Natural Resources for Human Health
eISSN: 2583-1194

Original Research

Received 23 May 2026
Revised 28 June 2026
Accepted 20 July 2026

Natural Resources for Human Health 2026, Vol. 6 (10s): 912–928

KEYWORDS:
IoT networks; AI; Generative Adversarial Network (N-GAN); Recurrent Noise Injector; SHAP method.

Federated Learning Model for Network Intrusion Detection with Generative Adversarial Network and Explainable AI in Edge Environments

Macherla Malleswara Rao[1*], Pavan Kumar Tummala[2]

[1] Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, India -522302, macherlamalleswararao@gmail.com
[2] Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, India -522302, pavankumar_ist@kluniversity.in

ABSTRACT: Accelerated growth of the IoT in recent years has been driven by advancements in wireless transmission and processing. Various edge devices have appeared within IoT networks, such as smartphones, smart military equipment, smart cars, and intelligent applications. These applications are gaining popularity as they effectively tackle real-time issues. However, IoT networks are susceptible to a variety of threats because of their low processing power and resources. Therefore, in order to preserve data security and maximize network performance, efficient intrusion detection methods are required. This paper presents a federated learning model for detecting network intrusions using GAN and explainable AI within edge environments. The dataset is processed by addressing missing values and standardizing the data to enhance detection accuracy. Missing information is filled using a Recurrent Noise Injector GAN (RNI-GAN) technique, and the data is normalized with a robust scaler. Network Intrusion Detection over Generative Adversarial Network (N-GAN) is then used to identify suspicious/malicious network activity with respect to each edge device. The updates of each edge device are sent to a central server where they are aggregated to produce a better global detection model. The global model can explain its decisions through the SHAP method and make the process of detection transparent and comprehensible. The updated global model is transmitted to all the edge devices. This enables every device to improve its threat detection capabilities by relying on information containing the entire network. The system operates continuously, enabling the global model to get better over time by learning from new and changing data collected at each edge device. The proposed model attained the performance metrics of 97.19% accuracy, PPV of 97.14%, 2.81% error and F1_score 97.16% with other existing algorithm. Thus the proposed model is well to detect the intrusion detection in networks.

1. INTRODUCTION

In military contexts, IoT systems are employed across numerous mission critical functions including battlefield surveillance, unmanned aerial vehicles (UAVs), real time troops and equipment tracking, and other remote sensing related functions that require fast processing time, and fast reaction times [1]. In the era where, military operations are closely linked to complex information systems. This makes it very important to protect the security and reliability of these networks. As technology keeps improving, the military depends more on digital systems, which increases the risk of cyber threats to these systems. Intrusion

detection is an essential part of establishing a safe and reliable military environment. Conventional Intrusion Detection Systems have difficulty managing high-dimensional, complex and partial threat information while protecting user nominality [2].

To address the limitations, artificial intelligence (AI) has become a pragmatic way to enhance IDS (Intrusion Detection System) in retail military networks specifically with Deep Learning and Machine Learning algorithm. Behavioral analysis is enhanced by machine learning as a pertinent activity enables accurate and timely detection, classification, and response to cyber threats [3]. Several machine learning techniques, such as SVM, decision trees, and KNN, are used to detect attacks. However, because they may misclassify normal network activity as abnormal, these systems can sometimes result in high false positive rates [4]. Their effectiveness depends on the quality and diversity of training data, making them vulnerable to adversarial assaults as well, in which attackers alter input data to avoid detection. Edge network tend to be dynamic and create different potential anomalies that are unidentifiable by the machine learning system (it was not trained to detect them). Finally, a full tagged dataset can be arduous and time-consuming to collect [5].

To tackle this problem DL model is employed, precisely identifying the type of attack and understanding complex traffic patterns [6]. DL-based detection methods can learn feature representations from raw data and perform better with complicated and large-scale network traffic, they have drawn greater attention and are more flexible in response to various threat situations [7]. DL algorithm including CNN, RNN, and DBNs are increasingly employed for the sophisticated detection of military network-based threats [8]. Nevertheless in military network environments, anomaly detection may be limited by the substantial computing resources needed for deep learning models [9]. Obstacles that must be overcome include data privacy, architectural complexity, and the requirement for substantial computing capacity to manage big datasets. In addition to recognized attacks it is unable to identify insider threats and network intrusions in virtual networks [10]. To overcome these issues, introduced a federated learning model for detecting network intrusions using generative adversarial networks and explainable AI in edge environments. This method improves security and privacy by enabling several devices to work together.

The major contribution of the work is given below,

- A network intrusion detection model that employs generative adversarial networks and explainable AI techniques in edge environments.
- Recurrent Noise Injector GAN (RNI-GAN) is used for missing value imputation and the robust scaler for data normalization are employed to enhance data quality.
- Network Intrusion Detection with Generative Adversarial Networks (N-GAN) is utilized to recognize harmful behaviour in network traffic.
- Explainable AI provides transparent, interpretable explanations of the global intrusion detection model's predictions using SHAP.

This research paper sections are organized using the way outlined below: Several articles regarding current intrusion detection using federated learning were described in Section 2. Section 3 outlines the recommended process for detecting and preventing intrusion in the edge devices. The outcomes of the proposed method are presented in Section 4. Section 5 provides the conclusions of the study.

2. LITERATURE REVIEW

A distributed machine learning method called FL allows several devices to collaborate in training an intrusion detection model without exchanging raw data. This review addressed federated method for detecting and resolving intrusions in edge devices.

Chen et al. [11] developed the Federated Learning-based Attention Gated Recurrent Unit (FedAGRU) for detecting intrusions in wireless edge networks. FedAGRU significantly lowers communication overhead while preserving learning convergence by applying an attention mechanism that increases the importance of key devices and blocks irrelevant updates from being sent to the server. The test findings showed that FedAGRU outperforms competing centralized learning algorithms in terms of detection accuracy, increasing it by around 8%.

Man et al. [12] created Federated learning with Attention mechanism for Convolutional Neural Network (FedACNN), an intelligent intrusion detection system that leverages federated learning to assist the CNN deep learning model in completing

the intrusion detection task. With 50% fewer rounds of communication, the FedACNN can achieved optimal accuracy by creatively integrating the attention mechanism to mitigate the communication delay limit of federal learning.

Javeed et al. [13] developed a horizontal FL model that effectively detects intrusions by combining Bidirectional Long-Term Short Memory (BiLSTM) and CNN. Specifically, the BiLSTM component captures temporal relationships and learns sequential patterns in the data, while CNN was used to extract spatial features, it allowed the model to recognize regional trends that could indicate intrusions. The developed model CNN-BiLSTM obtained the loss of 0.021 for 5 edge devices and 0.017 loss for 20 edge devices.

Mothukuri et al. [14] developed a distributed learning method for anomaly detection to proactively identify intrusions in IoT networks. Using federated training cycles on GRU models and only communicating the learned weights with the FL's central server, this method protects the data on nearby IoT devices. Experimental findings show that the method gives the attack detection accuracy rate and secures user data privacy compared to the traditional/centralized machine learning (non-FL) versions. This established model achieves 90.255% accuracy for anomaly detection.

