



Thesis-Thesis Defense - EF235401

GAN-LSTM for Feature Enhancement and Data Generation on Imbalance Dataset in IoT Malware Analysis

GREGORIUS EDO SATRIATAMA EKA SETIAWAN
NRP 6025231050

SUPERVISOR
Prof. Tohari Ahmad, S.Kom., M.IT., Ph.D.

MASTER PROGRAM
NET CENTRIC COMPUTING AREA OF EXPERTISE
MASTER OF INFORMATICS ENGINEERING STUDY PROGRAM
INFORMATICS ENGINEERING DEPARTMENT
FACULTY OF INTELLIGENT ELECTRICAL AND INFORMATICS TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2025



Thesis-Thesis Defense - EF235401

GAN-LSTM for Feature Enhancement and Data Generation on Imbalance Dataset in IoT Malware Analysis

GREGORIUS EDO SATRIATAMA EKA SETIAWAN
NRP 6025231050

SUPERVISOR
Prof. Tohari Ahmad, S.Kom., M.IT., Ph.D.

MASTER PROGRAM
NET CENTRIC COMPUTING AREA OF EXPERTISE
MASTER OF INFORMATICS ENGINEERING STUDY PROGRAM
INFORMATICS ENGINEERING DEPARTMENT
FACULTY OF INTELLIGENT ELECTRICAL AND INFORMATICS TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2025

[Page intentionally left blank]

THESIS APPROVAL PAGE

The thesis is prepared to meet one of the requirements for obtaining a degree
Master of Science in Computer (M.Kom.)

On

Institut Teknologi Sepuluh Nopember

By:

Gregorius Edo Satriatama Eka Setiawan

NRP. 6025231050

Date of Thesis Defense: 16-07-2025

Graduation Period: 09 2025

Approved By:

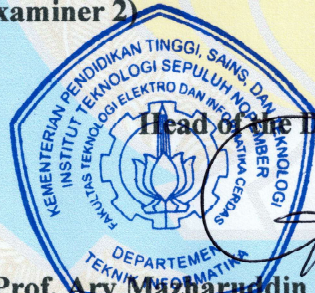
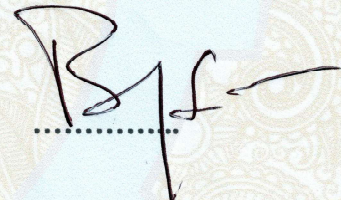
1. Prof. Tohari Ahmad, S.Kom., M.IT., Ph.D.
NIP. 197505252003121002
(Supervisor)



3. Hudan Studiawan, S.Kom., M.Kom., Ph.D.
NIP. 198705112012121003
(Examiner 1)



4. Bagus Jati Santoso, S.Kom, Ph.D
NIP. 198611252018031001
(Examiner 2)



Head of the Department of Informatics Engineering

Prof. Arif Maizharuddin Shiddiqi, S.Kom., M.Comp.Sc., Ph.D., IPM.

NIP. 19810620 200501 1 003

[Page intentionally left blank]

STATEMENT OF ORIGINALITY

The undersigned below,

Student Name (NRP)	: Gregorius Edo Satriatama Eka Setiawan
Supervisor (NIP)	: Prof. Tohari Ahmad, S.Kom., M.IT., Ph.D.
Study Program	: Master of Informatics Engineering
Department	: Informatics Engineering Department
Faculty	: Faculty of Intelligent Electrical and Informatics
Technology	

Hereby declare that the thesis entitled **“GAN-LSTM for Feature Enhancement and Data Generation on Imbalance Dataset in IoT Malware Analysis”** is my own work, is original, and written following the rules of scientific writing.

If in the future any discrepancy with this statement is found, then I am willing to accept sanctions in accordance with the provisions in force at the Institut Teknologi Sepuluh Nopember.

Surabaya, 28th July 2025



Gregorius Edo Satriatama Eka S.
NRP. 6025231050

Acknowledged,
Supervisor,

Prof. Tohari Ahmad, S.Kom., M.IT., Ph.D.
NIP. 197505252003121002

[Page intentionally left blank]

PREFACE

Praise be to God Almighty who has given His grace and gifts, so that the author can complete the Thesis and compile Thesis report with the title “GAN-LSTM FOR FEATURE ENHANCEMENT AND DATA GENERATION ON IMBALANCE DATASET IN IOT MALWARE ANALYSIS”. This report was made to complement the Thesis Course which is one of the graduation requirements for Informatics Engineering Department Institut Teknologi Sepuluh Nopember master’s students.

