National College of Ireland

Detecting Adversarial Network Behaviors in IoT Environment

MSc Research Project
Data Analytics

Tejas Sandeep Bafna
Student ID: x23211741

School of Computing
National College of Ireland

Supervisor: William Clifford

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Tejas Sandeep Bafna
Student ID:	x23211741
Programme:	Data Analytics
Year:	2024-25
Module:	MSc Research Project
Supervisor:	William Clifford
Submission Due Date:	12/12/2024
Project Title:	Detecting Adversarial Network Behaviors in IoT Environment
Word Count:	7611
Page Count:	23

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Tejas Sandeep Bafna
Date:	12th December 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Detecting Adversarial Network Behaviors in IoT Environment

Tejas Sandeep Bafna
x23211741

Abstract

The Internet of Things (IoT) has expanded rapidly across industries and enterprises bringing innovation and value to various sectors, while at the same time exposing critical cybersecurity risks arising from the growing complexity, heterogeneity, and resource constraints of IoT systems. IDS are not able to deal with threats such as zero-day threats, insider threat, encrypted traffic, polymorphic viruses and traffic camouflage. They also experience issues with low-and-slow attacks, IoT exploits, and Advanced Persistent Threats (APTs) that are stuck in normal behavior patterns, thus leaving the opportunity for detection gaps and false negatives. In the context of this work, deep learning models for IoT intrusion detection are examined, with a special emphasis on CNNs and the proposed Conv-GAN model for data augmentation. The standalone CNN model was tested with the RT-IoT 2022 dataset and showed excellent performance with 99.3% accuracy and good detection of most of the attacks. The Conv-GAN part of the feature set, when combined with synthetic data to tackle class imbalance, showed difficulties in synthetic data quality and incorporation leading to decreased performance compared with CNN. The results presented in this paper confirm the ability of CNNs and prove their potential for IoT intrusion detection, as well as identify further development possibilities for hybrid models.

1 Introduction

The growth of the Internet of Things (IoT) has exponentially increased in recent years, transforming connectivity by allowing the exchange of data between various objects across industries such as healthcare, manufacturing, and consumer goods. This transformation, however, comes with several security challenges. Yaras and Dener (2024) Predicted the number of IOT devices in both the consumer and enterprise sectors is expected to surpass 75 billion by 2025. As IoT systems become larger and more complex, they are increasingly vulnerable to various forms of cyberattacks, including malware, Distributed Denial of Service (DDoS) attacks, and intricate intrusions Kwon et al. (2022). Due to the heterogeneity of the IoT systems, which encompasses heterogeneous hardware platforms, operating systems, and communication protocols, IoT systems are highly vulnerable to adversarial threats. Therefore, Intrusion Detection Systems (IDS) are essential in IoT networks defense against such attacks. Nevertheless, traditional IDS techniques, such as signature based detection, are not suitable for new or new attack vectors. Anomaly based detection methods have been developed to overcome these limitations. IDS systems create baseline standards to monitor normal traffic activity while triggering alerts when traffic stray from that benchmark. Anomaly detection techniques show effectiveness but struggle with excessive incorrect alerts specifically for new and evolving IoT network environments. Recent advancements in machine learning technologies and deep learning methods including Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs) present possibilities to upgrade ID System (IDS) performance so they can more effectively protect IoT networks from such security threats. When attacks rely on newly developed vector signature-based detection approaches proven ineffective for this situation.

Researchers introduced anomaly detection strategies because they addressed previously existing limitations. This detection system runs defined baseline values on normal network activity and flags all divergences as possible cyberattacks. Anomaly-based detection demonstrates success but its implementation faces difficulty due to elevated false positives in IoT networks undergoing continuous change. Recent developments in deep learning techniques with CNNs and GANs present opportunities to boost IDS performance levels. CNNs excel at identifying patterns across large datasets but GANs generate synthetic attack data to solve minority class distribution problems.

A new hybrid Conv-GAN framework unites Convolutional Neural Networks and Generative Adversarial Networks to develop specialised intrusion detection systems for Internet of Things environments. The research integrates advanced feature extraction and sophisticated data synthesis beyond fundamental GANs combined with CNN only systems for more efficient class imbalance correction. The proposed model targets real-time IoT processing in resource-limited IoT systems because it outperforms standard models which do not consider efficiency or scalability requirements. Research potential is high because the immediate requirement exists for better detection of advanced IoT cybersecurity threats. The outstanding detection capabilities of CNNs for malicious activities remain limited because of insufficient labeled data that fails to include new kinds of attacks. GANs help address this problem through their ability to create synthetic data yet issues remain during their combination with deep learning frameworks. This research undertakes a study of security challenges while evaluating the performance of a proposed Conv-GAN model that merges CNN and GAN techniques to produce synthetic data. This research initiative seeks to enhance IoT network IDS detection through advanced scalability and flexibility and effectiveness for operational environments.

1.1 Research Background

There are a large number of devices connected to IoT networks now, and the security threats to these networks are incredible. However, maintaining the privacy, integrity, and access of the IoT networks is where IDS is most important. However, traditional IDS solutions are insufficient in responding to the sophisticated and dynamic nature of the IoT environment. The security threat to new complex IoT networks is also getting more complex as IoT networks become more advanced. Deep learning has been shown to be powerful for learning network behaviors to detect unknown threats with Artificial Intelligence. However, these models suffer from scarcity of labeled data for training and imbalance in attack distribution. Using synthetic data, such as that generated by GANs, has been proposed as a solution to address these issues, but integrating GANs with deep learning models remains an area for further research. This study seeks to address this gap by proposing the Conv-GAN model, a hybrid approach that combines CNNs and GANs to identify intrusions in IoT networks. The research aims to enhance the quality of IDS solutions and provide new perspectives on tackling data deficiency and class imbalance issues in the IoT context.

1.2 Research Questions

- What is the performance of CNN models as standalone intrusion detection systems in IoT environments, particularly in addressing diverse attack types and class imbalance?
- What challenges arise from integrating GAN-generated synthetic data into deep learning-based intrusion detection systems, and how do these challenges affect detection accuracy?

- How does the quality of GAN-generated synthetic data influence the overall performance of intrusion detection systems in IoT environments?
- What improvements or alternative methods can address class imbalance without compromising model accuracy in IoT intrusion detection systems?

1.3 Research Objectives

The main aim of this work is to assess the effectiveness of the proposed hybrid deep learning models for intrusion detection in IoT networks. Specifically, the study will evaluate the performance of standalone CNNs in terms of detecting multiple attack types and addressing class imbalance. Additionally, the research will explore the challenges associated with training models using GAN-based synthetic data. The findings will provide insights into how IDS models can be improved, as well as how synthetic data can be integrated without compromising detection accuracy.

2 Related Work

The rapid development of Internet-of-Things (IoT) technology now impacts diverse industries and scientific achievements in sectors throughout the speed of implementation. The fast growth of IoT infrastructure created multiple cybersecurity risks because IoT networks rely on heterogeneous architectures built for interconnection. As IoT devices proliferate the number of vulnerabilities increases to benefit attacking opportunities in networks. Mazhar et al. (2023) notice that the complicated nature of IoT environments leads to poor security outcomes. The current inadequate collection of effective security measures makes critical data and systems susceptible to major defects and breaches.

ML and Deep Learning In response to these concerns, researchers are gradually investigating integrated IDS and IDP solutions for IoT networks. Ling et al. (2023) notes that traditional approaches to cybersecurity fail in IoT circumstances and that anomaly detection and predictive analytics are critical concepts in ML to prevent cyber threats. The incorporation of these methods makes it easier also to notice any aberrations and intrusions in real-time which will go a long way in strengthening the IoT systems. In addition, Alsoufi et al's systematic review shows how deep learning models can be used in creating a specific IDS for IoT systems, which are based on anomaly- based detection Alsoufi et al. (2021).

