology

3.2. Data Pre-Processing

The two primary methodologies for data pre-processing are data conversion and data normalisation. Data conversion converts components of traffic from nominal to numeric to guarantee that almost all data has numerical for processing by intrusion-detection system. Data normalisation is used to condense a vast range of values into a manageable range. Furthermore, during normalisation, null values are deleted. To normalise high numbers and decrease their significance, we utilise a minimum–maximum scaling method to put values between 0 and one, as shown by the equation (1)

$$f_{ij} = \frac{f_{ij} - min(f_{ij})}{max(f_{ij}) - min(f_{ij})} \quad (1)$$

where , f_{ij} is the value of the feature in the dataset matrix's row i and column j.

3.3. Feature selection and ranking

For the purpose of feature selection, the cross-validation technique was utilised to see how well the statistical analysis results might be generalised to an in-dependent dataset. The Enhanced BPSO approach is used to focus on feature selection using swarm intelligence (BPSO). Because of the obstacles posed by Big Data, feature selection in intrusion detection improves classification performance by minimising computational operations. When opposed to normal KDD data sets for Intrusion Detection, feature selection plays a key role with great quality real - world data sets due to the selection of only the most connected features with defined classes. The collected data required to be separated into training and testing data in order to analyse those produced models.

3.3.1. Correlation–Based Feature Selection

CFS is a filters method whose main goal is to discover the optimal solution in search space by evaluating the relevancy of an extracted features to a class and the redundancy between selected subset of features. The features are chosen based on the correlation function's feature subset assessment result. This indicates that selected characteristics are most closely related to the class, and not to one another.

As shown in Equation (2) every feature with only a highest score predicts classes in the subspace as well as other characteristics.

$$CS = \frac{fdef}{\sqrt{f+f(f-1)d_{ff}}} \quad (2)$$

Where CS is the score for a s extracted features with f features, dcf is the mean level of similarity among features and the class label, and dff is the degree of inter-correlation mean between qualities, d_{ff} is the degree of inter-correlation mean with both features, and d_{ff} is the degree of inter-correlation mean between features, and d_{ff} is the degree of inter-correlation mean between features, and d_{ff}. A correlation approach known as feature subsets is used to assess CFS. Larger dcf or smaller d_{ff} result in greater evaluation value in specific subsets.

3.3.2. Binary Particle Swarm Optimization (BPSO)

BPSO is a variant of the PSO algorithm adapted for binary search spaces. It's particularly useful for feature selection in ML, where the goal is to identify a subset of features that leads to the best performance of a model. As illustrated in Equations (3) & (4), each particle in PSO adjusts its rate of change and location in every iteration depending on personal experiences (pbest) and the swarm's greatest experience (gbest) (3). The performance of all particles is measured using defined cost functions at the end of every iteration.

$$V_j[st+1] = W \times V_j[st] + F1d1\big(P_{jbest}[st] - P_j[st]\big) + F2d2\big(G_{best}[st] - P_j[st]\big) \quad (3)$$

$$P_j[st+1] = P_j[st] + V_j[st+1] \quad (4)$$

Each particle j is iterated at each iteration. Obtain three vectors of length N that represent the problem dimension: velocity, location, and personal best. The end condition is accomplished when the enhanced size of the global finest is less than the stop value () or when the maximum iteration count is reached, and PSO stops.

The number of populations to implement the BPSO is set at 100, the number of iterations is set at 10. Initialize swarm at random, with $X = (x1, x2, \ldots, xn)$ representing a particle as a feature vector and y [0,1] signifying a class label, with 0,1 and 1, respectively, corresponding to normal and abnormal. Then add the following value in the variable:

W stands for the inertia weight, which regulates the particle's velocity impact on present iteration and is normally between [0.4,0.9]. The acceleration coefficients F 1 and F 2 are constants with a range of [0.5]. while d 1 and d 2 are random counts in the range [0,1]. On the velocity changes, these parameters scale both personal and swarm knowledge. As a result, as stated in Equation (5), utilise the Activation Function to estimate each particle's fitness value, and then choose the particle with the best value.