Tabassum et al. [15] developed Intrusion Detection System (IDS) using Federated DL and GAN (FEDGAN-IDS) that uses a GAN to determine the cyberthreats in smart homes, smart cities, and smart e-healthcare systems. Comprehensive studies utilizing a variety of datasets proved the efficacy of the FEDGAN-IDS. For every dataset examined, the recorded accuracy was higher than 97%. Additionally, the model's loss drops somewhat in the FED-IDS instance, reaching its lowest value of 0.1.

Otoum et al. [16] established a flexible architecture that combines blockchain technology with federated learning to increase network security and trustworthiness. The solution may assess the trust values of end devices from various networks and deal with people's trust as a probability, provided that security requirements are met. According to the findings, the recommended model maintains its high detection rates and accuracy, with values of roughly 0.96 and 0.93, respectively.

Liu et al. [17] created a bi-LSTM-based asynchronous federated learning arbitration system and an attention mechanism (AsyncFL-bLAM). The leader node election method for constructing the asynchronous federated learning architecture is presented by the AsyncFL-bLAM. bLAM model performance, with a cross-entropy loss of less than 0.1. For validation, epoch 40 of the test dataset was used.

2.1 Research Gap

The aforementioned reviewed paper does better at detecting threat, however it also has issues in some circumstances. For long-term prediction, approaches are limited by issues like gradient disappearance and gradient explosion [11]. Models often utilized DL-based IoT intrusion detection include a sophisticated network topology and training procedure, along with additional parameters, which makes training more challenging and energy-intensive [12]. The model successfully identifies new online threats, but when assessing its effectiveness, its communication overhead rises. The model's communication overhead rises as its performance was assessed. The scheme's false-positive ratio was higher [13]. The training process for GRU is still challenging because of the duplication between the update and reset gates. The ratio of false positives was higher in the scheme [14]. The communication overhead can be increased beyond what would be required for a typical FL configuration if portions of the GAN are additionally federated [15]. Blockchain based FL system introduces additional layers of complexity and require computational resources and has the possibility of slow training times. [16]. If the size of the sliding window increases, accuracy decreases and cross-entropy loss rises [17]. In order to solve this issue and enhance edge device security and privacy in edge environments, a FL approach for network intrusion detection using explainable AI and generative adversarial networks is proposed.

3. PROPOSED METHODOLOGY

Network entry points handle data locally instead of transmitting it back to centralized servers for analysis are referred to as edge devices. These devices, which enable quicker data processing, lower latency, and improve network performance. However, due to their frequent placement in physically susceptible places and potential lack of processing power and memory, edge devices are usually less secure than centralized data centers. Because of this, it is difficult to properly deploy conventional security procedures. To tackle this issue, a FL model is proposed for the detection of network threat using generative adversarial networks and explainable methods in edge environments to enhance security and privacy for edge devices. Figure 1 represents the proposed Architecture for Intrusion Detection on Edge Devices Using Federated Learning and Adversarial Networks.

Natural Resources for Human Health
eISSN: 2583-1194

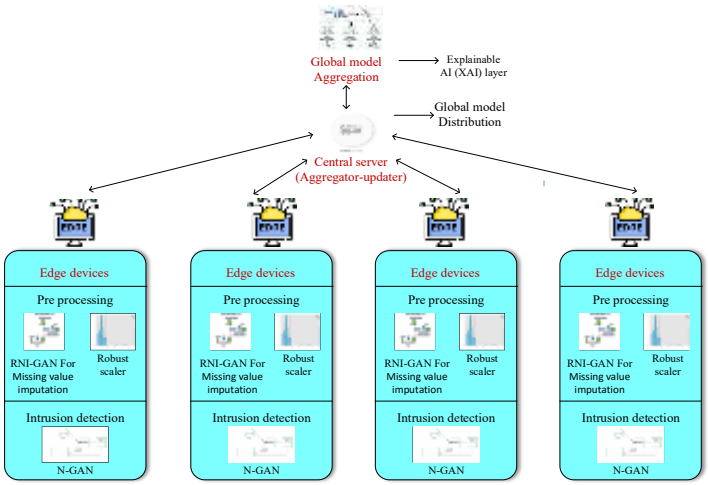

Figure 1. Proposed Architecture for Intrusion Detection on Edge Devices Using Federated Learning and Adversarial Networks.

An edge device collects network parameter based on UNSW_NB15 from Kaggle. The input dataset is pre-processed using missing value imputation and data normalization approaches to improve model accuracy. In pre-processing Recurrent Noise Injector GAN (RNI-GAN) based missing values are employed to replace the missing data and the robust scaler is used to normalize data. After pre-processing, Network Intrusion Detection with Generative Adversarial Network (N-GAN) is used to identify and flag suspicious or malicious activities in edge device. Next, each edge device transmits its model parameters to a central server, which updates the global model using these parameters. The global model's output is combined with Explainable AI (XAI) by applying the SHAP technique to enhance transparency and build trust in the system's predictions. The global model's knowledge is transmitted back to every edge device. This allows the individual edge devices to improve their own intrusion detection using the awareness of the whole network. This is the continuous process as the global model learns and advances with the distributed and ever evolving data on all edge devices.

3.1 Preprocessing

Data preparation is the transforming of unstructured, un-readable data to a form that is suitable for analysis and machine learning models. Through the missing value imputation process, RNI-GAN generates new, synthetic data instances that closely match the training data. The generator learns patterns in the data to predict missing values, the discriminator provides feedback to improve the generator's predictions by distinguishing between real values and imputed values. Adversarial training allows the RNI-GAN to generate imputation values that are high quality and contextually representative of the data space. Then the data was normalized using the Robust Scaler which is useful when dealing with outliers, as it removes the median and scales using the quantile range.

3.1.1 RNI-GAN

A deep learning model called a Recurrent Noise-Injector GAN is created to impute missing values from time series data particularly when complicated correlations and noise exist. It leverages the benefits of utilizing GAN to learn the distribution of the data and make realistic imputations with the consideration of the sequential nature of the data [18]. RNI-GAN (Recurrent Noise-Injector GAN) intends to fill-in missing values in time-series data through a GAN framework which includes adversarial noise to improve robustness. The generator produces filled-in imputations for missing values and the discriminator distinguishes the input as real or generated. The discriminator pushes the generator to be better to produce better quality generated data to then differentiate. The primary intent of adversarial learning is to include noise in the data, which will be focused most on the missing points in the data, this noise forces the model to learn more generalized noisy patterns providing robust learning to variations in the data. The combination of adversarial noise injection along with a recurrent structure which captures temporal dependency, means RNI-GAN is capable of producing realistic behaviorally-equivalent imputations while preserving the data distribution. Data imputation is carried out using the implicit generative model GAN. GANs represent a novel class of artificial neural networks that have been developed recently. This model is made up of two adversarial trained

neural networks called the discriminator and the generator. The primary purpose of the generator is to create a new data instance by taking random noise from a latent space. Meanwhile, the discriminator takes input from both the real data distribution and the data produced by the generator. It evaluates every data point and determines whether the data point is a part of the real training data or it was generated by a generator. As a result, the discriminator learns to differentiate between real and fake data, while the generator is designed to deceive the discriminator during training. A lot of cost function strategies can be applied in GANs. These schemes typically vary in $F^{(L)}$, rather than in $F^{(I)}$. A simple choice for the discriminator's cost function is given by,

$$F^{(I)}\left(\theta^{(L)}, \theta^{(I)}\right) = -\frac{1}{2} S_i \log\left(I(i)\right) - \frac{1}{2} S_y (1 - \log(I(L(y))) \quad (1)$$

Where S_i represent the predicted (or average) value determined over the distribution of i. L and I be the generator and discriminator, y and i be the inputs, $\theta^{(L)}$and $\theta^{(I)}$ are neural parameters.