Hnamte and Hussain (2023) introduced a new model which was a deep learning based hybrid approach, combining both the prowess of the deep convolutional neural networks alongside the bi-directional long short-term memory networks, resulting in a peculiar named model DCNNBiLSTM. This model was used for the feature extraction while also capturing the intricate temporal dependencies present inside the network traffic data for greater understanding and accuracy. In the research paper, they proposed this model which could notably reduce the false positive rate encountered during the epoch phase of a model's life on testing for the malicious activity within the IoT traffic Network. This model's architecture revolved around the combination of multiple architectures which could capitalize on the machine learning models strengths, thus creating a complex Network environment with better and robust solution.

In 2022 a new study came out which proposed our dual integrating signature based alongside behavior-based technique, proven to enhance the ideas performance across all IoT Network, thus reducing reliance on the predefined attack signatures while also increasing and enhancing the security of the IoT Network itself Kwon et al. (2022). In their research paper the intuition

detection system presented contained both the statistical filtering that is signature based and also a component of the behavior-based detection that is auto encoders . This model initially tried to filter out and shift through various known attacks using signature-based methods Benaddi et al. (2022) but with its advanced behavior-based detection reduced the workload required for the module to work. This intern reduces the workload of the deep learning module for anomaly detection which could deviate from the normal behavior, thus resulting in a high performance gain. This study showed that the tests were conducted on the DS2OS benchmark, on which the model performed outclass and the approach was particularly relevant towards the goal of our research where the volume of the data could be strained on the deep learning models used in the IoT Network.

In the study Aslan et al. (2021), explored the behavior based malware detection system but extended it to the cloud environment and gained significant recognition for their approach. In their approach, the use of the random Forest technique in combination with the API calls sequences gave the researchers a better accuracy boost to both the efficiency and security in the detection of the attack signatures. This was not only valuable to the IoT environment security concerns, but also to the ever evolving malicious attacks present. In the IoT settings and the context of the prevention of such cyber security attacks, the cloud services approach shows that the potential in creating comprehensive cloud compatible IDs solutions requires a combination of the machine learning approach like the random Forest used in this paper Bao et al. (2021).

In one of the studies of 2021, the researchers found out that the continuous static models could degrade performance over time in the context of the cyber security challenges faced by the IoT Network. Darem et.al researched into the solution of this problem where he proposed that the malware detection model could use incremental batch learning and the concepts of the drift detection could be enhanced with the behavioral-based detection models. This new adaptation of the patrons in the malware detection, made sure that the models used in the research were effective without retraining from scratch Huang et al. (2020). This made the security of the dynamic iot environment, particularly suitable with the models, incremental learning capability and ensured that the security of the iot environment is never compromised by making sure that the attack signatures of the various malware's and the emerging threats are dealt in real time malware detection Wu et al. (2021). While not explicitly for the iot environment, the research proposed by the Dutt et al. (2020) proved to be an immune inspired IDS which could mimic the biological immune systems adaptive response into the computational environment of the high load iot Network. In their research paper, they established that the combination of the negative selection and the clonal selection algorithms can recognize the attack signatures of the malicious iot services with better accuracy than the traditional machine learning models presented.

Lansky et al. (2021) Conducted a systematic review of the various deep learning based IDS , and also made sure to evaluate the various architectures and data sets presented in this domain to solve the problem with the advanced techniques. In his hybrid approach, which combined multiple deep learning techniques for enhanced detection and accuracies, he highlighted the increasing trend for the detection of the malicious scripts and attack signatures in the overall network of the iot. He emphasized that the various architectures such as CNN's LSTMs and generative adversarial networks, although offered a promising approach to balance out both the accuracy and processing speed, lacked in a adaptability paradigm. His review on the underscores of the necessity required for the hybrid models in the IoT to work was met with resounding success from the academic peers who acknowledged his understanding of the high variability in the data traffic and the unique challenges posed by them Nie et al. (2021).

Liu et al. (2021) reviewed rule-learning based IDS, focusing on their videos and few but mostly on the applicability side in the smart grid environments. While their focus was on smart grids, many of the challenges discussed are relevant to IoT, such as scalability, adaptability, and the need for real-time processing but not only stopping over there in their research, they also focused on the hybrid approach proposed by the combination of various machine learning and deep learning algorithms. The study examined techniques like association rule mining and sequential pattern mining, which could be adapted to IoT for detecting complex, multi-step attack patterns. The findings, although complex on the data streams and the IoT Network high payload environment, suggest that incorporating rule-learning approaches with deep learning architectures may provide a more comprehensive solution for IDS in IoT with the combination of the various machine learning algorithms such as random Forest, SVG, XG boost etc.

Building upon the need for adaptable and robust detection methods, hybrid models combining deep learning with traditional approaches offer promising avenues for IoT security and this is further enhanced by the need of the iot security against the malicious attack signatures present in the network. By leveraging the feature extraction strengths of CNNs and sequence learning of LSTMs, these models can capture both spatial and temporal aspects of network behavior, which are essential for detecting adversarial activities in IoT Ullah and Mahmoud (2021). Additionally, behavior-based approaches using autoencoders or Random Forest classifiers provide a layer of protection against the ever- evolving nature of the malicious user scripts as well as the attack signatures by offering the adaptability which could focus on deviation from normal behavior rather than static signatures, essential in environments like IoT where attack patterns evolve rapidly.

Network security challenges for IoT benefit from the exceptional characteristics of GANs combined with CNNs. The network architecture allows CNNs to bring effectiveness to intricate spatial relation examination alongside high-dimensional feature extraction from network traffic for IoT interaction analysis. According to Liu and Zhang CNN systems work effectively with multidimensional data storage arrays while their built-in feature extraction capabilities are crucial for IoT security detection Liu and Zhang (2020). According to Su et al. (2018) CNNs demonstrate scalability enabling them to extract basic features from data which makes them optimal for constrained IoT devices. Precise data pattern recognition functions as an essential requirement for IoT security threat detection in networks consisting of diverse and complex data structures Xiao et al. (2019).

GANs offer multiple solutions to address class imbalance which commonly occurs in intrusion detection datasets. By generating low-frequency attack patterns GANs help reduce datasets' imbalance while also enhancing detection capabilities. GANs apply to decentralized intrusion detection for IoT systems through anomaly detection according to Ferdowsi and Saad (2019). According to Cheng (2019) GANs successfully create network traffic data packets that help build better IDS systems. The synthetic data generated proves particularly useful when attack frequency is low because it enhances model learning through diverse training data Almarshdi et al. (2023).

Combining CNNs and GANs results in a synergistic model with the capabilities of both CNNs and GANs that was not present in either of the two models individually. CNNs are particularly used in classification and identifying patterns while on the other hand GANs are used in creating new samples in a dataset. This interlinkage is especially advantageous when dealing with IoT unique issues, like heterogeneity of data, fluctuating traffic rates, and a low occurrence rate of particular cyber threats. For example, the integration of CNNs with GANs can provide a solution

to enhancing the flexibility and extensibility of intrusion detection systems because traditional approaches are challenging to scale to meet current threats Odeh (2023). In turn, the increase in the general accuracy of detection allows the hybrid model to be considered as a promising option to protect IoT networks from various cyber threats Sayed et al. (2023).

3 Methodology

This section of the research study shows the complete methodology such that it develops and evaluates the proposed CNN, GAN and Conv-GAN hybrid model for detecting adversarial network behaviors in IoT environments.