$$F(X) = \alpha(1 - Pr) + (1 - a)\left(1 - \frac{N_s}{N_v}\right) \quad (5)$$

N_s is the size of the feature subset that was tested, while N_s denotes the total number of input variables available. Pr is a metric measuring how well a classifier performs. Total accuracy is represented on the left side of the equation, while the proportion of utilised features is represented on the right. Regular PSO equations are converted to operate in binary space to create BPSO. In addition, the sigmoid function in Equation (6) was utilised to convert $V(st+1)$ to the [0,1] range. In BPSO, the velocity vector represents the probability of a component in the position vector taking value 1, which is given in equation (7).

$$S(V\,j^{st\cdot 1}) = \frac{1}{1+e^{-(Vj^{st-1})}} \quad (6)$$

$$Pj^{st+1} = \begin{cases} 1 & if\ rand() < S(Vj^{st+1}) \\ 0 & otherwise \end{cases} \quad (7)$$

where $rand$ () is a value chosen at random from the range [0,1]. PSO then generates an optimum solution, which is the best vector globally, and checks for the stop condition, which must be met for PSO to exit.

i. Enhanced BPSO features based on CFS selection

CFS classical statistical approach has been developed to progress the feature selection of standard BPSO Algorithm. Algorithm 1 shows the pseudocode for improved BPSO algorithm. The following approach is used to implement the Enhanced BPSO Algorithm:

1. Using a CFS-based correlation equation, calculate a score for each attribute.
2. Set a threshold and then choose every feature that are greater than it.
3. Run BPSO on a subset of features you've chosen.
4. Feature selection to eliminate features and select the best group of features

Algorithm 1 The Improved BPSO Algorithm

Input: Training and testing sets;

output: Xbest subset of characteristics begin

initialize N particle population Xj best =(X1,...,DT), j=1,2,3,…..,N

set up fit(Xj) and Xbest.

While j=max number of iterations,

initialise solution storage, use Fit temp (j) and X temp (j). update Xj and Vj

for $\boldsymbol{j = 0\ to\ n}$.

if rand () returns Vj select Xj from X.

if Calculate Fit returns true create a new X new

end (X new)

temper (j) = temper (X new)

If Fit ($X\ j$) equals Fit (X new), then (X new) If Fit (X new)>=max Fit temp,

then X best = X new

end if end

3.4. Signature Detection Module (SDM)

SDM is used to identify all attack patterns by matching signatures maintained in the system. Position Aware Distribution Signature (PADS) is used to construct the signatures. A Generic Suffix Tree (GST) is used to keep the signature in the repositories, that might check signature in a time that is asymptotically ideal. The signature repository is checked using $O(m+n)$ if the signatures are m bytes long. In the SDM, the Light-Net technique is employed to detect network attack. Continuous weight networks make up the LightNet. The most of the weights are zero, and those that aren't can only be -1 or +1. Light Net uses synaptic pruning training to construct the active function, which is represent by odd or hyperbolic tangent expression. As an outcome, the arbitrary location is specifically defined in equation (8)

$$\mathrm{AT} = \tanh(x-p) + \tanh\left(-(x-p) + \sigma\right) \quad (8)$$

There are three layers to Light Net: Hidden, input, and output. Packet feature is considering input, and the hidden layer HMS is also clustering related packet feature. SDM in the HDM detects threats by comparing signature in the tree. Intruder packets are reported, and an ADM examines anomalous packets to determine the sort of assault.