3.1.2. Robust scaler

Robust Scaler is similar to the Min-max method of data normalization. The Robust scaler uses the quintile range to scale data, which is the only distinction. [19].

$$z = \frac{z_i - W_1(z)}{W_3(x) - W_1(x)} \quad (2)$$

The robust scaler is represented by equation 2, where "z" stands for the values and $W_1$and W_3 are the 25th and 75th quantiles, respectively. After pre-processing the local node was developed through federated learning. After pre-processing, a local model was trained using federated learning.

3.2 Federated Learning

Federated learning is a privacy-preserving distributed machine learning training method for networks. A central server coordinates training while maintaining a global model with dimensions. When users access sensitive private datasets, they update the global model locally by allowing,

$$C(w) = \sum_{i=1}^{N} \frac{B_i}{B} C_i(w) \quad (3)$$

Where N represents number of users, $C_i(w)$ represents the user's local goal function i, and B_i is the quantity of data points in user $i's$ private dataset D_i [20]. In order to update the global model, users engage in an iterative training process with the central server. The server updates the mobile users on the global model's current status at each iteration.

A Privacy-Enhancing Protocol is used at the node to aggregate the local modifications. This guarantees that the server learns only a significant portion of the local changes, ideally the total of all user models, without disclosing more details about any one model than the total. In the subsequent iteration, the server updates the global model using the aggregate of the local modifications

$$w^{(f+1)} = w^{(f)} - \gamma^{(f)} \sum_{i \in [N]} w_i^{(f)} \quad (4)$$

Where $\gamma^{(f)}$ shows the rate of learning and gives users access to the most recent model $w^{(f+1)}$. Before aggregation, each user must utilize random keys to disguise its local update according to conventional secure aggregation procedures. Usually, a key exchange protocol is used to create paired keys between the users. Every user pair decides on a pairwise random seed using the pairwise keys. If the user is delayed rather than dropped, Symmetric Shared Keys are not enough for privacy. The user additionally creates a private random seed in this case to maintain the confidentiality of the local update.

The Server gathers either the secret shares of a dropped user's pairwise shared random keys or the shares of a surviving user's pairwise shared random keys (but not both) in order to calculate the consolidated User Models. The private seeds of the surviving users and the pairwise shared random keys of the dropped users are then retrieved by the server from the total number of masked models. Following this, the local node executes an intrusion detection process to identify malicious activities.

Natural Resources for Human Health
eISSN: 2583-1194

3.3 Intrusion Detection

A network can be monitored for malicious activity or policy breaches by a hardware or software program called an intrusion detection system (IDS). It is common practice to use a security information and event management system to centrally notify or gather any harmful activities or violations. To find and mark questionable or malevolent activity in network traffic, N-GAN are utilized.

3.3.1 N-GAN

Deep neural networks are used to shape both the GAN and the encoder, which make up the N-GAN model [21]. The N-GAN model is trained in two steps: First, a large amount of benign network traffic data is utilized to train the GAN. Secondly, a limited number of malicious samples are employed in addition to benign data to train the encoder.

GAN training

Typical network traffic data is used for, the discriminator and generator are trained adversarial. Random noise samples taken from the latent space, the generator generates fictitious network traffic characteristics, which are then sent to the discriminator. In the discriminator, the generated features are separated from the real network traffic features. This procedure keeps on until the generator generates samples that roughly match the distribution of actual network traffic characteristics. As a consequence, the generator can successfully map latent features to the feature space of regular network traffic by the conclusion of training as it has learned the intrinsic structural variability of regular network traffic.

Encoder training

The encoder module is trained using the trained GAN module in this stage. This process is similar to an autoencoder, where the GAN generator functions as the decoder, with only the encoder parameters being updated during training while the generator weights stay fixed. Network traffic characteristics are mapped into latent features by the encoder. However, a contrastive loss function is used in place of the residual loss common to traditional reconstruction-based methods. This contrastive loss function takes into account a small number of malicious samples in addition to normal samples. This constrains the N-GAN model since the encoder is trained so that malicious samples are purposefully rebuilt poorly and benign samples are reconstructed properly.

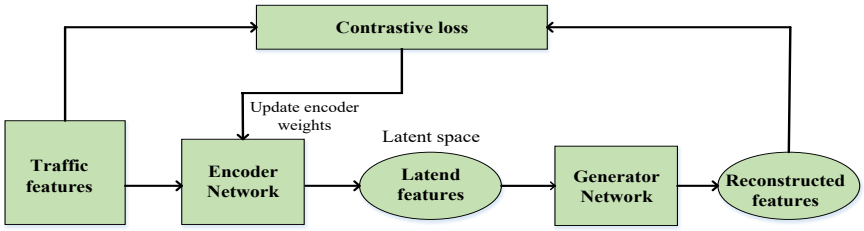

Figure 2. Architecture of encoder training setup.

Figure 2 illustrates a generative autoencoder model designed to learn robust feature representations from network traffic. The input "Traffic features" are compressed by an encoder network into a Compressed Latent Space. After that, a Generator Network rebuilds the original data using these latent characteristics. A Contrastive loss function, which calculates the difference between the original and reconstructed features, is used to train the model. The network is forced to develop a very effective representation of typical traffic as a result of this process, which makes it ideal for tasks like anomaly detection.

- *Contrastive loss function*

The data used to calculate the contrastive loss function comes from two classes: benign and malignant distributions. This approach aims to find a manifold that allows optimal reconstruction of data from the benign distribution while pushing data from the malicious distribution away from the manifold. Subsequently, the loss function is explained as follows:

$$U(N^+ \cup N^-) = \sum_{r \in N^+ \cup N^-} \max(0, v(r).m(r) - 1) \quad (5)$$

Let v(r) represent the example r's label, with $v(r) \in \{-1,1\}$; 1 and m(r) the example's separation from the manifold with $m(r) = ||r - \bar{r}||$. $N^+ \; and \; N^-$ are the two classes.

- *Anomaly score derivation*

Anomaly detection stage is no longer dependent on training once the model has been trained. The encoding (process of reconstruction) of network traffic features is performed by an encoder component, which generates latent features. Next, generator module then receives these latent characteristics and uses them as a decoder. These latent characteristics are mapped back to the network traffic feature space by the generator. Anomaly score is determined by the degree of divergence from these transformations, which are thought to approximate identity transformations. As a result these mappings generate very little variation for typical network traffic. Next, the central server receives the parameters and forwards them to the global model. The global model's output is then integrated with Explainable AI methods to reveal the logic behind the danger that was identified.