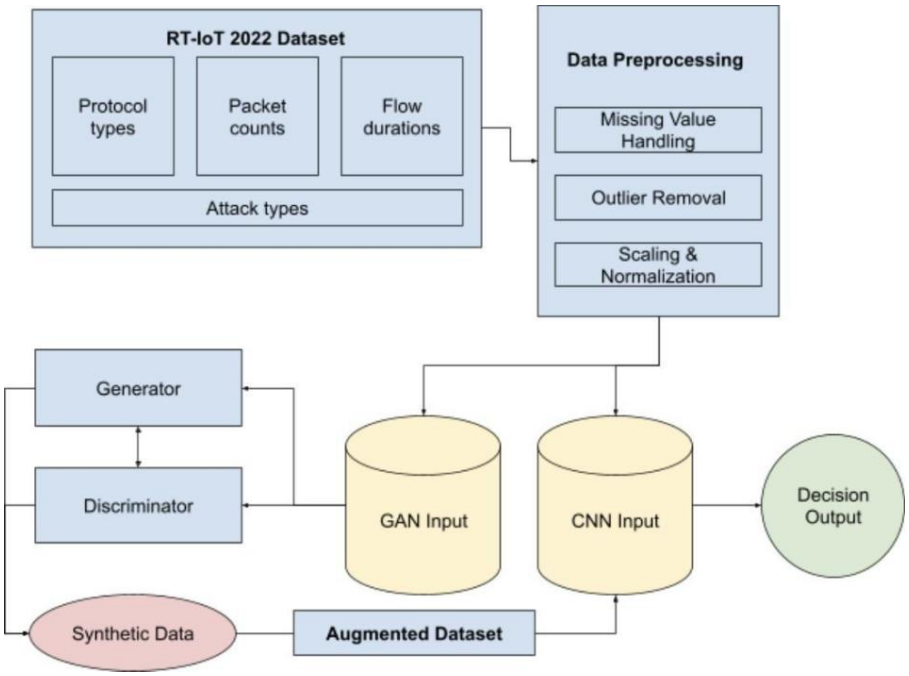

Figure 1: Process Flow Diagram illustrating the steps involved in developing and evaluating the proposed Conv-GAN hybrid model.

As shown in Figure 1, the methodology begins with data collection from the RT-IoT 2022 dataset, followed by preprocessing steps to clean and normalize the data. The dataset utilized in this study is the RT-IoT 2022, which provides comprehensive IoT network traffic simulations (UCI Machine Learning Repository, 2022). Subsequent stages involve exploratory data analysis to understand data distribution, feature engineering to select relevant attributes, and the development of the CNN and GAN models. The GAN is utilized for data augmentation to address class imbalance, after which the CNN is trained and evaluated using the augmented dataset. This methodology section consists of data collection and preprocessing, exploratory data analysis, feature engineering where each step addresses the unique challenges which were posed by the IoT networks and these challenges included a high data variability, class imbalance, and the need for real-time intrusion detection. Execution of process flow diagram in Figure 1 explain as follows:

3.1 Data Collection

The dataset used for this work is the RT-IoT 2022 dataset obtained from UCI machine learning repository because it includes a rich IoT network traffic simulation. This dataset covers both normal behavior and different types of attacks, among which are Denial of Service (DoS), Distributed Denial of Service (DDoS), and malware attacks. Currently, it contains over 123,000 records and 85 different features, which provides a solid foundation for analysis. Some of the recorded metrics include network protocol, flow duration, forward and backward packet counts, payload size, inter-arrival time, and active/idle time. Additionally, the target variable, attack type is categorical and classifies traffic as either non malicious or associated with specific types of attacks. The inclusion of real-world data and diverse attack scenarios makes this dataset highly suitable for testing the effectiveness of the proposed model in identifying and categorizing threats within an IoT network.

3.2 Data Preprocessing

3.2.1 Data Cleaning and Transformation

Preprocessing is one of the most significant steps when performing any machine learning study to obtain meaningful results. In order to diagnose missing values in this study, the null function was used and missing data were handled depending on the type of the missing data. To handle the missing values numeric missing values were imputed with median of the respective feature since they are signs of outliers while categorical missing values were imputed with the mode which is the frequent value of the feature. The use of duplicate function to find all duplicate entries which could lead to bias during the training of the model and these were deleted. Both the detection and treatment of outliers were done by using the Interquartile Range (IQR) where any number that fell below 1.5 IQR from the first quartile or above 1.5 IQR from the third quartile was considered an outlier. Moreover, features were being normalized and data points with z-score between over 3 or less than -3 were being considered as outliers. These few cases were either truncated at the 5th and 95th percentiles of their respective distributions or simply dropped out to obtain more refined inputs for the subsequent modeling steps.

3.2.2 Data Normalization and Scaling

In order to maintain model stability and to achieve faster convergence of the gradient descent, all feature scaling was performed as follows:

- Numerical features were normalized to have a zero mean and unit variance using StandardScaler.
- Some features were standardized to a range between 0 and 1 when required.
- Encoding Categorical Variables:
  - The proto and service categorical features were label-encoded to transform them into numerical values.
  - For the target variable Attack type, one-hot encoding was discussed, but label encoding was ultimately used.
  - Ensured the compatibility of the features for model input, ensuring no feature was in an inappropriate format for the model.

3.3 Exploratory Data Analysis (EDA)

To gain insights into the data distribution, identify important features, and understand patterns of different attack types, an exploratory data analysis (EDA) was conducted. The results of the EDA revealed a significant class imbalance in the dataset:

- **Dominant Attack Types:** DOS_ SYN Hping accounted for approximately 76% of the dataset, indicating the major presence of DoS attacks in the collected traffic.
- **Minority Classes:** Attacks such as Metasploit Brute Force SSH and NMAP FIN SCAN were underrepresented, each contributing less than 0.1% of the dataset.

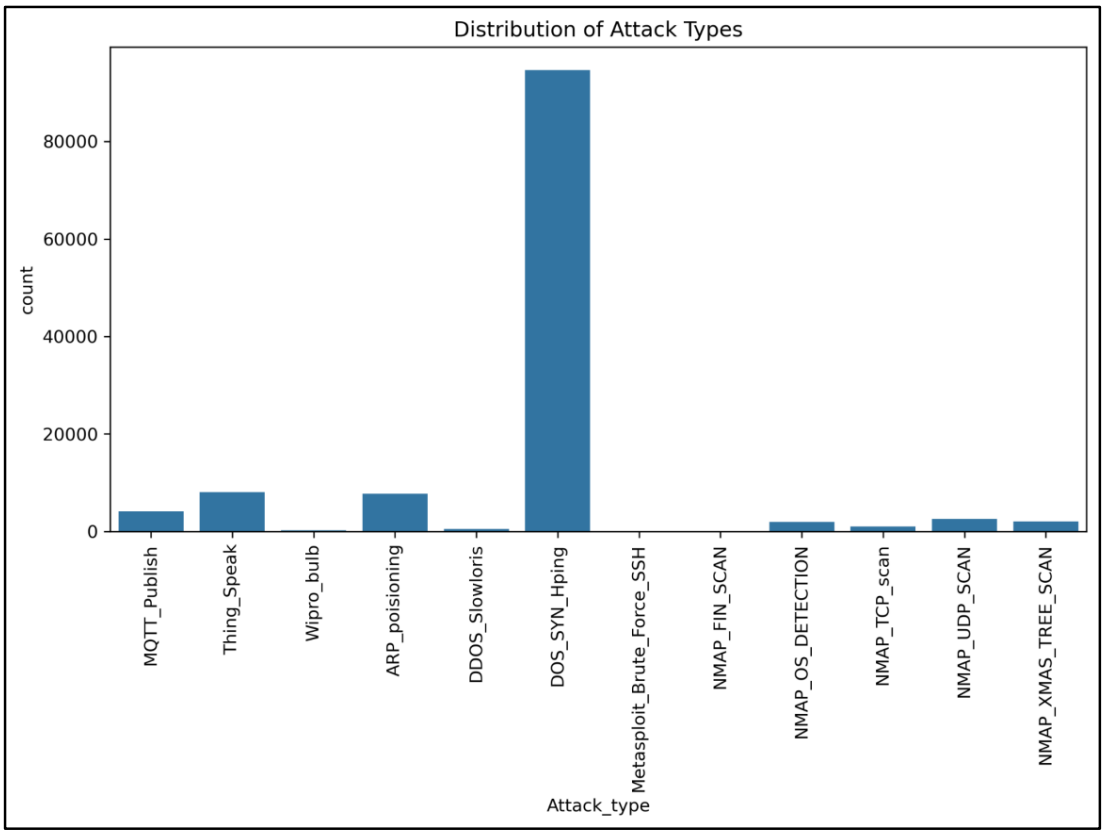

Figure 2: Illustration depicting various attack types and their count in the dataset

From the preliminary exploration of the data, there were some important things to note with regards to the class imbalance and the characteristics of the data set whereby methods to deal with the class bias that may occur in the modelling process were also learned. Majority of the observed flows were determined to have low staying times with a mean of 3.81 secs and large standard deviation of 130.01 secs, implying variability. Forward and backward packet totals showed low means along with high standard deviations, which indicate high packet fluctuations. Furthermore, the first forward and backward window sizes were left-skewed, as with the DOS_ SYN_ Hping attack.