3.5. Anomaly detection Module (ADM)

A deep Q-learning algorithm analyses SNR and bandwidth characteristics classifies assaults as DoS, U2R , or (R2L) in the ADM. The agent learns about its surrounds by Q-learning, which results in a Q-table matrix containing states and action. In contrast, Q-learning is best suited to small-scale situations, while the Internet of Things is a vast system. As a result, Q-learning and deep learning are combined to build a Deep Q-learning system capable of simultaneously processing several unexpected attack packets. Each packet is an input to the input layer of the Deep Q-learning algorithm, containing bandwidth and SNR. The best classification technique is deep learning, which works well with big data sets. Because of the grouping of these characteristics, deep Q-learning can apply categorization. Let's call the states ($S1$, $S2$, $S3$, … St) and ($A1$, $A2$, $A3$, … At) correspondingly. In Deep Q-learning, the Q-value is calculated using the following given equation (9)

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma maxQ(S_{t+1}, a) - Q(S_t, A_t) \quad (9)$$

According to the attack detection decision, R_{t+1} is specified as a reward by 1 on each timestep. Deep Q- learning uses an epsilon-greedy policy to carry out the activities. The suggested approach uses the packet's SNR and bandwidth, as well as other critical packet properties, to detect four potential assaults (DoS, Probe, U2R, and R2L) that aren't described in SDM. Algorithm 2 shows the pseudocode for Q-learning algorithm.

Algorithm 2. Q-learning

Begin

For all (St, At);

If S is terminal then

Compute new initial state from the reward Else

S← S;

Return attack type{DoS,Probe,U2R,R2L} End

The traffic's byte frequency distribution is determined and compared to the regular traffic distribution. A huge discrepancy is seen as unusual. The signature length w is positioned in relation to the byte frequency and percentage in an anomalous signature, where W is the signature width in bytes. It is categorised as Dos, Probe, U2R, R2L, or normal using ADM.

3.6. Hybrid Detection Module (HDM)

The categorization technique in the previous modules (SDM and ADM) was shown to have significant problems. When anticipating unexpected sample, the SDM has been unable to recognize unexpected classes, but the ADM has a higher probability of false alarms. As a result, the Hybrid technique was employed to achieve a balance between detection precision and false alarm rate. This method involves training several classes on the same dataset and then blending the results. It is thought that the Meta-AdaBoostM1 approach improves detection precision. AdaBoost is a boosting method for creating new classifier that look for and focus on examples that were previously misinterpreted by a classifier. Typically, this method uses the training data to train a weak classifier. It uses a decision stump for the weak classifier. A decision tree with only one level is called a decision stump. One internal node (root) connects all of the terminal nodes together. Using the same training dataset, the weak classifier is retrained with modified weights for precise classification. Reclassifying the weak classifier is done using a strong classifier. One powerful classifier that is employed is the meta-AdaboostM1 method. One powerful classifier that is employed is the meta-AdaboostM1 method. Algorithm 3 shows the pseudocode for Q-learning algorithm.

Algorithm 3 HDM Meta adaBoostM1 Algorithm

Input to the procedure: (Data sample) (D))

Output: $H(x)$ = sin($\sum T$ $\alpha ihi(x)$) is the final Hypothesis:

Function: Initialization of the weight $D(i)$ = 1 for i=1 to m.

Begin

For each class i=1 to m over distribution $D(i)$ do Train the weak classifier $D(i)$

Calculate Hypothesis h(i). Weak hypothesis (i).

Return final hypothesis

End

4. RESULTS AND DISCUSSION

The approach was implemented using the Python programming language platform. The proposed NEST approach was analyzed by applying measurements and compares to certain other current models that use DL and hybrid rule-based models. The numbers of correct and wrong outputs were totalled and compared with results of the

reference in a categorization exercise. precision, Accuracy, specificity, recall, and F1-score are among the most commonly used matrices. The NSL-KDD collection contains 77,054 normal records and 71,460 assault documents. The proposed NEST model's effectiveness is contrasted with existing XGBOOST-DNN [15], HNGEA [17], and HDLNIDS [19] methods