3.4 Explainable Ai (Xai)

Upon identification as an attack subsequently analyzed by Explainable AI, utilizing SHAP to elucidate the genesis of an attack by precisely highlighting the contributing features. A collection of advanced methods known as Explainable AI (XAI) aims to make the outcomes of the solution intelligible to humans in order to validate real-world applications of artificial intelligence (AI). It completely contradicts machine learning's (ML) concept of the "black box." XAI may enhance the experience of using services or goods by giving end consumers the assurance that AI will make the best choice.

3.4.1 SHapley Additive exPlanations (SHAP)

The concept of SHAP originated in game theory, where it is used to determine the contribution of each player to a collective game. Depending on the participants' contributions and results, the total benefit was to be divided among them. Players are not given equal rewards depending on local accuracy, consistency, and the null effect when using Shapley values; instead, meaningful incentives are given to each participant in an objective way. This approach has returned in ML difficulties because of the black box nature of ML models, which may be utilized to interpret model outcomes, enhancing transparency and building trust in their usage [22]. Three characteristics of the Shapley value estimate process are distinctive: consistency, missingness, and local accuracy. Local correctness is the amount of relevant information included in the explanation; consistency is the degree of connection between explanations and predictions; and missingness is the evaluation of explanation changes between runs. The average of each feature permutation's marginal contributions is known as the Shapley value. This is the mathematical expression.

$$\emptyset_k = \sum_{X \subseteq Y \setminus \{k\}} \frac{|X|!(r-|X|-1)!}{r!} [w(X \cup \{k\}) - w(X)] \quad (6)$$

Where $\emptyset_k$ is feature $k's$ contribution, Y is a collection that includes every feature, r represents the number of features in Y, X is the subset of Y that contains feature k, and $w(Y)$ denotes the base value, which, in the absence of feature values, indicates the expected result for each feature in Y. The shapley values are predicated on the notion that the significance of a single variable should be assessed by taking into account the results of every potential player combination. If the data is classified as a non-attack, it undergoes encryption using a Lightweight Elliptic Curve Integrated Homomorphic Encryption scheme.

4. RESULT AND DISCUSSION

This method presented a FL model for the detection of network intrusions usingGAN. RNI-GAN and Min-Max are used to pre-process the data for normalization and missing value imputation. N-GAN is used to identify Anomalous activities in Network. The weight parameters are then sent to the central server by each edge device. The global model receives the parameters from the central server. The global model incorporates SHAP for transparency and reliability. Information from the global system is returned to each local unit, boosting their individual detection. This continuous feedback loop helps the global system learn from widely distributed, changing data. The proposed platform is developed using Python 3.8 and powered by an Intel Core i7 CPU, NVIDIA GeForce RTX 3070 GPU, and 16GB of RAM.

Table 1. Simulation Parameter for RNI-GAN, N-GAN.

Parameter	Methods	Values
Generator learning Rate		1e-4
Discriminator learning rate		1e-4
Optimizer		Adam ($\beta_i = 0.5, \beta_2 = 0.999$)

Batch size	RNI-GAN	64
epochs		200
Loss function		Binary cross entropy
Activation functions	N-GAN	tanh
Optimizer		Adam
Learning rate		0.0001
Batch size		64
Epochs		100

The simulation parameter for RNI-GAN, N-GAN are listed in Table 1.

4.1 Dataset Description

The IXIA PerfectStorm program, created by the Australian Centre for Cyber Security's (ACCS) Cyber Range Lab, generated the raw network packets for the UNSW-NB15 dataset by combining simulated contemporary attack patterns with common real-world network traffic. Using the Tcpdump tool, it is possible to capture up to 100 GB of raw data, including Pcap files. Worms, shell code, reconnaissance, backdoors, analysis, fuzzers, exploits, and denial-of-service assaults are among the nine attack types found in this dataset [23]. Twelve algorithms are created to provide 49 attributes with the class label using the Argus and Bro-IDS tools. The UNSW-NB15_features.csv file contains descriptions of these features. The four CSV files—UNSW-NB15_1.csv, UNSW-NB15_2.csv, UNSW-NB15_3.csv, and UNSW-NB15_4.csv—contain two million and 540,044 records. The event list file and ground truth table are named UNSW-NB15_LIST_EVENTS.csv and UNSW-NB15_GT.csv, respectively. The data is divided into two, UNSW_NB15_training-set.csv to train and UNSW_NB15_testing-set.csv to test. It has a total of 82332 samples, 16467 of which (20 %) are training samples and 65865 of which (80 %) are testing samples. Both training and testing sets contain a mix of normal and attack records.

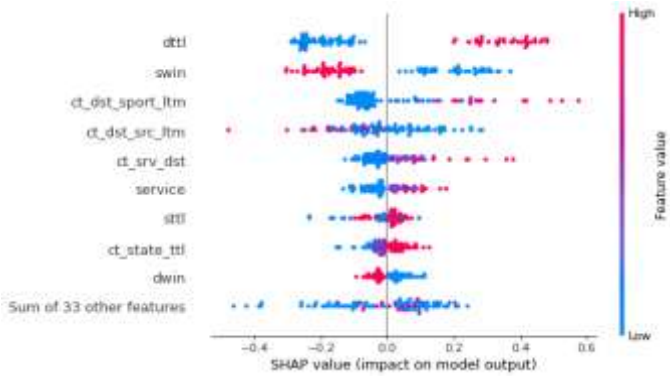

Figure 3. SHAP plot of feature importance and impact on model output.

The figure 3 demonstrates the SHAP summary plot, which ranks features in order of their contribution to the output of the model. A dot can indicate an instance with the position on the x-axis indicating the SHAP value, i.e. direction and strength of impact on prediction with positive values increasing predictions and negative values decreasing them. The most significant features are listed on the y axis with dttl, swin, and ct_dst_sport_ltm being the most influential and minor features are included as the total of 33 other features. The scale of blue to red depicts the real values of features with blue being low values and red being high values, the interpretation is that the impact of different ranges of the feature values on the prediction outcome are understandable. The plot gives feature importance at a global level as well as understanding of the impact of feature values on the decision making process of the model.

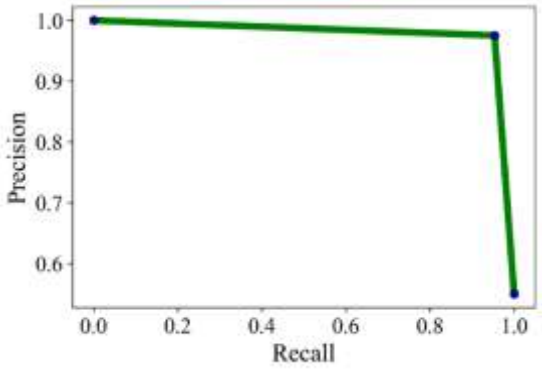

Figure 4. Precision – Recall Curve of the Proposed Method.

