SmartShield: A CGAN-Boosted Model for Detecting IoT Cyber Threats

AMJAD JUMAAH FRHAN
Department of Education and Islamic Studies, Sunni Endowment Diwan
University of AL Mashreq, Department of Cybersecurity Engineering Technology
Baghdad, IRAQ

Abstract: As the Internet of Things (IoT) has evolved very quickly, serious security concerns have cropped up, with ensuring linked devices emerging vital. This study presents a data-driven system that combines synthetic data generation and machine learning (ML) to accurately detect cyberattacks on IoT devices. It uses CGAN technology to generate synthetic attack data and then uses LightGBM to identify attack patterns. The approach includes cleaning of IoT data and a LightGBM feature choice strategy. It learns to distinguish between different types of attacks, such as DoS attacks, ARP poisoning attacks, and data theft attacks. Additionally, it makes use of a gradient boosting architecture, which strikes a useful balance between computing cost and detection accuracy. Moreover, the suggested model outperformed previous intrusion detection models with an accuracy of 87% in detecting attacks on the RT-IoT2022 dataset supplemented with GAN-generated data. These outcomes say that the tree-guided gradient strengthening procedure may greatly decrease the expense of processing and has many possibilities for deployment in IoT contexts. Studies suggest the intrusion detection efficiency of assaults provide use with a useful tool for boosting businesses' safety ratings in an increasingly linked and prone electronic zoon.

Keywords: Intrusion detection systems (IDS), Cyber-Security, Machine learning, IoT, LightGBM Model.

Received: June 25, 2025. Revised: August 2, 2025. Accepted: August 25, 2025. Published: October 22, 2025.

1. Introduction

Networked computer systems have grown increasingly essential in our society as a result of the Internet's explosive expansion. The Internet has a negative aspect in addition to its many positive aspects. In particular, people and groups who target and abuse computer systems create new hazards every day. The number of computer attacks has skyrocketed in recent years, according to the Computer Emergency Response Team/Coordination Center (CERT/CC) [1].

Cybersecurity issues have become a top concern for businesses. Today's network topologies rely heavily on innovation, connectivity, and emerging technologies, and business operations are frequently impacted by such breaches, typically resulting in significant financial losses [2]. On the other side of the scale, advances in machine learning and artificial intelligence have assisted researchers handle a variety of issues. This paper's goal is to use some of these technologies to address cybersecurity issues. We examine and evaluate transactional data, create models, and generate predictions from them so as to identify threats [3].

Connected technologies like 5G communication, smart grids, and the Internet of Things (IoT) have advanced quickly in recent years [4]. These developments have completely changed how people and industries engage with the digital environment. While there are many advantages to this expansion, the complexity of communication networks and the number of linked devices have increased exponentially, increasing the range of security threats [5] .

There are challenges associated with implementing ML-based IDS. The effectiveness of these systems relies on the quality and diversity of the training data, the choice of relevant features, and the optimization of the models. Moreover, the need for real-time detection requires a high level of computational efficiency, which can be a considerable [6]

Machine Learning (ML) offers an attractive solution to these challenges by providing advanced analytical methods capable of learning from data and identifying patterns indicative of intrusions. Unlike traditional approaches, ML algorithms can adapt to new and evolving threats, offering a proactive method for intrusion detection. Employing ML for IDS necessitates the training of models to identify both established and new attack patterns using historical data, thereby enhancing the system's ability to detect a range of intrusions with high accuracy [7].

The goal of security threat detection is to identify suspicious or malicious activity that may indicate an ongoing cyberattack or hacking attempt, understand the nature of attacks and the vulnerabilities they exploit, and improve your security posture by implementing better preventative measures.

The paper is structured as follows: Intrusion detection systems are discussed in Section 2, and the data set and pre-processing techniques are analyzed in Section 4. The IDS architecture is described in Section 5, and performance findings are presented in Section 6. As laid out in more detail in Section 4.3, the ensemble strategy averages normalized probabilistic outputs from CNN-BiLSTM, Random Forest, and SVM classifiers adopting a soft voting procedure.

Below is a summary of this study's primary highlights:

- Through a soft-voting ensemble scheme, we demonstrate a hybrid intrusion detection framework that combines CNN–BiLSTM deep features along with conventional machine learning classifiers.
- We put into practice thorough investigation preprocessing pipeline who is tailored for the CIC-

IDS2017 dataset, covering normalization, feature selection, and PCA.

- For improved resilience against zero-day and polymorphic infections, we merge the ideas of learning-based and signature-based detection techniques.

2. Related Work

Investigation in the field of research in biology has proved wide-ranging, applying a variety of techniques, and conclusions have varied depending upon the conditions. Here will be some of the most significant aspects of this study in the last few years.

Researcher introduces S2CGAN-IDS, a lightweight framework. In order to enhance the number of minority classes in both the data space and the feature space, the suggested framework makes use of the properties of network traffic distribution. This greatly boosts the minority class detection rate while maintaining the accuracy of majority class detection. Utilizing the CICIDS2017 dataset, the suggested approach was shown to be beneficial in reducing the influence of scarcity on experiments. The results of the experiments show that the suggested strategy performs better than the current one in terms of precision and recall, in particular, with an enhancement of 10.2% in F1 score. This approach's drawbacks include how resource-intensive it is and how unreliable intrusion detection systems are [8].