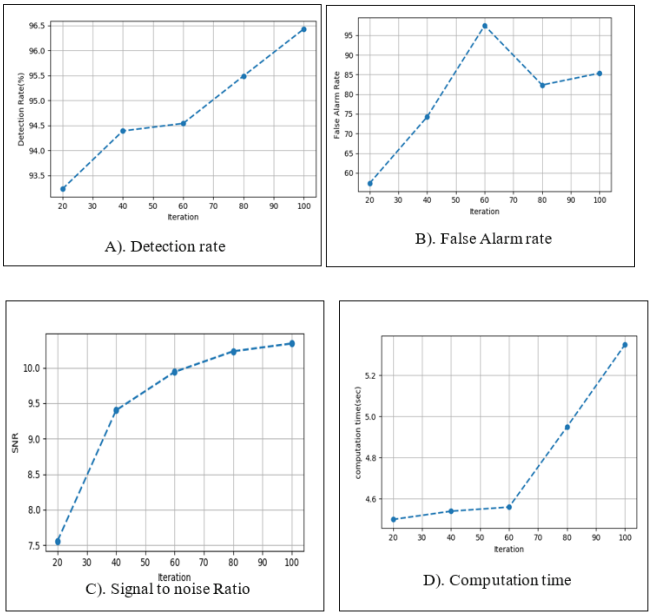

Figure 2. Performance of the BPSO and CFA methods with the GST, Q-learning, Meta- AdaboostM1 classifier for NSL-KDD dataset: A). Detection rate, B). False alarm rate C). Signal to noise Ratio D) Computation time.

Figure 2 displays the Performance of the BPSO and CFA methods with the GST, Q-learning, Meta- AdaboostM1 classifier for NSL-KDD dataset. All detection rates, false alarm rates, signal to noise ratio, specificity, F-measure, and computation time criteria decrease as the number of features grows. Furthermore, when compared to the results obtained using the BPSO and CFA methods, the proposed NEST strategy outperforms them in all classifiers. Furthermore, the use of 20 features produces the best results when compared to alternative numbers of features.

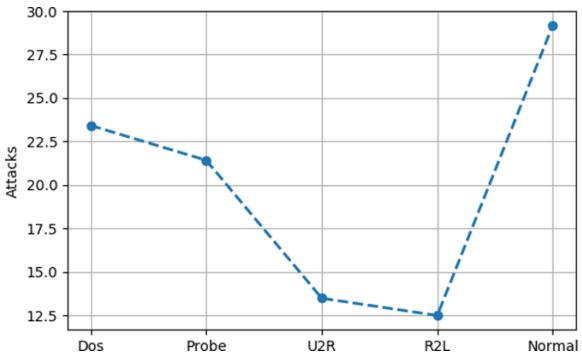

Figure 3. NSL-KDD dataset with Five classes

Figure 3. shows the attack records. To determine the types of records such as DoS, normal, U2R, R2L, and probe, both datasets showed overall higher performance for intrusion detection, even though some results, such as U2R, are not very high.

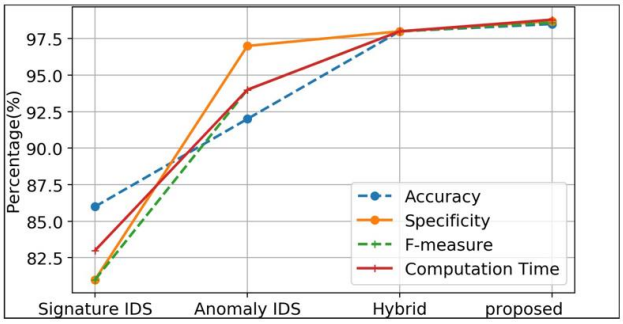

Figure 4. Performance comparison

Figure 4 displays the performance evaluation of our proposed NEST method's three elements.

Table 1. Classes of training and testing: normal and attack

Attack/Normal class	NSL-KDD Train+	NSL-KDD Test+
Normal	67354	7653
DoS	45654	8766
Prob	12237	2537
R2L	679	3678
U2R	34	200
Total	125958	22834

These three subsystems are hybrid detection, anomaly-based, and signature-based intrusion detection systems. The findings of three module employed in the proposed NEST method are shown in Fig. 4. The results also showed that our suggested NEST method outperformed the other three modules in terms of performance (SDM, ADM and HDM). Table 1 shows the Normal and attack class of training and testing set

It's a labelled flow-based dataset that was used to evaluate anomaly-based intrusion detection. There are both normal and attack classes in it. A traffic record is classed as normal, aggressor, unknown, suspicious, or victim in every occurrence.