Figure 4 displays the Precision-Recall curve, which is used to assess the suggested method's categorization performance. The graph indicates that the approach performs well for the majority of the recall values, starting with high precision but poor recall. The model's overall performance stability, which balances accuracy and recall, is indicated by the thick green line.

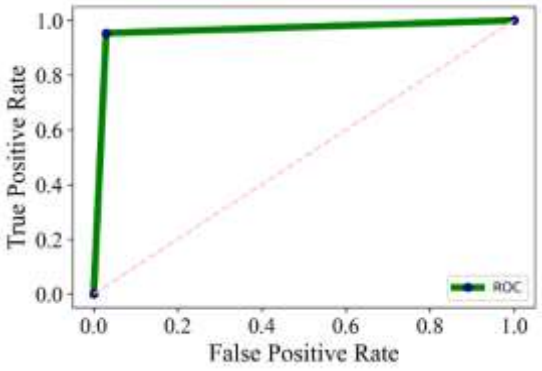

Figure 5. Proposed methods for AUC and ROC plot.

The suggested model's ROC curve and AUC score are displayed in Figure 5. The curve will increase sharply towards the upper-left hand corner, meaning that there is a high rate of true positives and low rate of false positives. This means high sensitivity and low prediction error. In contrast to the diagonally pink dashed line that indicates the model's random guessing (baseline), the thick green curve shows the model's maximum capacity for discrimination.

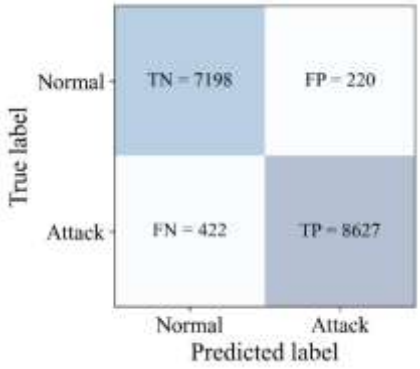

Figure 6. The proposed method's confusion matrix.

Figure 6 presents the confusion matrix for the proposed method. The confusion matrix is used to evaluate the accuracy of the classification approach. To evaluate the effectiveness of the suggested method, two distinct classes based on vulnerability detection are taken into consideration. Class 1 (attack) is predicted to be 8627 and class 0 non-attack (normal) to be 7198.

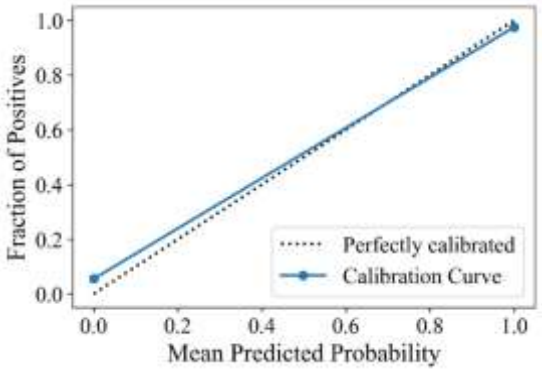

Figure 7. Calibration curve for the suggested method.

The calibration curve is shown in Figure 7 with the actual proportion of positive instances on the y-axis and the mean estimated probability on the x-axis. The graph shows a calibration curve that contrasts the actual results with the probability predicted by the model. A properly calibrated model, where the actual and projected probabilities are identical, is indicated by the dotted diagonal line. The blue solid line represents model's calibration it closely aligns with diagonal for indicating an accurate probability predictions. The curve maintains the close match throughout the range. Both the predicted probability and an actual fraction of positives reach a value of 1.0 conforming strong calibration accuracy.

4.2 Comparison Analysis

The effectiveness of a proposed N-GAN is evaluated by some of existing techniques including learning-driven detection mitigation (LEDEM) [24], Multi-Feature Extraction Extreme Learning Machine (MFE-ELM) [25], feature selection-whale optimization algorithm deep neural network (FS-WOA–DNN) [26]. Accuracy, PPV, TPR, Error, TNR, FNR, FPR, F1-score, FDR, NPV, FOR, MK, TS, MCC and Kappa are some of the performance measures used in this comparison. A graph of these factors shown below.

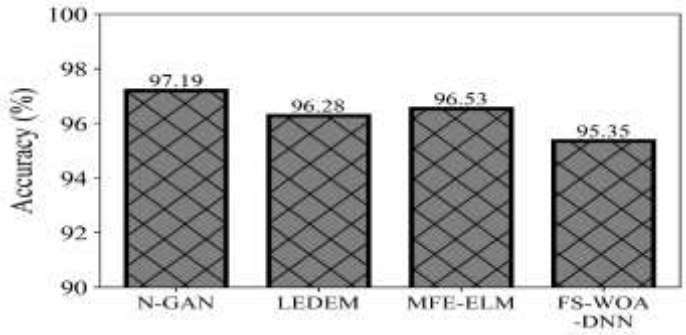

Figure 8. Comparison of N-GAN Accuracy among different Algorithms.

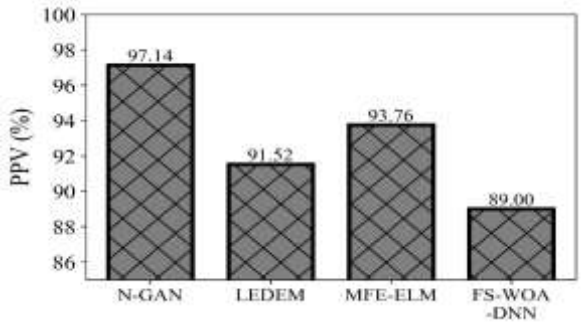

Figure 9. Positive Predictive Value compared to the current algorithm and N-GAN algorithm.

Figures 8 and 9 compares the Accuracy value and Positive Predictive Value. The suggested N-GAN performs better than current methods, with a 97.19% accuracy rate and a 97.14% positive predictive value. The Accuracy and Positive Predictive Value of techniques like LEDEM, MFE-ELM, FS-WOA-DNN are 96.28%, 96.53%, 95.35% and 91.52%, 93.76%, 89.00% respectively. According to these results, the N-GAN is better than other existing techniques. It Enhanced detection accuracy by generating synthetic attack samples to balance datasets.

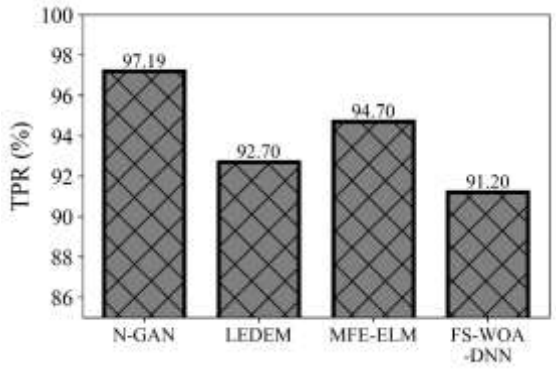

Figure 10. N-GAN's TPR in comparison to other algorithms.