GAN (generative adversarial network) introduced for establishing entirely distributed intrusion detection system (IDS) for Internet of Things (IoT) devices. This approach can identify erroneous activity sans relying on a single control system, making it effective against threats from the inside out. Simulation results show that the proposed DID achieves higher intrusion detection accuracy compared to a standalone IDS based on a single dataset of IoT devices. More specifically, the study results show that the GAN-based DID achieves up to a 20% increase in accuracy, a 25% increase in precision, and a 60% decrease in false positive rate compared to a standalone GAN-based IDS [9].

In this research, the researcher presented a two-stage intrusion detection framework designed to secure the Internet of Things (IoT) environment and according to two detectors. Using generative adversarial neural networks (GANs), the researchers initially proposed an adversarial training strategy. This approach aims to help the first detector train on robust features by providing it with competitive examples as "validation sets." The effectiveness of the proposed approach was evaluated in terms of detection accuracy and robustness against adversarial attacks. Experimental results conducted on a novel cybersecurity dataset showed that the proposed methodology was effective in detecting both intrusions and persistent adversarial attacks. The weighted precision, recall, F1-score, and accuracy achieved values of 96%, 95%, 95%, and 95%, respectively [10].

In this research, the researcher proposed a conditional generative tabular network (CTGAN)-based intrusion detection system (IDS) to recognize denial of service (DoS) as well as global denial of service (DDoS) attacks on Internet of Things (IoT) networks. This system uses a generative network to collect synthetic network traffic data. The generated synthetic tabular data is then utilized to train various shallow and deep learning classifiers, which, in turn, enhances the detection performance of the model. The Bot-IoT dataset was used to assess the effectiveness of the suggested methodology, which measured F1 score, recall, precision, and detection accuracy. Results from experiments showed that this suggested methodology was highly accurate in identifying DDoS and DoS assaults on IoT networks [11].

Another novel Hybrid Detection Approach for Intrusion Detection Systems (HDA-IDS), designed to effectively identify both known and unknown Denial-of-Service (DoS) attacks and botnets by integrating signature-based and anomaly-based detection methodologies. A key contribution of this research is the development of a new anomaly detection model termed CL-GAN. This model leverages a combination of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks with Generative Adversarial Networks (GANs) to establish a baseline representation of normal network behavior, thereby facilitating the detection of malicious traffic. The CL-GAN model notably enhances detection accuracy while simultaneously reducing the time required for both training and testing in the context of attack and botnet identification. Experimental results demonstrate that the proposed HDA-IDS model outperforms other existing intrusion detection systems, exhibiting an average overall improvement of 5% in precision, recall, and F1 score across diverse datasets including NSL-KDD, CICIDS2018, and Bot-IoT [12].

LightShield an IDS that attempts to improve detection of intrusion in connected cars by leveraging high-performance computing. LightShield includes IoT data preprocessing, a ReliefF algorithm for choice of features, and an innovative identifying model based on LightGBM, a gradient-boosting scheme. The system uses graphics processing unit (GPU) acceleration for faster model validation. The binary classification model demonstrated exceptional accuracy, achieving 99.82% accuracy in identifying potential attacks. Meanwhile, the multi-class classification model achieved a respectable 97.25% accuracy in classifying various types of attacks [13].

Proposed a hybrid intrusion detection system using the LightGBM algorithm. It is used for traffic-level anomaly filtering, while MobileNetV2 uses packet-level detection. The proposed Hybrid NNIDS model outperformed other intrusion detection models on the ACI-IoT-2023 dataset with an accuracy of 94%, an F1 score of 91%, and a precision rate of 93%. The results indicate that the developed asymmetric algorithm can significantly reduce processing cost and has potential for implementation in IoT environments [14].

Synthesis of a denoising auto encoders with a LightGBM learner. Examples are classified using the LightGBM classifier, and the proposed model outperforms other current methods of intrusion detection on nine standard datasets for binary as well as multiclassification tasks. The maximum detection rate of the model has been higher than 99.60% for CIDDS-001, 99.90%

for CIDDS-002, 97.00% for ISCX-Tor2016, 96.11% for UNSW-NB15, 99.86% for CIC-IDS17, 97.76% for ISCX-URL16, 99.91% for BoT-IoT, and 97.43% for the IoTID2020 and Kyoto 2006+ datasets, respectively. The training lasts for 1.10 to 21.78 seconds. The model's efficacy confirms that it can be applied in real-time to any industrial network traffic, including fog cloud computing and intelligent Internet of Others settings [15].

Intrusion detection system

Intrusion detection appliances, which are a vital element of privacy and security, utilize many ways to examine a wireless network for criminal activity or assaults and observe the analysis of electronic data, thus. Signage-based systems and anomaly-based technologies are the two primary groups in which they reside [16].

In order to detect malicious activity, Similar security lapses, such as attacks that take place within or outside the infrastructure, intrusion detection systems are popular cybersecurity devices that collect, process, and analyze data either from connections or computing hosts [17].