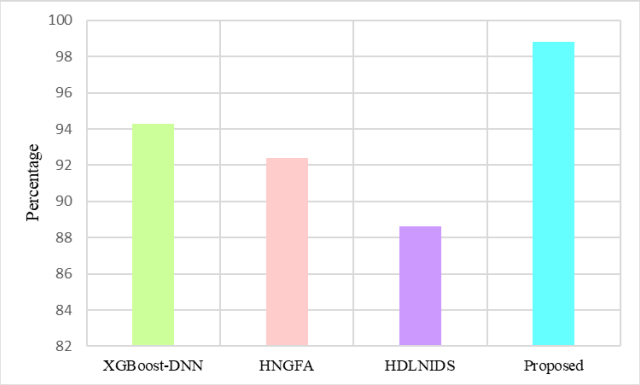

Figure 5. Comparison in terms of detection rate

Figure 5 illustrates a performance comparison of the detection rates between the proposed NEST technique and the existing methods: XGBOOST-DNN [15], HNGEA [17], and HDLNIDS [19]. It assesses how effectively each system or procedure can identify or locate the intended target among all possible targets. The proposed NEST technique demonstrates a higher detection rate compared to the current methods, with improvements of 4.55%, 6.47%, and 10.32% over XGBOOST-DNN, HNGEA, and HDLNIDS, respectively.

5. CONCLUSION

This research discovered a number of intrusion detection issues that put the availability, integrity, and confidentiality of mobile edge networks at jeopardy. To address current intrusion detection difficulties, this work developed a NEST approach for a mobile edge computing environment. Several detection modules using various classifiers make up the suggested detection system. The traffic packets next enter the hybrid IDS phase, which uses signature matching and the GST algorithm to implement SDM. The ADM processes all strange packets, and the deep Q-learning algorithm uses SNR to identify assaults. The detection rate of the proposed NEST method is 4.55%, 6.47%, and 10.32% higher than the existing XGBOOST-DNN, HNGEA, and HDLNIDS, techniques respectively. The ADM is found to outperform earlier IDS approaches after data analysis. This ADM system should be expanded in the future to incorporate the following: Include other major attacks in other datasets and use deep learning methodologies with optimization to test the network's performance. Our technology might be connected to the Instruction Prevention System (IPS), which would automatically protect against deep learning-based assaults. Assure that redirected IoT traffic originates from a registered or unregistered user, and authenticate separate safety using biometric and other authentication mechanisms.

CONFLICTS OF INTEREST

Not applicable.

FUNDING STATEMENT

Not applicable.

ACKNOWLEDGEMENTS

The author would like to express his heartfelt gratitude to the supervisor for his guidance and unwavering support during this research for his guidance and support.

REFERENCES

[1] Z. Azam, M.M. Islam, and M.N. Huda, "Comparative analysis of intrusion detection systems and machine learning based model analysis through decision tree", *IEEE Access.* 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] Ö. Aslan, S.S. Aktuğ, M. Ozkan- Okay, A.A. Yilmaz, and E. Akin, "A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions", *Electronics,* vol. 12, no. 6, pp.1333. 2023. [CrossRef] [Google Scholar] [Publisher Link]

[3] S. Vesic, and M. Bjelajac, "Cyber security of a critical infrastructure", *Law Theory & Prac.*, vol. 40, p.77. 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] D.P. Möller, "Intrusion detection and prevention," In *Guide to Cybersecurity in Digital Transformation: Trends, Methods, Technologies, Applications and Best Practices*, pp. 131-179, 2023. Cham: Springer Nature Switzerland. [CrossRef] [Google Scholar] [Publisher Link]