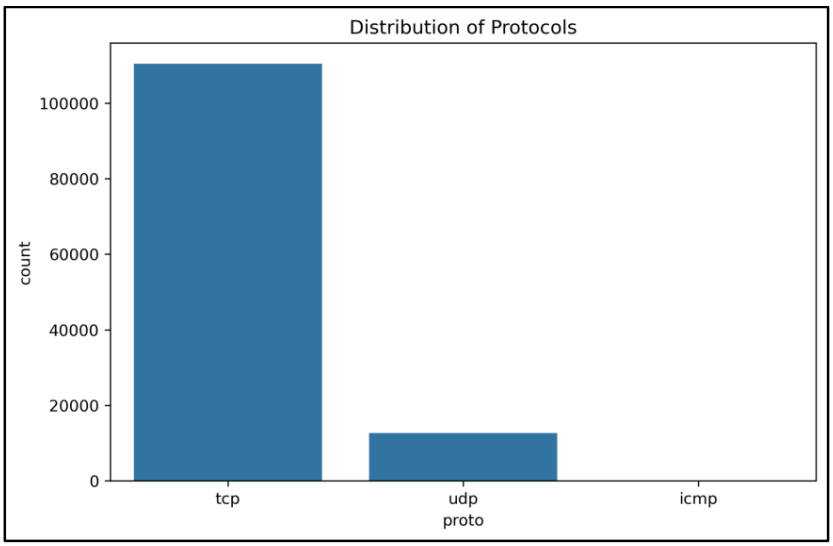

Figure 3: Bar graph showing distribution of the various protocols

In order to beforehand define these characteristics, several visualizations were made. Histograms showed numeric characteristics, like flow duration, total forward packets and total backward packets. Count plots shown in Figure 3 depicted the proportion of attack types and protocols, with the result showing that attackers prefer to use TCP. Simple linear regression plots were used to identify correlation between the features for instance flow duration against forward packets totals. Finally, the correlation between features was tested and, to do so, heatmaps were created to better understand the structure of the used dataset.

3.4 Modelling

The proposed Conv-GAN architecture combines a Convolutional Neural Network for feature extraction and classification with a GAN that addresses the class imbalance of IoT traffic data. To capture other patterns of the CNN the Conv1D layers of 64 and 128 filters are used, max pooling, dropout to avoid overfitting and finally the dense layers with softmax for the multi-class classification. It is compiled with the Adam optimizer with the learning rate equal to 0.001. The GAN synthesises samples for minority classes including a generator that creates new data from noise and a discriminator that distinguishes between real and synthesized data.

An adversarial training method based on binary cross-entropy for data augmentation and an enhanced modeling the present study, the use of CNN layers with 64 and 128 filters has been adopted bearing in mind the performance and efficiency considerations within an IoT network traffic context. The first layer with 64 filters aimed at detecting relatively simple patterns which include shapes and low level features in order to easily understand the architecture of the network traffic. This is in agreement with Su et al., who observe that CNNs can be fine tuned to pick basic features from data, which makes them appropriate for the IoT devices who are resource limited Su et al. (2018). The second layer with 128 filters comes on top of this by identifying even the interactions of more complicated features of the network behavior and the possible attacks. This layered approach is in agreement with Fu et al. Observing that the number of filters must be changed in order to enhance the classification accuracy in the case of network traffic Fu et al. (2023).

When using fewer filters, for example 32, events with important details were filtered out thus reducing the accuracy of the model. Regarding this analysis, Alabsi et al. (2023) also establish that lack of adequate filter sizes compromises feature extraction, which impacts IDSs' performance. On the other hand, extending the filter amount to 256 highly affected the model's efficiency and consumed a lot of memory, which is not suitable for IoT devices. These problems are not unique to this study but are standard issues in deep learning as highlighted by Wang et al. (2023). Hence, after much experimentation, the number of filters was decided to be 64 and 128 since it offered both, precision to identify threats and the necessary speed of IoT applications.

Therefore, the proposed decision of using 64 & 128 filter CNN layers has empirical evidence and more research is ongoing to improve deep learning models for IoT security. This way the configuration ensures that it captures key aspects of the IoT traffic while at the same time keeping the model light and capable of handling the dynamic nature of the traffic

3.5 Model Evaluation

The models were then evaluated by the test set and performance of the standalone CNN compared to the Conv-GAN hybrid was done. Metrics of interest in the assessment included accuracy, defined as the percentage of correctly classified samples, and generalizing about model performance. It was particularly emphasized that precision and recall of the model were evaluated, to give balanced performance in all categories. A confusion matrix was also used to visualize misclassifications, to provide insights into where to improve specific areas by showing patterns of error across different classes.

3.6 Hyper-Parameter Selection

The hyper-parameters for both the CNN and GAN models were selected using a grid search approach to systematically explore combinations of parameters and identify the set that yielded the best performance on the validation set. Parameters such as the number of filters, kernel sizes, learning rates, and dropout rates were varied within predefined ranges based on insights from existing literature (e.g., Su et al., 2018; Fu et al., 2023). This systematic tuning ensured an optimized balance between model accuracy and computational efficiency.

4 Design Specification

This proposed model can detect adversarial network behaviors and learning the pattern of benign and malicious in the IoT environments, thus, the design of the hybrid model is the major objective of this research. The model architecture used, data processing pipelines, and the integration of all the relevant components are thoroughly specified in this section as this design is designed to address the challenges the IoT intrusion detection system presents such as very high dimensionality of data, class imbalance, and need for real time processing.

4.1 Architectural Overview

The Conv-GAN model consists of two primary components i.e. the Convolutional Neural Network (CNN) for feature extraction, and the Generative Adversarial Network (GAN) for synthetic data generation to reduce the high class imbalance of rare malicious attacks. Following are the various architectural steps to implement the proposed hybrid Conv- GAN.

4.1.1 Convolutional Neural Network Design

The first one of the two components that make up the hybrid Conv-GAN architecture is the CNN, which is a network designed to process input tensors of a particular shape and configuration. The input tensors are of the form (batch size, sequence length, num features), where sequence length is the number of features after preprocessing and feature selection, while num features is 1. Architectural design of each layer in CNN model shown in Table 1. For compatibility with the CNN's expected input dimensions, features are normalized first to scale them to the same range before reshaping the features into a tensor structure acceptable by the CNN. This preprocessing is crucial to maintaining the CNN's consistency and achieving optimal performance.