To create the starting point framework anomaly-based computers replicate the system's typical actions and compare it to the behavior being observed [18].

Standard cybersecurity tools known as intrusion detection systems are designed to gather, process, and analyze data from computer hosts or networks in order to identify security reaches attacks that take place inside or outside of the infrastructure and other destructive behavior [8].

Data Description

Our study is based on the publicly available named RT-IoT2022 dataset [1]. It was created from real-time internet of things (IoT) infrastructure, i.e., from monitoring real network traffic in an IoT environment. It provides an extensive list comprising a variety of internet of things and several networked assault measures. The objective is to offer an extensive and realistic dataset for the purpose of intrusion detection system (IDS) training and assessment in Internet of Things scenarios. This dataset includes information from IoT devices including Thing Speak-LED, Wipro-Bulb, and MQTT-Temp, as well as simulated attack scenarios such SSH brute-force attempts and DDoS attacks employing Hping, Slowloris, and Nmap patterns. This allows the dataset to include both neutral and hostile network behaviors. RT-IoT2022 provides a detailed perspective on the complex nature of network traffic. The Zeek network monitoring tool and flow sensor extension are used to accurately record the bidirectional characteristics of network traffic. To improve the performance of intrusion detection systems (IDSs) and support the development of resilient and robust security solutions for real-time IoT networks, researchers can use the RT-IoT2022 dataset.

This dataset was later donated to the UCI Machine Learning Repository, making it available to cybersecurity and IoT researchers. A small CSV file containing a total of 123117 records and 49 features (see Table 1) was used. Class labels were provided so that binary and multi-class classification of network activity could be performed using supervised learning. A thorough description of handling inequalities is provided. To even out minority attack classes, we specifically used SMO TE oversampling on the training material.

Additionally, we used adjusted cross-entropy loss after CNN–BiLSTM to confirm performances robustness.

Table 1 shows the class labels so that binary and multi-class classification of network activity can be performed using supervised learning.

A. Integer Features

Variable Name	Role	Description
id.orig_p	Feature	Originating port number of the source host
id.resp_p	Feature	Responding port number of the destination host
fwd_pkts_tot	Feature	Total number of forward packets
bwd_pkts_tot	Feature	Total number of backward packets
fwd_data_pkts_tot	Feature	Forward packets that contain actual data
bwd_data_pk	Feature	Backward packets for actual data

B. Continuous Features

Variable Name	*Role*	*Description*	*Units*
service	*Feature*	*Value representing service behavior or type*	*N/A or Encoded*
flow_duration	*Feature*	*Total duration of the network flow*	*Microseconds*
fwd_pkts_per_sec	*Feature*	*Rate of forward packets per second*	*Packets/sec*

C. Categorical Features

Variable Name	Role	Type	Description	Units
proto	Feature	Categorical	Protocol used (e.g., TCP, UDP, ICMP)	Protocol Name

TABLE 1. IOT INTRUSION DETECTION DATASET FEATURES (GROUPED BY TYPE)

Preprocessing

The complete RT-IoT2022 dataset was uploaded for preprocessing. the raw RT-IoT2022 dataset is scrutinized to eliminate errors like missing, incomplete, and inconsistent values Additionally, duplicates were removed from both rows and columns.

The tag encoder replaces categorical features in a dataset containing the attack category, all TCP and IP tags, and protocol type with numeric values for future processing. Specific features such as the source IP address, destination IP address, and timestamp are also removed from the RT-IoT2022 dataset, which are not necessary for detecting attacks.

RT-IoT2022 Attack Features Analysis and Distribution

The attack types and their distribution ratios were analyzed, primarily by calculating and visualizing the distribution of values for the 'Attack type' column and checking whether the column named 'Attack type' is present in the data frame to count the occurrences of each unique value in the 'Attack type' column. The result (attack counts) is a Pandas string where the index represents the attack types and the values represent the number of occurrences of each type. This is an important step to avoid errors if the column is missing.

The collected information will be shown on a chart, with the x-value representing various kinds of attacks and the y-axis expressing the number of occurrences of each attack type. Figure 1 displays an animation of the percentages of violence kinds.

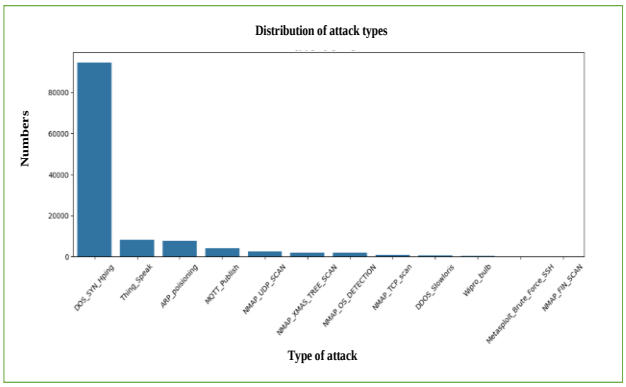


Fig 1. Types of attacks

RT-IoT2022 Attack Features Distribution

The direct relationships between the numeric variables in the data frame were examined and shown. First, columns with numeric data were chosen, and an association matrix was created between these columns. The ratio of correlation coefficients within each pair of variables are displayed in this matrix; values near 1 indicate a strong positive connection, values near -1 indicate a strong negative correlation, and numbers near 0 indicate a weak or negligible relationship. This aids in detecting possible multicollinearity and understanding the pattern of connections among numeric features in the data.