[5] R. Kumar, K.S. Sangwan, C. Herrmann, and S. Thakur, "A cyber physical production system framework for online monitoring, visualization and control by using cloud, fog, and edge computing technologies", *International Journal of Computer Integrated Manufacturing*, vol. 36, no. 10, pp.1507-1525, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] M. Zhu, M. Liang, H. Li, Y. Lu, and M. Pang, "Intelligent acceptance systems for distribution automation terminals: an overview of edge computing technologies and applications", *Journal of Cloud Computing*, vol. 12, no. 1, pp.149, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] M. Talebkhah, A. Sali, V. Khodamoradi, T. Khodadadi, and M. Gordan, "Task offloading for edge-IoV networks in the industry 4.0 era and beyond: A high-level view," *Engineering Science and Technology, an International Journal*, vol. 54, pp.101699, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] Z. Lv, and W. Shang, "Impacts of intelligent transportation systems on energy conservation and emission reduction of transport systems: A comprehensive review", *Green Technologies and Sustainability*, 1(1), p.100002, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[9] L. Islam, and M.T. Hassan, "Performance Evaluation of Vehicle-Centered Traffic Management Using Fog Computing-based Wireless Network", In *2023 26th International Conference on Computer and Information Technology (ICCIT)*, pp. 1-6, 2023. IEEE. [CrossRef] [Google Scholar] [Publisher Link]

[10] S.M. Rajagopal, M. Supriya, and R. Buyya, "FedSDM: Federated learning based smart decision-making module for ECG data in IoT integrated Edge-Fog-Cloud computing environments", *Internet of Things*, pp.100784, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] M. A. Hossain, M. S. Hossain, and R. Karim, "Comprehensive architectural network design based on intrusion detection system," *International Journal of Communication and Information Technology*, vol. *4*, no. 2, pp. 12-19, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] D. Shankar, G.V.S. George, J.N. JNSS, and P.S. Madhuri, "Deep analysis of risks and recent trends towards network intrusion detection system," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 1, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[13] M. Bhavsar, K. Roy, J. Kelly, and O. Olusola, "Anomaly-based intrusion detection system for IoT application." *Discover Internet of Things*, vol. 3, no. 1, pp.5, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] D. Papamartzivanos, F.G. Mármol, and G. Kambourakis, "Introducing deep learning self-adaptive misuse network intrusion detection systems", *IEEE access*, vol. 7, pp.13546-13560, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[15] P. Devan, and N. Khare, "An efficient XGBoost–DNN-based classification model for network intrusion detection system." *Neural Computing and Applications*, vol. 32, no. 16, 12499-12514, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[16] S. Venkatraman, and B. Surendiran. "Adaptive hybrid intrusion detection system for crowd sourced multimedia internet of things systems", *Multimedia Tools and Applications*, vol. 79, no. 5, pp. 3993-4010, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[17] R. Elhefnawy, H. Abounaser, and A. Badr, "A hybrid nested genetic-fuzzy algorithm framework for intrusion detection and attacks," *IEEE Access*, vol. 8, pp.98218-98233, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[18] W. Seo, and W. Pak, "Real-time network intrusion prevention system based on hybrid machine learning", *IEEE Access*, vol. 9, pp.46386-46397, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[19] E.U.H. Qazi, M.H. Faheem, and T. Zia, "HDLNIDS: hybrid deep-learning-based network intrusion detection system", *Applied Sciences*, vol. 13, no. 8, pp.4921, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] V. Hnamte, H. Nhung-Nguyen, J. Hussain, and Y. Hwa-Kim, "A novel two-stage deep learning model for network intrusion detection: LSTM-AE", *IEEE Access*. 2023. [CrossRef] [Google Scholar] [Publisher Link]

AUTHORS

A. Biju is an Assistant Professor in Maria College of Engineering and Technology, Attoor. He is the Head of the Master of Computer Applications Department. He received his BE and ME degrees in Anna University, Chennai, in 2006 and 2011, respectively.

S. Wilfred Franklin Professor and Head in the Department of Electronics and Communication Engineering, CSI Institute of Technology, Thovalai, Kanyakumari District, Tamil Nadu, India.

Arrived: 18.09.2024
Accepted: 21.10.2024