Table 1: CNN Layers Design

Layer Type	Details	Purpose
Convolutional Layer	First Conv1D: Filters: 64, Kernel Size: 3, Activation: ReLU, Batch Normalization applied.	Captures local patterns and extracts low-level features.
Pooling	MaxPooling1D: Pool Size: 2	Reduces spatial dimensions and focuses on dominant features.
Regularization	Dropout: Rate: 0.4	Mitigates overfitting by randomly disabling neurons.
Convolutional Layer	Second Conv1D: Filters: 128, Kernel Size: 3, Activation: ReLU, Batch Normalization applied.	Learn higher-level abstractions and complex patterns.
Pooling	MaxPooling1D	Further reduces dimensionality.
Regularization	Dropout: Rate: 0.5	Mitigates overfitting by randomly disabling neurons.
Transition Layer	Flatten	Transforms multidimensional output to one-dimensional vector for dense layers.
Dense Layer	Units: 128, Activation: ReLU	Integrates features learned from previous layers.
Regularization	Dropout: Rate: 0.6	Further mitigates overfitting by randomly disabling neurons.
Output Layer	Units: 12 (Number of classes), Activation: Softmax	Provides probability distributions over classes for multi-class classification.

4.1.2 Generative Adversarial Network (GAN) Design

The second component of the proposed hybrid Conv-GAN is the GAN component which is designed to generate synthetic data samples for low representative classes to balance the dataset. Architectural design of each layers in CNN model shown in Table 2.

Table 2: GAN Layers Design

Network	Input	Architecture	Output Layer	Purpose
Generator	Noise Vector: Latent dimension (100) Class Labels: One-hot encoded.	Dense: 128, 256 units (ReLU).	Units: Equal to features Activation: Tanh.	Generates synthetic feature vectors.
Discriminator	Data Samples: Real/Synthetic vectors Class Labels: One-hot encoded.	Dense: 256, 128 units (ReLU).	Units: 1 Activation: Sigmoid.	Classifies inputs as real or fake.

The use of CNN and GAN combined requires applying the GAN to generate additional samples of training data sets for CNN especially when there is a condition of class imbalance. In this process, the GAN is used to create new synthetic samples for each of the minority class to bring their count to the level of sample size in the majority class. These synthetic samples are combined with the training dataset to form the new augmented training data set. The training sequence starts with the training of the GAN to only provide high quality synthetic data, meaning that the generated samples should be realistic and should cover different input space regions. Afterwards, the CNN is trained once more from the GAN enriched dataset that has a better generalization performance for all the classes.

4.2 Data Flow and Processing Pipeline

The data processing pipeline means that data is turned and prepared correctly for the model components by following processes. First, the data is inputted into the system, this forms the basis on which other processes will be performed on the data. Subsequently, the records are addressed to ensure that there is no missing data and all the data is clean to avoid any skew in the results. Outlier treatment is then done by applying techniques such as IQR and Z-score to eliminate bias that comes with outliers. Feature scaling is done using the standard scaler because the range of values in each feature can vary widely and they need to be standardized for compatibility. Finally the feature encoding is done through a method called label encoding to ensure that the model is in a position to understand the data provided in the categorical form appropriately. The data splitting strategy is shown in Table 3:

Table 3: Data Splitting and preprocessing

Step	Details	Purpose
Initial Split	Training Set: 70%, Test Set: 30%	Separates data for model training and testing.
Resampling	Apply SMOTE on the training set.	Handles class imbalance before GAN use.
Final Split	Training: 80% of resampled data, Validation: 20% of resampled data.	Prepares training and validation subsets.
GAN Augmentation	Generate synthetic samples for minority classes, combined with resampled data.	Balances training set with synthetic data.

To optimize the model, a sequence of model training and evaluation is followed as a procedure in this interface implementation. First when training GANs the discriminator and generator are trained in turns; there should be a close check on the generator loss to be sure that it has reached the convergence point. After this, the Convolutional Neural Network (CNN) is trained for the augmented dataset and includes early stopping para- meters in terms of validation loss to avoid overfitting. After the training session, the model evaluation is performed on the CNN with an aim of testing the trained network on the test set. Measurements for assessment are derived, and contingency tables are produced to give more clarity about the model's prediction and prediction errors. SMOTE used to handle class imbalance, ensuring that training is well-balanced for better generalization.

5 Implementation

The implementation of the hybrid model consisted of setting up the computational environment, data preprocessing, model development, training, and evaluation such that it outlines the practical steps taken in highlighting the technical difficulties and various integration challenges.

5.1 Environmental Setup

The development and execution of the models were conducted using Python 3.8 for scientific computing and machine learning such that the computational environment was established on a workstation with following hardware and software which mention in Table 4:

Table 4: Environmental setup and Libraries required to execute code

Component	Details
Processor	Intel Core i7-10700K CPU @ 3.80GHz
Memory	32 GB DDR4 RAM
GPU	NVIDIA GeForce RTX 3080 with 10 GB GDDR6X VRAM
Operating System	Ubuntu 20.04 LTS

GPU Utilization	Accelerated training of computationally intensive models (e.g., CNN and GAN).
Key Libraries/Frameworks	
- TensorFlow 2.5	For building and training deep learning models.
- Keras API	High-level neural network interface.
- Scikit-learn	Data preprocessing, feature scaling, encoding, and evaluation metrics.
- Pandas and NumPy	Data manipulation and numerical computations.
- Matplotlib and Seaborn	Data visualization during exploratory data analysis.
- Imbalanced-learn	Handling class imbalance (e.g., SMOTE).
- Jupyter Notebook	Enabled interactive coding and iterative experimentation.

5.2 Data Processing

The RT-IoT 2022 dataset was loaded using Pandas, and an initial exploration, along with Exploratory Data Analysis (EDA), was conducted to better understand and visualize the data structure and its content. During the data preprocessing phase, we determined there were no missing values, nor duplicate rows in the dataset. For that, categorical features like 'proto' and 'service' were converted into numerical representations. The dataset was finally divided into training, validation and test sets with a 70:15:15 split to allow for a good model development and evaluation.

5.3 Model Development

5.3.1 Data Balancing with SMOTE

After the first train-test split, Synthetic Minority Over-sampling Technique was applied to the training set to handle class imbalance. SMOTE, the synthetic generation to complete under represented classes, by interpolation of the known minority class instances. This approach helps to achieve spread of attacks in the training set, which is an important factor for improving the ability of the model to discriminate between abnormal attacks. Through minority class supplementation, SMOTE ensures that the model has enough exposure to these rare but important attack patterns, which leads to more robust training and better generalization. Figure 4 shows the performance of SMOTE in balancing the data set by exhibiting the class distribution of the training data before and after the application of the SMOTE.

```
Class distribution after applying SMOTE and splitting:
Training set class distribution:
Attack_type
NMAP_XMAS_TREE_SCAN           53009
NMAP_UDP_SCAN                 53009
DOS_SYN_Hping                 53009
ARP_poisioning                53009
Thing_Speak                   53009
NMAP_FIN_SCAN                 53009
Metasploit_Brute_Force_SSH    53009
Wipro_bulb                    53009
DDOS_Slowloris                53009
NMAP_OS_DETECTION             53008
NMAP_TCP_scan                 53008
MQTT_Publish                  53008
Name: count, dtype: int64

Validation set class distribution:
Attack_type
NMAP_OS_DETECTION             13253
MQTT_Publish                  13253
NMAP_TCP_scan                 13253
Thing_Speak                   13252
Wipro_bulb                    13252
NMAP_FIN_SCAN                 13252
NMAP_UDP_SCAN                 13252
NMAP_XMAS_TREE_SCAN           13252
DDOS_Slowloris                13252
Metasploit_Brute_Force_SSH    13252
ARP_poisioning                13252
DOS_SYN_Hping                 13252
Name: count, dtype: int64
```

Figure 4: Dataset Balance after SMOTE

5.3.2 Convolutional Neural Network

The CNN architecture was implemented using TensorFlow and Keras and the model consists of the following layers shown in Table 5:

Table 5: CNN Implementation Code Design

Layer Type	Details	Purpose
Input Layer	Reshaped input data with a singleton dimension.	Ensures compatibility with Conv1D layers.
Convolutional Layers	- Conv1D: 64 filters, Batch Normalization, MaxPooling, Dropout. - Conv1D: 128 filters, Batch Normalization, MaxPooling, Dropout.	Extracts features, prevents overfitting, and reduces spatial dimensions.
Flatten Layer	Converts 3D convolutional output to a 1D vector.	Prepares data for fully connected layers.
Dense Layer	Fully connected layer with 128 neurons (ReLU activation), followed by a Dropout layer.	Learn high-level features.
Output Layer	Dense layer with 12 units (softmax activation).	Outputs probability distribution for classes.