CGAN (Network of Generative Adversaries)

CGAN was proposed by Ian Goodfellow et al.[1] in 2014 in this paper. The GAN architecture is made up of two parts known as Generator and Discriminator. To put it simply, the generator's job is to produce new data (such as numbers or images) that closely corresponds to the dataset provided as input, at the same time the discriminator's job is to distinguish between generated data and actual input data [19]. In CGAN, the generator G tries to produce synthetic (fake) samples by imitating the distribution of real sample, while the generator receives a random vector z as input, which has the probability distribution pz. It transforms this vector into samples G(z) that follow the probability distribution pg. The discriminator D differentiates between real samples and those that are fake. As a binary classifier, it utilizes the labels 1 and 0 to denote real and fake data, respectively.

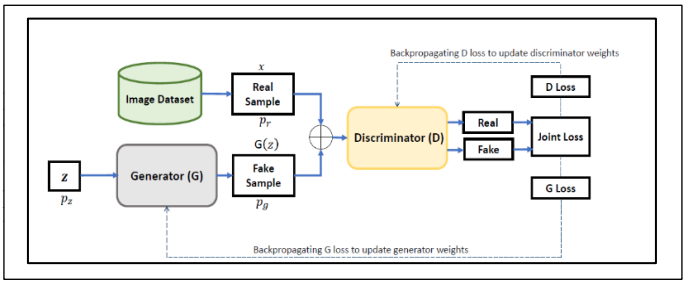


Fig. 2: Showes how the discriminator work

Artificial intelligence (AI) and machine learning (ML) have transformed the way commercial security functions in real-time by providing sophisticated intrusion detection techniques in contemporary security systems. To fully understand the significance and possibilities of AI-based intrusion detection, it is essential to comprehend the main algorithms and techniques used in these systems.

Machine learning relies heavily on intrusion detection systems (IDSs) to enhance their ability to detect and respond to security threats. IDSs use machine learning models to analyze network analyze system functioning and traffic patterns to spot unusual activities that might point to an attack (True Protection, 2024). Various machine learning algorithms can be trained on large datasets to identify complex patterns that are difficult to detect manually. Some common algorithms used in IDSs include the "LightCGM" algorithm, which is not a standard algorithm and is widely recognized for its specific set of equations in the field of intrusion detection. We can infer its potential mathematical structure as a lightweight conditional generative model designed to operate efficiently in IDSs.

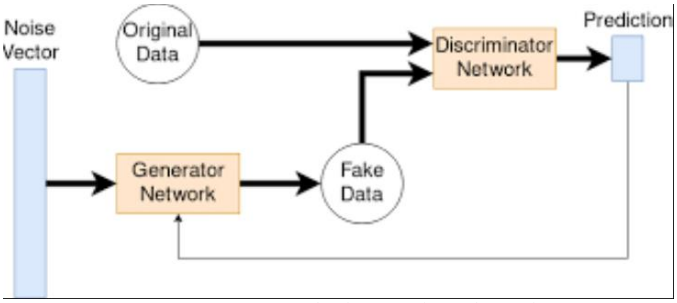


Fig. 3: CGAN (Network of Generative Adversaries)

3. How a GAN trains both generator and discriminator network at the same time

1. **Generators**

The model's Generator Network is the more intricate of the two. Using a random seed, often known as a noise sample, to train a "vanilla" GAN (the original Goodfellow model), the generator immediately begins generating samples. The initial endeavors yield poor results since they consist mainly of random noise. However, as the discriminator sends more feedback to the generator, the latter gradually enhances the quality of its samples, aligning them more closely with those in the training set. After convergence is achieved or the full number of training epochs has run, the Generator is typically the part of the model that is retained [16]. When the training is finished and the generator can create samples that are nearly indistinguishable from real ones, it is prepared to be employed for its intended use.

2. Discriminators

Usually, after the generator has been effectively trained, the discriminator in a model is removed [16]. Analyzing and correctly classifying the samples provided by the generator, regardless of their legitimacy, is the fundamental job of a discriminator. Over time, the discriminator should grow less effective, eventually resembling a computer lottery, as the generator receives feedback and gradually modifies its weights to create more accurate samples. Sometimes, the tool that discriminates wins the game, and depending on the kind of GAN model, this can produce an accurate and useful classifier [20].

3. Cost Functions

The training process involves a minimax game with the following loss functions:

Discriminator Loss (LD):

$$LD(\theta D, \theta G) = Ex \sim pdata(x), y \sim pdata(y)[logD(x,y)] + Ez \sim pz(z), y \sim pdata(y)[log(1 - D(G(z,y;\theta G),y))]$$

The discriminator aims to maximize this, correctly distinguishing real and synthetic data conditioned on y.