The author realizes that without the help and support of various parties, this thesis report could not be completed properly. Therefore, the author would like to thank those who have provided guidance, support, and opportunities to the author until this thesis report can be completed, including:

1. Prof. Tohari Ahmad, S.Kom., M.IT., Ph.D. as a Thesis supervisor who always provides useful knowledge and tireless guidance until the writing of this report is completed.
2. Dr. Eng Darlis Herumurti, S.Kom, M.Kom. as an academic advisor to the writer who always provides guidance on the author’s courses.
3. RD. Ignatius Sadewo Setiabudi as head of Campus Ministry who provides guidance, enthusiasm and material support to the writer.
4. The family, especially the author’s parents and younger siblings who always provide support, attention, and affection during lectures and Thesis work.
5. Angelica Yuliany as writer’s lover who always accompanies and provides moral support, attention, and affection during the writing of this Thesis.
6. Friends of the Informatics Engineering master’s study program who always accompany and provide many memories and support.
7. All lecturers and employees of the Informatics Engineering Department F-ELECTICS, Institut Teknologi Sepuluh Nopember, who provide knowledge and time to prepare the author to enter the world of work.
8. And all parties who cannot be mentioned one by one, but have provided support during the writing of this Thesis.

The author realizes that in the making of this thesis there are still many shortcomings. Therefore, the author expects constructive criticism and suggestions

from readers for mutual improvement and progress. The author hopes that this Thesis Report can be useful and contribute to the development of science and technology, especially in the field of network security.

“Whatever you have learned or received or heard from me, or seen in me—put it into practice. And the God of peace will be with you.”

~Philipians 4:9~

Surabaya, 30th June 2025

Gregorius Edo

GAN-LSTM UNTUK PENINGKATAN FITUR DAN PEMBUATAN DATA PADA DATASET TAK-IMBANG DALAM ANALISIS MALWARE IOT

Nama : Gregorius Edo Satriatama Eka Setiawan
NRP : 6025231050
Pembimbing : Prof. Tohari Ahmad, S.Kom., M.IT., Ph.D.

ABSTRAK

Perluasan layanan internet dan penggunaan Internet of Things (IoT), yang didorong oleh kemajuan jaringan seluler 5G, telah mengubah berbagai sektor, seperti komunikasi, data sharing, dan e-commerce. Pertumbuhan ini meningkatkan risiko keamanan siber mengingat aliran data yang besar dan sifat otonom sistem IoT yang membuat sistem tersebut rentan terhadap ancaman di dunia maya. Pada tahun 2024, lebih dari 1,2 miliar malware dan aplikasi yang berpotensi tidak diinginkan (PUA) terdeteksi, yang mencerminkan lonjakan ancaman di lapisan aplikasi. Meskipun metode enkripsi seperti Transport Layer Security (TLS) melindungi data privacy, metode enkripsi juga meningkatkan kompleksitas dalam mengidentifikasi lalu lintas berbahaya, meningkatkan kebutuhan akan mekanisme deteksi lanjutan. Selain bahaya tersebut, ketidakseimbangan data di dalam dataset juga menjadi perhatian karena data yang tidakimbang akan menekan kinerja deteksi malware, sehingga masalah tersebut harus diselesaikan. Penelitian ini memperkenalkan model FE-CGAN-LSTM yang dapat mengatasi ketidakseimbangan data dan meningkatkan representasi fitur secara efektif. Dataset asli menghasilkan hasil moderat pada deteksi malware dengan akurasi sekitar 90,7–91,0% dan skor F1 yang lebih rendah. Menggunakan dataset yang telah ditingkatkan fiturnya oleh FE-CGAN-LSTM, terdapat peningkatan performa, dimana semua model klasifikasi mencapai skor 99,99% pada metrik akurasi, presisi, recall, dan skor F1.

Kata Kunci: Dataset takimbang, Generative Adversarial Networks, IoT-23, Long-Short Term Memory, Malware, Pembuatan data sintetis, Peningkatan fitur

[Page intentionally left blank]

GAN-LSTM FOR FEATURE ENHANCEMENT AND DATA GENERATION ON IMBALANCE DATASET IN IOT MALWARE ANALYSIS

Name : Gregorius Edo Satriatama Eka Setiawan
Student Identity Number : 6025231050
Supervisor : Prof. Tohari Ahmad, S.Kom., M.IT., Ph.D.