The model was compiled with the following configuration:

- Optimizer: Adam optimizer with a learning rate of 0.001.
- Loss Function: SparseCategoricalCrossentropy, as it is intended for integer-encoded labels.
- Performance Metric: Accuracy was utilized to evaluate and monitor the performance of the model during training.

The training was carried out over five epochs with a batch size of 32. Early stopping was introduced to minimize overfitting and enhance the model's generalizability. The model achieved high accuracy rates on the validation set, demonstrating that it was effectively learning from the balanced dataset.

5.3.3 Generative Adversarial Network (GAN)

The GAN was developed to generate synthetic samples for the minority classes, addressing the class imbalance for rare malicious attacks. The GAN training process included the following steps:

- Adversarial Training Loop:
  - Discriminator Training: Alternated between training on real samples with true labels and synthetic samples with false labels.
  - Generator Training: Trained to produce synthetic samples that could deceive the discriminator.
- Loss Function: Binary cross-entropy was used for both the generator and discriminator.
- Optimizer: Both networks used the Adam optimizer with a learning rate of 0.0002 and a beta 1 of 0.5.

Due to the complexity of GAN training and resource constraints, the number of training epochs was limited. However, despite these limitations, the generator produced synthetic samples that resembled the distribution of the minority classes.

5.3.4 Integration of CNN and GAN (Conv-GAN)

The GAN synthetic data was then augmented with the original training data in order to create an augmented dataset with balanced class distributions. After using the same architecture and training parameters, as before, the CNN was retrained on this augmented dataset. Balanced class distributions. The CNN was then retrained on this augmented dataset, following the same architecture and training parameters as before. But integration was not easy. It was found that when tested on the test set, the Conv-GAN model performed worse than the standalone CNN. This decrease in performance indicated that the quality of the synthetic data or the training process could be an issue that needed to be investigated further.

6 Evaluation

The evaluation phase is critical in assessing the effectiveness of the Conv-GAN hybrid model in detecting adversarial network behaviors within IoT environments. This section presents a

comprehensive analysis of the model's performance, comparing it against the standalone Convolutional Neural Network (CNN) model. Various metrics are employed to provide a detailed understanding of the strengths and limitations of the proposed approach. The standalone CNN model demonstrated robust performance on the test set. The key results are summarized as follows:

- Accuracy: The model achieved an overall accuracy of 99.30%, indicating that it correctly classified the vast majority of instances.
- Precision and Recall: High precision and recall values were observed across most classes, particularly for the majority class DOS SYN Hping.

Following is the Table 6 shown results for the Classification Report for CNN Model:

Table 6: CNN Result

Attack Type	Precision	Recall	F1-Score	Support
ARP_poisioning	0.986	0.9389	0.9619	2,325
DDOS_Slowloris	0.75	0.9938	0.8548	160
DOS_SYN_Hping	1	1	1	28,398
MQTT_Publish	1	0.9976	0.9988	1,244
Metasploit_Brute_Force_SSH	0.0935	0.9091	0.1695	11
NMAP_FIN_SCAN	0.875	0.875	0.875	8
NMAP_OS_DETECTION	1	1	1	600
NMAP_TCP_scan	0.9934	1	0.9967	301
NMAP_UDP_SCAN	1	0.9305	0.964	777
NMAP_XMAS_TREE_SCAN	1	0.9967	0.9983	603
Thing_Speak	0.9774	0.9782	0.9778	2,433
Wipro_bulb	0.7979	0.9868	0.8824	76

The CNN model as shown in Table 6 exceptional performance, particularly for the majority class and several minority classes. Notable observations include:

- High Precision and Recall: For classes like DOS SYN Hping, NMAP OS DETECTION, and MQTT Publish, the model achieved perfect or near perfect precision and recall.
- Minority Classes: Despite the class imbalance, the model performed reasonably well on some minority classes. For instance, NMAP FIN SCAN had an F1-score of 0.8750.

- Challenges with Rare Classes: The Metasploit Brute Force SSH class had a low precision but high recall, indicating that the model correctly identified most instances but also had a high false positive rate for this class.

The following is the line chart graph of the performance of the CNN training and validation accuracy and loss shown in Figure 5.

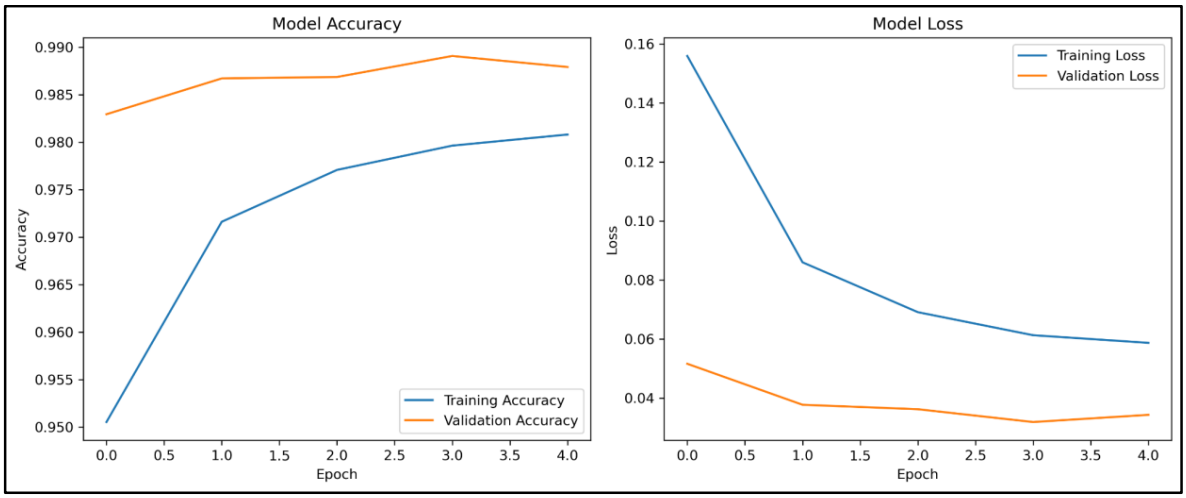

Figure 5: CNN model accuracy and model loss as line chart

Comparison of the results presented in the Table shows that the proposed approach achieves remarkably different detection rates while addressing different types of cybersecurity attacks. DOS SYN Hping with 28,398 samples has the highest support but no precision, recall, and F1-Score to detect anything, and ARP Poisoning has 2,325 samples and an F1-Score of only 0.0115. For example, Metasploit Brute Force SSH has an R-Recall of 0.3636 but has very low P-Recall 0.0006 and F1-score of 0.0011, revealing high false positives. The highest F1-Score overall is given by DDOS Slowloris: 0.3186, precision: 0.5455 and recall: 0.225 over 160 samples. Moderate performance is observed for Thing Speak F1-Score: 0. The two identified attacks are NMAP TCP Scan (Precision: 0.7586, recall: 0.4529) and NMAP UDP Scan (F1-Score: 0.2047, recall: 0.3115). However, some of them including MQTT Publish, NMAP TCP Scan and NMAP Xmas Tree Scan are shown to have zero performance in all three parameters, thus, meaning that they are entirely ineffective in detection. When analyzing the results, it is possible to identify which of the studied attacks are easier for detection: DDOS Slowloris and Thing Speak were detected most often, DOS SYN Hping, as well as numerous attacks associated with the use of NMAP, reveal the most significant vulnerabilities which show in Table 7 shows results for the Classification Report for Conv-GAN Model:

Table 7: Conv-GAN Result

Attack Type	Precision	Recall	F1-Score	Support
ARP_poisioning	0.0067	0.0404	0.0115	2,325
DDOS_Slowloris	0.5455	0.225	0.3186	160
DOS_SYN_Hping	0	0	0	28,398
MQTT_Publish	0	0	0	1,244

Metasploit_Brute_Force_SSH	0.0006	0.3636	0.0011	11
NMAP_FIN_SCAN	0	0	0	8
NMAP_OS_DETECTION	0	0	0	600
NMAP_TCP_scan	0	0	0	301
NMAP_UDP_SCAN	0.1525	0.3115	0.2047	777
NMAP_XMAS_TREE_SCAN	0	0	0	603
Thing_Speak	0.0784	0.4529	0.1337	2,433
Wipro_bulb	0.3333	0.0132	0.0253	76

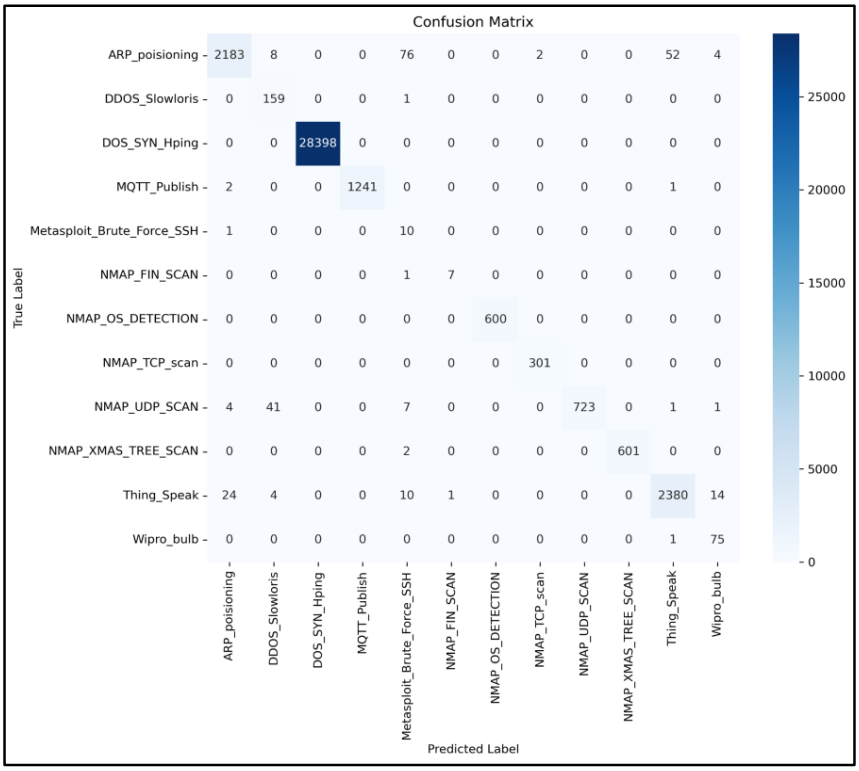

Figure 6: CNN confusion matrix and performance of the dataset

The above Figure 6 was for the confusion matrix of the baseline CNN and now the following Figure 7 depicts the confusion matrix of the Conv-GAN:

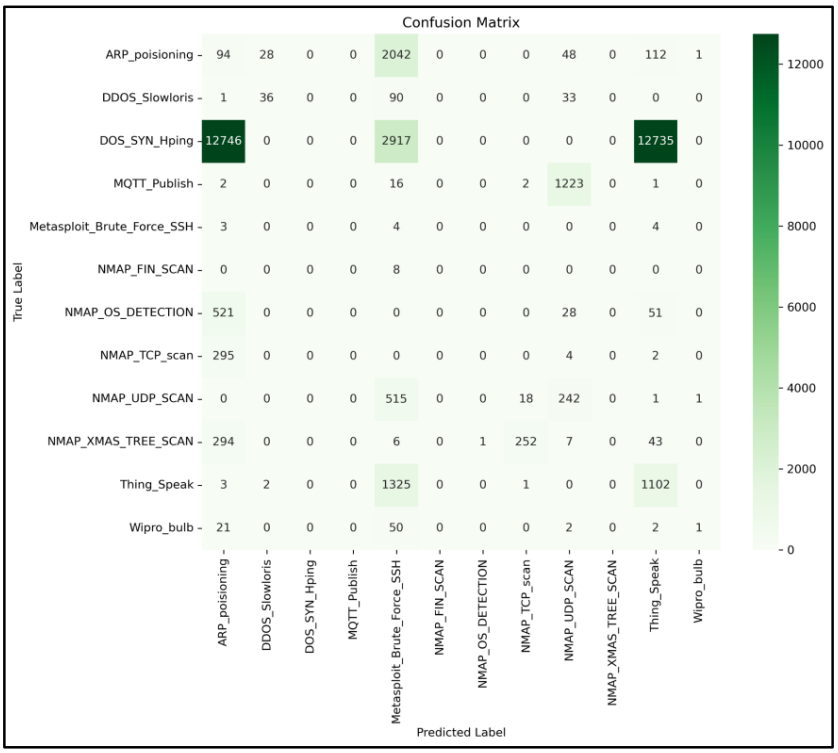

Figure 7: Proposed hybrid model Conv-GAN's confusion matrix

The following is the line chart graph of the performance of the Conv-GAN's training and validation accuracy and loss shown in Figure 8:

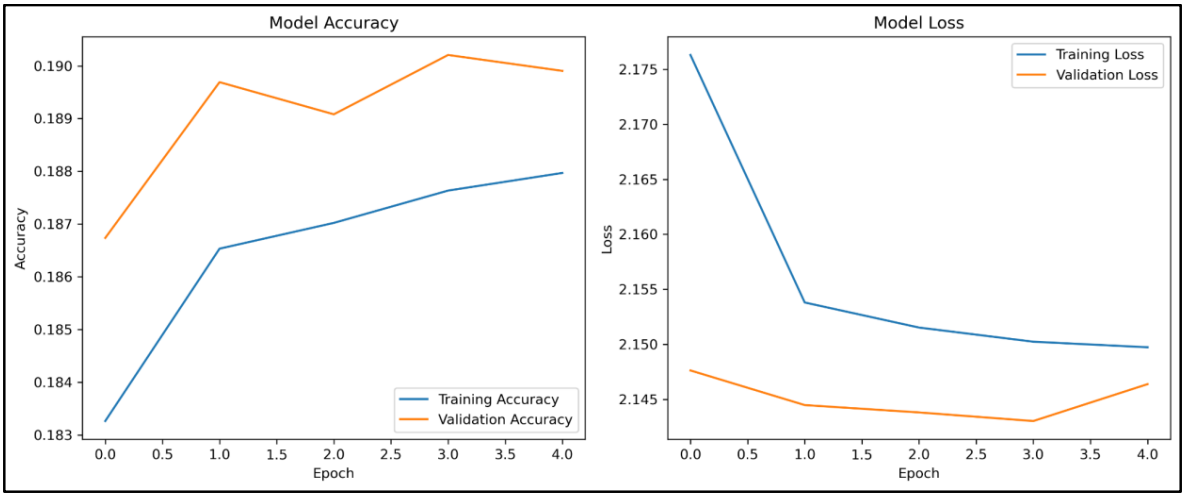

Figure 8: Proposed hybrid model Conv-GAN's confusion matrix

The Conv-GAN model's poor performance can be attributed to several factors:

- Synthetic Data Quality: The GAN may not have generated high-fidelity synthetic samples that accurately represent the minority classes. Poor quality synthetic data can introduce noise and misleading patterns.
- Model Overfitting or Underfitting: The CNN retrained on the augmented dataset may have overfit to the synthetic data patterns or failed to generalize to real data.

- Training Instability: GANs are challenging to train, often requiring careful tuning of hyperparameters and a significant number of epochs to converge. Insufficient training could lead to inadequate generator and discriminator performance.