The minmax model, which is central to the GAN model, was introduced in Equation 1. The input $pz(z)$ includes the variable z, which serves as the generator's initial data, whereas the function p illustrates the noise distribution. $V\ (D, G)$ offers the value function where G is the generator and D is the discriminator, with the value function denoted as $G\ (z;\ \theta g)$. The result of the discriminator function $D\ (x;\ \theta D)$ is a likelihood value (just one integer), which shows whether the input x originated from either the generator or the training set.

$$minmaxV(D,G) = E_{X\sim pdata(x)}[log\ D(x)] \quad (1)$$

G D

$$+ E_{z\sim pz(z)}[log(1 - D(G(z)))] \quad (2)$$

Equation (1), this is the expected value (E) of the logarithm (log) of the probability that the Discriminator (D) assigns to a real data sample (x) being real.

Equation (2), this indicates that the generator tries to fool the discriminator into believing that the generated data G(z,y), conditioned on label y, is real and belongs to that classy.

Generating attack data and the models Definition using a GANM

At the beginning of the model construction, additional attack data is generated using GANM to learn the characteristics of real attacks and produce similar data. The generated data has been preprocessed by routine normalization (BatchNorm), which comes after a sequence of linear layers. A distribution that aims to differentiate between actual data and data produced by the generator comes next.

This model also consists of a series of linear layers, but the LeakyReLU activation function and Markovic property dropout layers are used to help prevent overfitting and improve stability during training. The network terminates with a single linear layer, which produces a value between 0 and 1 representing the probability that the input data is real and not generated. The forward function in both models determines how data flows through the model layers.

Training a Conditional Generative Adversarial Network (CGAN) Model

The data is trained on a conditional generative adversarial network (CGAN) model. The machine to be used for training is first determined (CPU or GPU). The algorithm that discriminates is initially trained to differentiate between the generator's bogus and authentic data. Both datasets are used to generate the discriminator's loss, and the discriminator's optimizer is used to update its weights. The generator is then trained to try to fool the discriminator by generating data that appears to be real. The generator loss is calculated based on the discriminator's ability to classify its data as real, and the generator's weights are updated using its optimizer. Training progress is displayed every 10 epochs. Finally, after training is complete, the generator and discriminator loss curves across epochs are plotted to demonstrate the training process, and the function returns the trained generator model. The technique of tricking both models into training this behavior is called adversarial training.

Fig. 4 illustrates the loss function and model training on the generated attack data. To evaluate the proposed LightGBM model, the collected data for selected features was randomly split into a training set (80%) and a test set (20%). Machine Learning (ML) provides an appealing answer to these difficulties, furnishing sophisticated analytical methods that can draw lessons from data and recognize patterns suggestive of intrusions. In contrast to conventional strategies, ML algorithms can adjust to new and changing threats, providing a proactive strategy for intrusion detection. Using ML for IDS requires training models on past data so they can identify known and unknown attack patterns, thus improving the system's capacity to detect various intrusions with great precision.

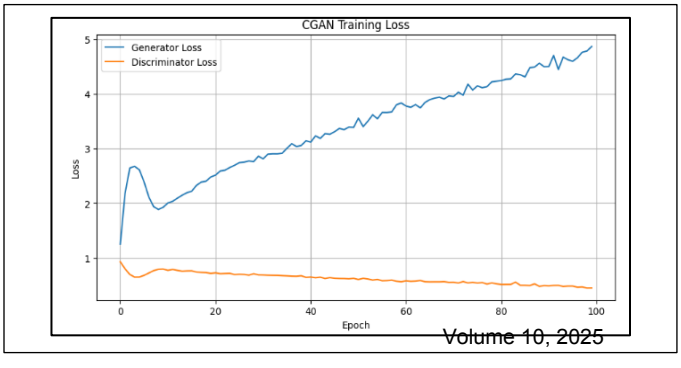


Fig 4. Illustrates the Loss Function and the Model's Training on the Generated Data.

Generating and Displaying Synthetic Data Similar to the Original Data

A. Data-Generating Models

The model responsible for generating data is the generator, which was trained as part of a Generative Adversarial Network (GAN). This model is trained within the GAN architecture to transform random noise into synthetic data, attempting to mimic the distribution of real data. The synthetic data is used to reverses the scaling process to return the data to its original scale. This makes the synthetic data more realistic and comparable.

B. Objective Function of GANs

The Objective Function of GANs an artificial neural network's objective function is the difference between the probability distribution of generated samples (pg) and the distribution of real samples (pr). The objective function is assessed using the binary cross-entropy loss. A joint loss function for the discriminator and generator is the binary cross-entropy V (D, G). It decreases the Jensen-Shannon divergence (JSD) between the produced and actual data distributions.