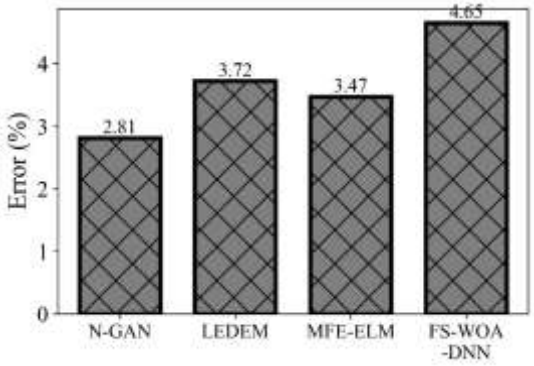

Figure 11. The error value of N-GAN compared to other methods.

Figures 10 and 11 compares the True Positive Rate and Error. The proposed N-GAN method outperforms other techniques by attaining a true positive rate of 97.19% and an error rate of 2.81%. The TPR and error of techniques like LEDEM, MFE-ELM, and FS-WOA-DNN are 92.70%, 94.70%, 91.20% and 3.72%, 3.47%, 4.65% respectively. According to these results, the N-GAN is better at discriminating than other existing techniques. It improved anomaly detection capabilities through advanced pattern recognition.

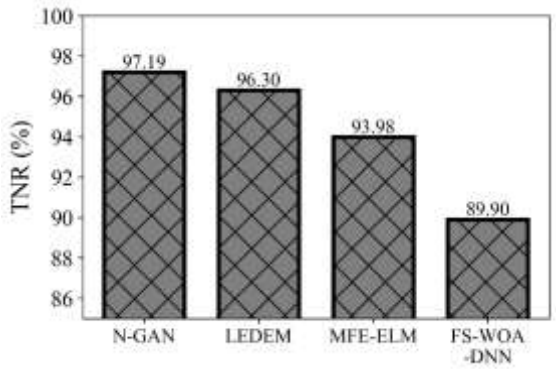

Figure 12. TNR compared to the current algorithm and N-GAN algorithm.

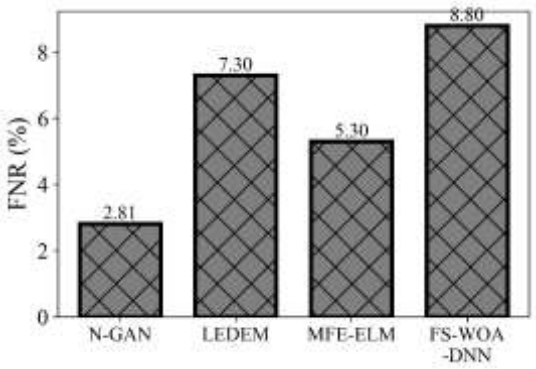

Figure 13. FNR value of N-GAN in comparison to other current algorithms.

Comparison of the True Negative Rate and False Negative Rate values is shown in figures 12 and 13. With a TNR of 97.19% and a FNR of 2.81%, the proposed N-GAN outperforms existing methods. LEDEM, MFE-ELM, FS-WOA-DNN have respective TNR and FNR of 96.30%, 93.98%, 89.90% and 7.30%, 5.30%, 8.80%. These outcomes demonstrate the N-GAN outperforms the others in terms of distinction. The suggested method can adjust and adapt to emerging threats by modeling new attack vectors.

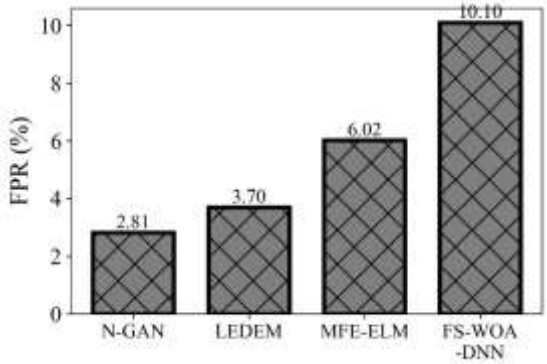

Figure 14. N-GAN's FPR compares to other existing techniques.

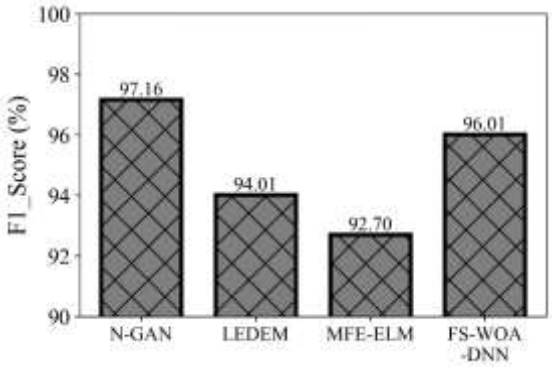

Figure 15. Comparison of F1-score for N-GAN's with various existing algorithms.

Figures 14 and 15 compares the False Positive Rate value and F1-score. The suggested N-GAN performs better than the other methods, with an FPR of 2.81% and an F1-score of 97.16%, respectively. The corresponding FPR and F1-Scores of methods like LEDEM, MFE-ELM, and FS-WOA-DNN are 3.70%, 6.02%, and 10.10% and 94.01%, 92.70%, and 96.01%, respectively. According to these results, the N-GAN is better discriminating than other existing techniques. The approach has reduction of false positives by refining classification boundaries.

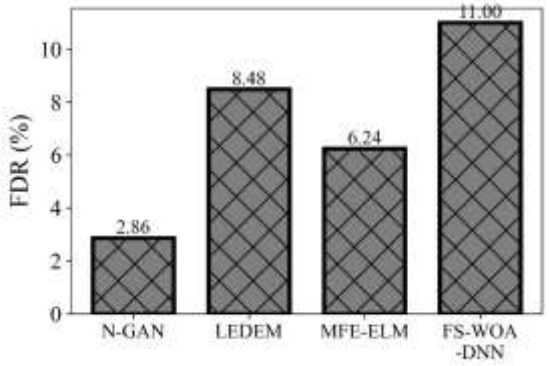

Figure 16. FDR for N-GAN compared to other current algorithms.

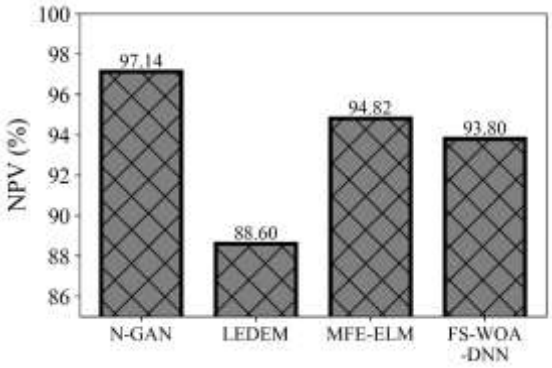

Figure 17. Value of NPV for N-GAN compared to various existing methods.