7 Conclusion and Future Work

This research for the proliferation of the IoT devices considers the necessity for robust and efficient intrusion detection systems to identify and counteract adversarial network behaviors. The goal of this work was to develop a hybrid Conv-GAN model that combines the Convolutional Neural Networks with Generative Adversarial Networks to enhance intrusion detection in IoT. Finally, the study finds that the standalone CNN model performed well and strongly on most attack types, indicating that the CNN model extracted features that were relevant and could detect intrusion in the IoT network traffic. Additionally, GAN generated synthetic data integration into the training process was challenging, and it also degraded model performance. The results also showed that the Conv-GAN model was not able to use and generate the synthetic data as well as the CNN model.

Further research in this area must investigate how Conditional GANs and Wasserstein GANs could improve synthetic data generation quality and model execution. Conditional GANs produce structured data synthesis through the integration of class labeling inputs that enables improved minority class accuracy. Success rates from training Wasserstein GANs show improved stability alongside distribution modeling capabilities that could solve the synthetic data quality problems found in the present Conv-GAN model. The implementation of superior GAN architectures produces synthetic samples with superior fidelity leading to enhanced intrusion detection system reliability. The research outcomes show CNNs demonstrate potential as intrusion detection systems in IoT environments but Generative Adversarial Networks need careful application. The performance quality of GAN-generated synthetic data directly affects the entire system because weak GAN model training creates noisy output which degrades system performance. Future research must address these problems through advanced GAN architectures including conditional GANs and Wasserstein GANs as well as training stability techniques for synthetics quality enhancement. The development of solid assessment techniques for synthetic data quality represents a critical need because these methodologies must demonstrate that generation methods effectively maintain the underlying data characteristics without creating exceptional biases. On the last, I would have to conduct extensive hyper parameter tuning of both the GAN and CNN parts to get best results. Applications of such synthetic data in future work might use advanced architectures such as Wasserstein GANs or conditional GANs to produce higher quality synthetic data and robust methods to evaluate the quality and fidelity of the generated data. Moreover, hybrid models are optimized for real time intrusion detection in IoT networks that are large, dynamic and require large scale solutions that are scalable and efficient in terms of resources. Future research can then address aspects of these properties to build more reliable and efficient intrusion detection systems appropriate in the context of the IoT environment.

References

Alabsi, B., Anbar, M. and Rihan, S. (2023). Conditional tabular generative adversarial based intrusion detection system for detecting ddos and dos attacks on the internet of things networks, *Sensors* **23**(12): 5644.

Almarshdi, R., Nassef, L., Fadel, E. and Alowidi, N. (2023). Hybrid deep learning based attack detection for imbalanced data classification, *Intelligent Automation & Soft Com- puting* **35**(1): 297–320.

Alsoufi, M., Razak, S., Siraj, M., Nafea, I., Ghaleb, F., Saeed, F. and Nasser, M. (2021). Anomaly-based intrusion detection systems in iot using deep learning: a systematic literature review, *Applied Sciences* **11**(18): 8383.

Aslan, , Ozkan-Okay, M. and Gupta, D. (2021). Intelligent behavior-based malware detection system on cloud computing environment, *IEEE Access* **9**: 83252–83271.

Bao, Z., Lin, Y., Zhang, S., Li, Z. and Mao, S. (2021). Threat of adversarial attacks on dl-based iot device identification, *IEEE Internet of Things Journal* **9**(11): 9012– 9024.

Benaddi, H., Jouhari, M., Ibrahimi, K., Othman, J. B. and Amhoud, E. (2022). Anomaly detection in industrial iot using distributional reinforcement learning and generative adversarial networks, *Sensors* **22**(21): 8085.

Cheng, A. (2019). Pac-gan: packet generation of network traffic using generative ad- versarial networks, *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*.

Dutt, I., Borah, S. and Maitra, I. (2020). Immune system-based intrusion detection system (is-ids): A proposed model, *IEEE Access* **8**: 34929–34941.

Ferdowsi, A. and Saad, W. (2019). Generative adversarial networks for distributed intru- sion detection in the internet of things, *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6.

Fu, T., Sun, B. and Zhang, C. (2023). A deep learning model for accurate and robust internet traffic classification, *Applied and Computational Engineering* **6**(1): 725–730.

Hnamte, V. and Hussain, J. (2023). Dcnnbilstm: An efficient hybrid deep learning-based intrusion detection system, *Telematics and Informatics Reports* **10**: 100053.

Huang, Y., Wang, W., Wang, H., Jiang, T. and Zhang, Q. (2020). Authenticating on- body iot devices: An adversarial learning approach, *IEEE Transactions on Wireless Communications* **19**(8): 5234–5245.

Kwon, H., Kim, T. and Lee, M. (2022). Advanced intrusion detection combining signature-based and behavior-based detection methods, *Electronics* **11**(6): 867.

Lansky, J., Ali, S., Mohammadi, M., Majeed, M., Karim, S., Rashidi, S., Hosseinzadeh,

M. and Rahmani, A. (2021). Deep learning-based intrusion detection systems: A systematic review, *IEEE Access* **9**: 101574–101599.

Ling, M. (2023). Machine-learning-based network sparsification modeling for iot security analysis, *Proceedings of SPIE* **188**.

Liu, G. and Zhang, J. (2020). Cnid: Research of network intrusion detection based on convolutional neural network, *Discrete Dynamics in Nature and Society* **2020**: 1–11.

Liu, Q., Hagenmeyer, V. and Keller, H. (2021). A review of rule learning-based intrusion detection systems and their prospects in smart grids, *IEEE Access* **9**: 57542–57564.

Mazhar, T., Talpur, D., Shloul, T., Ghadi, Y., Haq, I., Ullah, I. and Hamam, H. (2023). Analysis of iot security challenges and its solutions using artificial intelligence, *Brain Sciences* **13**(4): 683.

Nie, L., Wu, Y., Wang, X., Guo, L., Wang, G., Gao, X. and Li, S. (2021). Intrusion detection for secure social internet of things based on collaborative edge computing: A generative adversarial network-based approach, *IEEE Transactions on Computational Social Systems* **9**(1): 134–145.

Odeh, A. (2023). Ensemble-based deep learning models for enhancing iot intrusion de- tection, *Applied Sciences* **13**(21): 11985.

Sayed, N., Shoaib, M., Ahmed, W., Qasem, S., Albarrak, A. and Saeed, F. (2023). Augmenting iot intrusion detection system performance using deep neural networks, *Computers Materials & Continua* **74**(1): 1351–1374.

Su, J., Vargas, D., Prasad, S., Sgandurra, D., Feng, Y. and Sakurai, K. (2018). Light- weight classification of iot malware based on image recognition, *2018 IEEE 42nd An- nual Computer Software and Applications Conference (COMPSAC)*, pp. 664–669.

Ullah, I. and Mahmoud, Q. (2021). A framework for anomaly detection in iot networks using conditional generative adversarial networks, *IEEE Access* **9**: 165907–165931.

UCI Machine Learning Repository. (2022). RT-IoT 2022 Dataset [Data set]. Retrieved from https://archive.ics.uci.edu/dataset/942/rt-iot2022

Wang, Z., Huang, H., Du, R., Xing, L. and Yuan, G. (2023). Iot intrusion detection model based on cnn-gru, *Frontiers in Computing and Intelligent Systems* **4**(2): 90–95.

Wu, Y., Nie, L., Wang, S., Ning, Z. and Li, S. (2021). Intelligent intrusion detection for internet of things security: A deep convolutional generative adversarial network- enabled approach, *IEEE Internet of Things Journal* **10**(4): 3094–3106.

Xiao, Y., Cheng, X., Zhang, T. and Zhong-kai, Z. (2019). An intrusion detection model based on feature reduction and convolutional neural networks, *IEEE Access* **7**: 42210– 42219.

Yaras, S. and Dener, M. (2024). Iot-based intrusion detection system using new hybrid deep learning algorithm, *Electronis* **13**(6): 10