Applications of Gans in Detection Intrusion Attacks

In the field of attack intrusion detection, generative adversarial networks (GANs) have been used in innovative ways to enhance the capabilities of intrusion detection systems (IDSs).Recently researchers proposed an intrusion detection system based on a Conditional Tabular GAN (CTGAN) to address the data imbalance problem in Internet of Things (IoT) networks by generating synthetic data for rare attacks [21] Similarly, another study explored the use of GANs to improve the performance of intrusion detection in networks by generating diverse synthetic attack samples that help better generalize detection models against new attacks . Additionally, GANs have been used to generate adversarial samples to assess vulnerabilities in intrusion detection systems and enhance their robustness through adversarial training, as reported in a 2023 MDPI post [22] . The use of GANs has extended to specific environments such as SCADA systems, where GAN-based network intrusion detection systems have been developed and achieved high accuracy in anomaly detection [21] .

Evaluation Metrics

The following metrics were used in order to guarantee a comprehensive evaluation:

• Accuracy: The general correctness of the predictions made by the model.

• Precision: The percentage of actual positive forecasts out of all positive ones.

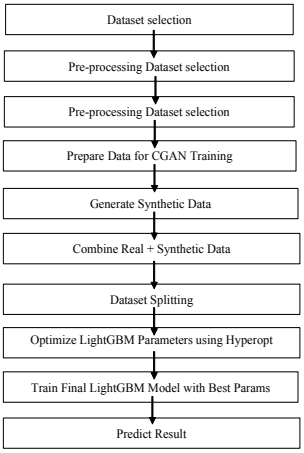
• Recall: The percentage of all true positives that are true positives.

• F1-score: A harmonic representation of recall and precision.

Classification of Performance

The LightGBM model's classification of performance for traffic-level anomaly detection was evaluated. Fig 5. Illustrate the Flowchart of Proposed Methodology.

Our work is the initial effort to combine spatial–temporal deep learning elements with standard machine learning algorithms in



a single ensemble structure, which is in contrast to previous research that assessed standalone ML or DL models. On the CIC-IDS2017 dataset, our hybrid algorithm enhances detection

Fig 5. Flowchart of Proposed Methodology

reliability and interpretability in conjunction with thorough preprocessing and soft-voting fusion.

The confusion matrix provides a visual illustration of the prediction performance. Figure 6 shows the LightGBM model evaluation report.

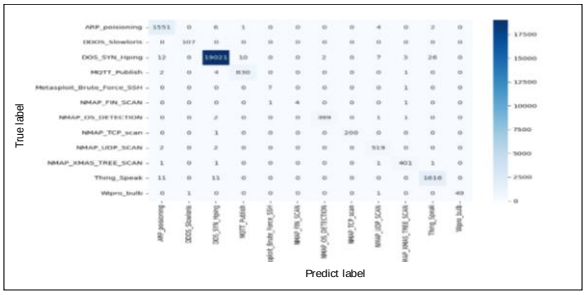


Fig 6. Confusion Matrix for Light GBM Model

Features Extraction

After data preprocessing, relevant features were selected and identified from the raw data to improve the predictive performance of the classified model for attacks. This step is essential because the quality and importance of features directly

impact the accuracy and efficiency of machine learning models. Fig 7. Shows Receiver Operating Characteristic (ROC) Curve.

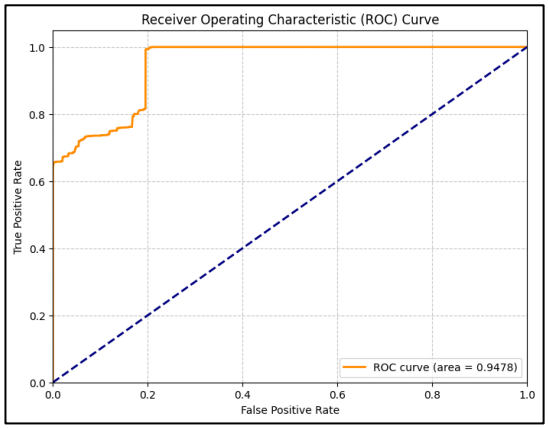


Fig 7. Receiver Operating Characteristic (ROC) Curve

The graph represents the receiver operating characteristic (ROC) curve, a tool for evaluating the performance of a binary classification model at different classification thresholds. The model's outstanding capacity to differentiate between the two groups is indicated by the very high AUC value of 0.9478. A high AUC value indicates that the model can maintain low rates of false positives while achieving a high true positive rate.
Detailed Classification Report:

Table 2. The Result of Accuracy

	precision	recall	F1-score	
Normal	0.84	1.00	0.91	439683
Attack	0.99	0.66	0.79	251723
accuracy	0.87	691406		
macro avg	0.91	0.83	0.85	691406
weighted avg	0.89	0.87	0.87	691406

The table displays the performance evaluation results of a binary classification model, which aims to distinguish between normal and attack traffic. The top part of the table represents the confusion matrix, which summarizes the model's performance by comparing the actual classifications with the predicted classifications. The matrix shows that the model correctly classified 438,627 normal instances, while misclassifying 1,056 normal instances as attacks. For attacks, the model correctly classified 165,114 attacks, but misclassified 86,609 as normal.

The Important of Features in the Model

This chart represents a horizontal bar showing the importance of the first 20 features in a trained machine learning model. This analysis of feature importance helps understand which features in network data were most significant in distinguishing between normal and malicious traffic for this particular model. This can provide valuable insights for improving the model, understanding network behavior, or even developing security strategies that focus more on these important features. Figure 8 shows the importance of 20 features in the proposed model.