Figures 16, 17 compare the False Discovery Rate (FDR) and Negative Predictive Value (NPV). The proposed N-GAN has better performance than the other models with a FDR of 2.86% and a NPV of 97.14% respectively. The LEDEM, MFE-ELM, FS-WOA-DNN have achieved FDR of 8.48%, 6.24%, 11.00%. Moreover NPV for LEDEM, MFE-ELM, and FS-WOA-DNN are 88.60%, 94.82%, 93.80% respectively. According to findings, the N-GAN framework has the ability to discriminate better than the alternatives. It increased robustness against adversarial attacks through training with diverse data.

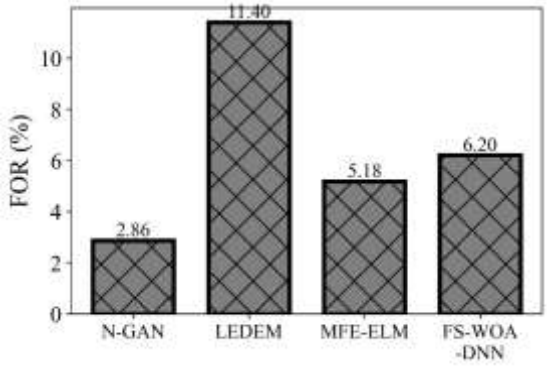

Figure 18. Comparison of FOR value for N-GAN with various existing algorithms.

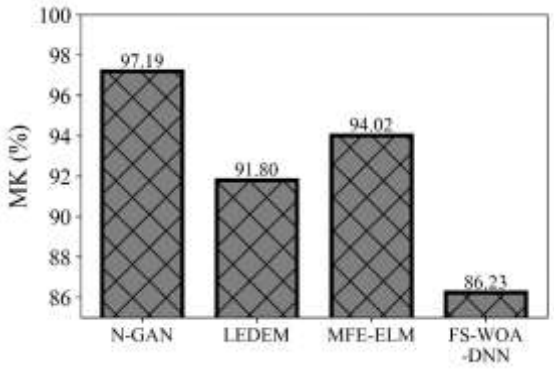

Figure 19. MK for N-GAN compared to other current algorithms.

Comparison of the False Omission Rate and Markedness values is shown in Figures 18 and 19. With a false omission rate (FOR) of 2.86% and a miss rate (MK) of 97.19%, the recommended N-GAN outperforms previous methods. The corresponding FOR and MK values for LEDEM, MFE-ELM, and FS-WOA-DNN are 11.40% and 91.80%, 5.18% and 94.02%, and 6.20% and 86.23%, respectively. These results clearly indicate that the N-GAN algorithm outperforms others in terms of performance. Furthermore, N-GAN can learn complex distributions of the data indicating better ability for generalization.

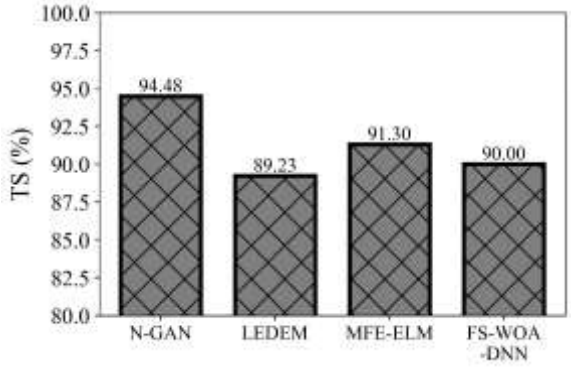

Figure 20. Comparison of TS for N-GAN's with various existing algorithms.

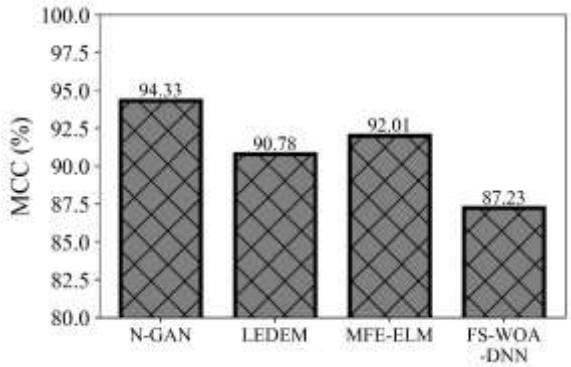

Figure 21. MCC value of N-GAN in comparison to other current algorithms.

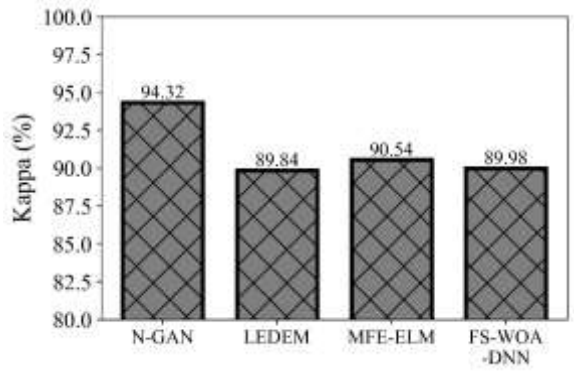

Figure 22. The Kappa value of N-GAN compared to other methods.

Figures 20, 21, and 22 compare the Threat Score and Matthews Correlation Coefficient value and Kappa. The proposed N-GAN has better performance than the other models with a TS value of 94.48% and a MCC of 94.33% and Kappa value of 94.32% respectively. The LEDEM, MFE-ELM, FS-WOA-DNN have achieved TS of 89.23%, 91.30%, 90.00%, respectively. Moreover, the MCC values for LEDEM, MFE-ELM, and FS-WOA-DNN are 90.78%, 92.01%, 87.23%, respectively. The corresponding results for Kappa are 89.84%, 90.54%, 89.98%. According to the findings, N-GAN algorithm has the ability to discriminate better than the alternatives. It helps to facilitation of detection and response to network anomalies.

5. CONCLUSION

Edge devices can be defined as essential connectors for data exchange between local networks and cloud resources in military sector. However, distributed devices possess unique exposure vulnerabilities and sparse resources, making intrusion detection difficult. The intrusion detection process is made more complicated because often assume a centralized security approach in less well-defined edge environments. In order to address these challenges, a Federated Learning Model for Network Intrusion Detection which integrates Generative Adversarial Networks and Explainable AI, specifically engineered for deployment within edge settings. Data obtained from Kaggle website and pre-processing the input dataset using procedures for managing missing values and normalizing data, the model's accuracy could be increased. In particular, the Min-Max method for normalization and Recurrent Noise Injector GAN (RNI-GAN) to fill any missing values. The Network Intrusion Detection based Generative Adversarial Network (N-GAN) was utilized to identify malicious activity within network communications. Following a local data analysis, edge devices transmit model parameters to a central server. This server collects the data from all edge devices, updates it, and then shares it with the global model. To enhance transparency and confidence in the system's predictions, the global model's output incorporates the SHAP explainability technique. Once the global model's outputs have been transmitted to each edge device, each device uses the network's aggregate knowledge to augment its own intrusion detection capabilities. This happens constantly enabling the global model to refine its understanding of the evolving diverse data produced by all edge devices over time. Compared to other existing algorithms, the suggested model achieved performance metrics of 97.19% TPR, 2.86% FDR, 97.14% PPV, and 97.16% F1_score. Therefore, the proposed approach performs effectively in detecting network intrusions.