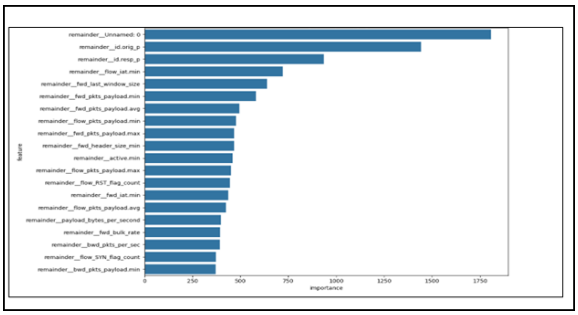


Fig 8. Features Importance in the Proposed Model

4. Discussion and Results

To assess the offered breach detection model in light of these findings, accuracy reflects the total efficacy of the model. However, it might be deceptive on data that is unbalanced. The training and testing data's quality implies that it depicts realistic situations is clean, and free of significant biases, and that the process of splitting the data into training and testing sets was correct and random to avoid data leakage.
The confusion matrix provides a breakdown of the number of correct and incorrect classifications for each class, helping us understand the types of errors the model makes. Here, the model appears good at classifying normal cases but misclassifies a large number of attacks as
Overall, although the model demonstrates high accuracy and excellent accuracy for predicted attacks, the relatively low recall of attacks raises some concerns about its ability to detect all real attacks. Thorough validation requires a thorough evaluation of all metrics, understanding the application context (what is the cost of missing an attack compared to false alarms?), comparing performance with other models, and testing the model on new data. Based on the code output, the results of the evaluated model can be summarized in the following Table 3:

Table 3. The Result of Accuracy

Accuracy:	**(87.32%)**
Precision for Attack:	(99.36%)
Recall for Attack:	(65.59%)
F1 Score for Attack:	(79.02%)
Area under the ROC Curve (AUC):	(94.78%)

For the "Normal" class, the model achieved a precision of 0.84, a recall of 1.00, and an F1 of 0.91. For the "Attack" class, it achieved a high precision of 0.99, but a relatively lower recall of 0.66, resulting in an F1 of 0.79. The high precision for the "Attack" class indicates that the model is very accurate when predicting an attack, but the low recall indicates that it misses a large number of actual attacks.

Analysis of the Confusion Matrix Table and Classification Report

The table shows the performance evaluation results of a binary classification model. The first part is the Confusion Matrix, which compares the actual classifications with the model's predicted classifications. The second part is the Classification Explanation, which provides more detailed performance metrics for each class (Normal and Attack) in addition to the overall metrics. Table 4. Show the Numbers of normal and attack predictions

Table 4. Number of normal and attack predictions

class	Predicted Attack	Predicted Normal
Actual Normal	438627	1056
Actual Attack	86609	165114

Computational Cost:

A workstation with an Intel Core i7 (3.6 GHz) CPU, NVIDIA RTX 3080 GPU, and 32 GB RAM had been used for all experiments. It took almost 4 hours to train each fold, using an elevated memory use of 9 GB.

5. Conclusion

An entirely novel approach is proposed that blends GANs into the Light model structure for neural networks to overcome the essential issue of data exclusion in IDS training datasets. The suggested method entails creating and deploying a unique GAN model and producing artificial network traffic data that faithfully replicates actual network activity.

Through extensive experiments on a benchmark dataset (RT-IoT2022) and generated datasets, the results indicate that incorporating GANs into the Light model can lead to improvements in intrusion detection performance. A machine learning model for network intrusion detection, the LightGBM model, was evaluated. The top 10 features that contributed to the model's predictions were presented, providing valuable insights into the most indicative factors of intrusion in the data. The program's ability to provide a comprehensive evaluation of the model can be assessed using multiple metrics. The evaluation results obtained indicate that the model achieved an accuracy of 0.8732. It also demonstrated high attack precision (0.9936), meaning that when the model predicts an attack, it is highly accurate. However, the recall of attacks was average (0.6559), indicating that the model misses some actual attacks. The F1 measure for attacks (0.7902) reflects this balance between precision and recall. The confusion matrix indicates that the model is good at classifying normal cases but commits a high number of false negatives. Finally, the area under the ROC curve (AUC) of 0.9478 indicates the model's good ability to distinguish between normal and malicious traffic.

References

[1] B. Patrick and F. Huston, “A Survey on Supervised vs Unsupervised Learning Models for Network Intrusion Detection,” 2025.

[2] A. Shaout, NevrusKaja+, and M. Borovikov++, “Security Solution for Cloud Computing Using a Hardware Implementation of AES Adnan,” 2014. doi: 10.13140/2.1.4072.8321 CITATIONS.

[3] H. Iguer, H. Medromi, A. Sayouti, S. Elhasnaoui, and S. Faris, “The Impact of Cyber Security Issues on Businesses and Governments: A Framework for Implementing a Cyber Security Plan,” in 2014 International Conference on Future Internet of Things and Cloud, 2014, pp. 316–321. doi: 10.1109/FiCloud.2014.56.

[4] G. Karatas, O. Demir, and O. Sahingoz, “Deep Learning in Intrusion Detection Systems Gozde,” 2018, Ankara, Turkey. doi: 10.1109/IBIGDELFT.2018.8625278.

[5] M. Mahmoud, Y. Youssef, and A. Abdel-Hamid, “XI2S-IDS: An Explainable Intelligent 2-Stage Intrusion Detection System,” MDPI, vol. 17, p. 28, 2025, doi: 10.3390/fi17010025.

[6] S. Tripathy and D. B. Behera, “A Review of Various Datasets for Machine Learning Algorithm-Based Intrusion Detection System: Advances and Challenges,” Intell. Syst. Appl. Eng., no. 2147–67992, p. 26, 2024.

[7] O. Ogundairo and P. Broklyn, “Machine Learning Algorithms for Intrusion Detection Systems,” J. Cyber Secur., 2024.

[8] I. Martinsa, J. Resendea, P. Sousaa, S. Silvaa, L. Antunesa, and J. Gamaa, “Host-based IDS: a review and open issues of an anomaly detection system in IoT,” 2022.

[9] A. Ferdowsi and W. Saad, “Generative Adversarial Networks for Distributed Intrusion Detection in the Internet of Things,” 2019.

[10] M. Ferrag, D. Hamouda, M. Debbah, L. Maglaras, and A. Lakas, “Generative Adversarial Networks-Driven Cyber Threat Intelligence Detection Framework for Securing Internet of Things,” 2023.

[11] B. Alabsi, M. Anbar, and S. Rihan, “Conditional Tabular Generative Adversarial Based Intrusion Detection System for Detecting Ddos and Dos Attacks on the Internet of Things Networks,” MDPI, vol. 23, p. 20, 2023, doi: 10.3390/s23125644.

[12] S. Li, Y. Cao, S. Liu, Y. Lai, Y. Zhu, and N. Ahmad, “HDA-IDS: A Hybrid DoS Attacks Intrusion Detection System for IoT by using semi-supervised CL-GAN,” Expert Syst. Appl., vol. 238, p. 122198, 2024, doi: https://doi.org/10.1016/j.eswa.2023.122198.

[13] K. Singal, N. Kandhoul, and S. K. Dhurander, “Evolutionary LightGBM-Based Intrusion Detection System for IoT Networks,” Int. J. Commun. Syst., vol. 38, no. 5, p. e70031, Mar. 2025, doi: https://doi.org/10.1002/dac.70031.

[14] Y.-M. Yang, K.-C. Chang, and J.-N. Luo, “Hybrid Neural Network-Based Intrusion Detection System: Leveraging LightGBM and MobileNetV2 for IoT Security,” MDPI, vol. 17, p. 19, 2025, doi: 10.3390/sym17030314.

[15] S. A. H. Ayubkhan, W.-S. Yap, E. Morris, and M. B. K. Rawthar, “A practical intrusion detection system based on denoising autoencoder and LightGBM classifier with improved detection performance,” J. Ambient Intell. Humaniz. Comput., vol. 14, no. 6, pp. 7427–7452, 2023, doi: 10.1007/s12652-022-04449-w.

[16] O. H. Abdulganiyu, T. Ait Tchakoucht, and Y. K. Saheed, “A systematic literature review for network intrusion detection system (IDS),” Int. J. Inf. Secur., vol. 22, no. 5, pp. 1125–1162, 2023, doi: 10.1007/s10207-023-00682-2.

[17] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, and B. D. Payne, “Evaluating Computer Intrusion Detection Systems: A Survey of

Common Practices," ACM Comput. Surv., vol. 48, no. 1, Sep. 2015, doi: 10.1145/2808691.

[18] J. I. I. Araya and H. Rifà-Pous, "Anomaly-based cyberattacks detection for smart homes: A systematic literature review," Internet of Things, vol. 22, p. 100792, 2023, doi: https://doi.org/10.1016/j.iot.2023.100792.

[19] M. Saad, R. O'Reilly, and M. Rehmani, "A Survey on Training Challenges in Generative Adversarial Networks for Biomedical Image Analysis," 2022. doi: 10.48550/arXiv.2201.07646.

[20] B. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, and Y. Bengio, "Generative Adversarial Networks," Commun. ACM, vol. 63, p. 6, 2020, doi: 10.1145/3422622.

[21] S. Aldhaheri and A. Alhuzali, "SGAN-IDS: Self-Attention-Based Generative Adversarial Network against Intrusion Detection Systems," MDPI, p. 18, 2023.

[22] X. Zhao∗, K. Fok, and V. Thing, "Enhancing Network Intrusion Detection Performance using Generative Adversarial Networks Xinxing," ST Eng. Singapore. arXiv2404.07464v1, p. 12, 2024.

Contribution

This paper presents a hybrid progressive inference a position that brings together traffic- and packet-level analysis to identify intrusions in IoT networks in a thorough and extensible ways. It has a strong, cheap detection technology that reduces false positives and ensures consistent performance across many different devices. Furthermore, the study analyzes creative techniques for delivering and integrating attack examinations into the database, which different from previous strategies to improving model efficient execution.