5.1 Limitation and future scope

FL for IDS is susceptible to high latency and false alarms caused by broadcasting global models and transmitting learned models. Implementing federated learning-based IDS faces significant challenges, including high processing costs, model complexity, and the requirement for seamless integration with existing security frameworks. To overcome these issues, additional research and evaluation of these models are essential. Future research focus should be on better domain generalization and raising the overall accuracy of intrusion detection in any network setting. Furthermore, it is crucial to concentrate on creating more resilient federated learning algorithms that can effectively handle non-independent and identically distributed (Non-IID) data, building training frameworks that support communication, and enhance adversarial resilience where necessary. Leveraging developing technologies, federated learning can be even better operationalized with emerging technologies such as blockchain for auditability, homomorphic encryption for secured computations, and reinforcement learning for dynamic resource management.

ACKNOWLEDGEMENT

Funding. The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Conflict of Interest. The authors declared that they have no conflicts of interest to this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Availability of data and material. Not applicable

Code availability. Not applicable

Author contributions. The corresponding author claims the major contribution of the paper including formulation, analysis and editing. The co-authors provides guidance to verify the analysis result and manuscript editing.

Compliance with ethical standards. This article is a completely original work of its authors; it has not been published before and will not be sent to other publications until the journal's editorial board decides not to accept it for publication.

Clinical Trial Number: Not applicable.

REFERENCE

[1] Kholidy, H. A., 2021. *Detecting impersonation attacks in cloud computing environments using a centric user profiling approach.* Future Generation Computer Systems 117, 299–320.

[2] Singh, P., Kaur, A., Aujla, G. S., Batth, R. S., Kanhere, S., 2020. *Daas: Dew computing as a service for intelligent intrusion detection in edge-of-things ecosystem.* IEEE Internet of Things Journal 8(16), 12569–12577.

[3] Samunnisa, K., Kumar, G. S. V., Madhavi, K., 2023. *Intrusion detection system in distributed cloud computing: Hybrid clustering and classification methods.* Measurement: Sensors 25, 100612.

[4] Rajagopal, S., Kundapur, P. P., 2021. *Towards effective network intrusion detection: From concept to creation on Azure cloud.* IEEE Access 9, 19723–19742.

[5] Aldribi, A., Traoré, I., Moa, B., Nwamuo, O., 2020. *Hypervisor-based cloud intrusion detection through online multivariate statistical change tracking.* Computers & Security 88, 101646.

[6] Thabit, F., Alhomdy, S., Jagtap, S., 2021. *A new data security algorithm for the cloud computing based on genetics techniques and logical-mathematical functions.* International Journal of Intelligent Networks 2, 18–33.

[7] Karthick, S., Muthukumaran, N., 2024. *Deep RegNet-150 architecture for single image super resolution of real-time unpaired image data.* Applied Soft Computing 162, 111837.

[8] Rabbani, M., Wang, Y. L., Khoshkangini, R., Jelodar, H., Zhao, R., Hu, P., 2020. *A hybrid machine learning approach for malicious behaviour detection and recognition in cloud computing.* Journal of Network and Computer Applications 151, 102507.

[9] Durai, K. N., Subha, R., Haldorai, A., 2021. *A novel method to detect and prevent SQLIA using ontology to cloud web security.* Wireless Personal Communications 117(4), 2995–3014.

[10] Karuppusamy, L., Ravi, J., Dabbu, M., Lakshmanan, S., 2022. *Chronological salp swarm algorithm-based deep belief network for intrusion detection in cloud using fuzzy entropy.* International Journal of Numerical Modelling: Electronic Networks, Devices and Fields 35(1), e2948.

[11] Chen, Z., Lv, N., Liu, P., Fang, Y., Chen, K., Pan, W., 2020. *Intrusion detection for wireless edge networks based on federated learning.* IEEE Access 8, 217463–217472.

[12] Man, D., Zeng, F., Yang, W., Yu, M., Lv, J., Wang, Y., 2021. *Intelligent intrusion detection based on federated learning for edge-assisted Internet of Things.* Security and Communication Networks 2021(1), 9361348.

[13] Javeed, D., Saeed, M. S., Adil, M., Kumar, P., Jolfaei, A., 2024. *A federated learning-based zero trust intrusion detection system for Internet of Things.* Ad Hoc Networks 162, 103540.

[14] Mothukuri, V., Khare, P., Parizi, R. M., Pouriyeh, S., Dehghantanha, A., Srivastava, G., 2021. *Federated-learning-based anomaly detection for IoT security attacks.* IEEE Internet of Things Journal 9(4), 2545–2554.

[15] Tabassum, A., Erbad, A., Lebda, W., Mohamed, A., Guizani, M., 2022. *FedGAN-IDS: Privacy-preserving IDS using GAN and federated learning.* Computer Communications 192, 299–310.

[16] Otoum, S., Al Ridhawi, I., Mouftah, H., 2021. *Securing critical IoT infrastructures with blockchain-supported federated learning.* IEEE Internet of Things Journal 9(4), 2592–2601.

[17] Liu, Z., Guo, C., Liu, D., Yin, X., 2023. *An asynchronous federated learning arbitration model for low-rate DDoS attack detection.* IEEE Access 11, 18448–18460.

[18] Kazemi, A., Meidani, H., 2021. *IGANI: Iterative generative adversarial networks for imputation with application to traffic data.* IEEE Access 9, 112966–112977.

[19] Siddiqi, M. A., Pak, W., 2021. *An agile approach to identify single and hybrid normalization for enhancing machine learning-based network intrusion detection.* IEEE Access 9, 137494–137513.

[20] So, J., Güler, B., Avestimehr, A. S., 2020. *Byzantine-resilient secure federated learning.* IEEE Journal on Selected Areas in Communications 39(7), 2168–2181.

[21] Iliyasu, A. S., Deng, H., 2022. *N-GAN: A novel anomaly-based network intrusion detection with generative adversarial networks.* International Journal of Information Technology 14(7), 3365–3375.

[22] Pradhan, B., Dikshit, A., Lee, S., Kim, H., 2023. *An explainable AI (XAI) model for landslide susceptibility modeling.* Applied Soft Computing 142, 110324.

[23] Mr Wells David, 2019. *UNSW-NB15.* Kaggle. Accessed 24-07-2025.

[24] Ravi, N., Shalinie, S. M., 2020. *Learning-driven detection and mitigation of DDoS attack in IoT via SDN-cloud architecture.* IEEE Internet of Things Journal 7(4), 3559–3570.

[25] Lin, H., Xue, Q., Feng, J., Bai, D., 2023. *Internet of things intrusion detection model and algorithm based on cloud computing and multi-feature extraction extreme learning machine.* Digital Communications and Networks 9(1), 111–124.

[26] Agarwal, A., Khari, M., Singh, R., 2022. *Detection of DDoS attack using deep learning model in cloud storage application.* Wireless Personal Communications 127(1), 419–439.

[27]