ABSTRACT

The expansion of internet services and the use of Internet of Things (IoT), which are driven by advances in 5G cellular networks, have transformed multiple sectors, that include communication, data sharing, and e-commerce. This growth amplifies cybersecurity risks given the massive data flow and the autonomous nature of IoT systems, making them susceptible to cyber threats. In 2024, over 1.2 billion instances of malware and potentially unwanted applications (PUAs) were detected. These conditions reflects a surge in threats at the application layer. Although encryption methods like Transport Layer Security (TLS) safeguard data privacy, they also increase complexity in identifying malicious traffic, raising the demand for advanced detection mechanisms. In addition to these dangers, data imbalance is also a concern because imbalanced data will hinder the performance of malware detection, so it must be resolved immediately. This research introduces the FE-CGAN-LSTM model that can address data imbalance and improve feature representation in effective way. The original dataset yields moderate results with accuracies around 90.7–91.0% and lower F1 scores in malware identification. Using the dataset improved by FE-CGAN-LSTM, performances increase is observed, with all classifiers achieving scores of 99.99% accuracy, precision, recall, and F1-score.

Keywords: Data generation, Feature enhancement, Generative adversarial networks, Imbalance dataset, IoT-23, Long-short term memory, Malware

[Page intentionally left blank]

TABLE OF CONTENTS

TITLE PAGE	i
THESIS APPROVAL PAGE.....	iii
STATEMENT OF ORIGINALITY	v
PREFACE	vii
ABSTRAK	ix
ABSTRACT.....	xi
TABLE OF CONTENTS.....	xiii
LIST OF TABLES	xvii
LIST OF FIGURES	xix
CHAPTER 1 INTRODUCTION	1
1.1. Background	1
1.2. Research Questions	2
1.3. Research Objective.....	2
1.4. Research Benefits.....	3
1.5. Research Contributions	3
1.6. Research Scope	3
CHAPTER 2 THEORETICAL BASIS.....	5
2.1. Related Works.....	5
2.1.1. Support Vector Machines.....	5
2.1.2. Synthetic Minority Oversampling Technique.....	6
2.1.3. Generative Adversarial Networks for Malware Detection.....	7
2.1.4. Temporal Pattern Learning on Generative Models.....	8
2.1.5. Improving Tabular Data Diversity using GANs	9
2.1.6. Advancing Classification Performance Through Synthetic Feature Augmentation.....	10
2.2. Theoretical Basis.....	11
2.2.1. Internet of Things (IoT)	11
2.2.2. Malware on IoT.....	12

2.2.3.	Generative Adversarial Networks	13
2.2.4.	Long Short-Term Memory	15
2.2.5.	Feature Enhancement	17
2.2.6.	Confusion Matrix	19
2.2.7.	Conditional GANs.....	20
CHAPTER 3	RESEARCH METHODOLOGY.....	23
3.1.	Dataset.....	24
3.2.	Data Preprocessing.....	26
3.2.1.	One-Hot Encoding	26
3.2.2.	MinMax Scaler.....	27
3.3.	Train Test Split.....	28
3.4.	Hybrid FE-GAN-LSTM.....	29
3.4.1.	Generator.....	30
3.4.2.	Discriminator.....	32
3.5.	Model Evaluation	34
3.5.1.	Visual Evaluation using Principal Component Analysis	34
3.5.2.	Silhouette Score	35
3.5.3.	Malware Identification	36
CHAPTER 4	RESULTS AND DISCUSSIONS	38
4.1.	Experiment Environment	38
4.2.	Dataset.....	38
4.2.1.	Dataset Features	39
4.2.2.	Class Label	42
4.3.	Data Preprocessing.....	44
4.4.	Performance Behavior on Original and Feature Enhanced Datasets	47
4.4.1.	Performance on Original Dataset	48
4.4.2.	Performance on FE Dataset.....	56

4.5.	Quality of The FE Dataset.....	65
4.5.1.	Visual Analysis using PCA.....	66
4.5.2.	Cluster Analysis using Silhouette Score	70
4.6.	Performance of FE and Non-FE Model	72
4.7.	Limitations	74
CHAPTER 5 CONCLUSIONS AND SUGGESTIONS.....		77
5.1.	Conclusion	77
5.1.1.	Implementation of CGAN-LSTM to Enhance Dataset and Improve Classifier Performance.....	77
5.1.2.	Quality of Generated Data and Model Performance Compared to Other Models	77
5.2.	Suggestion.....	77
REFERENCES.....		79
AUTHOR BIOGRAPHY		87

[Page intentionally left blank]

LIST OF TABLES

Table 4.1 Experiment environment.....	38
Table 4.2 Features Contained in IoT-23 Dataset (Zeghida et al., 2024).....	39
Table 4.3 Name and total of attacks inside IoT-23 dataset.....	42
Table 4.4 Original Dataset	45
Table 4.5 Dataset after Preprocessing.....	47
Table 4.6 Generated CGAN Features	48
Table 4.7 CNN Classification Report (Original Dataset)	49
Table 4.8 RNN Classification Report (Original Dataset)	51
Table 4.9 LSTM Classification Report (Original Dataset)	53
Table 4.10 CNN Classification Report (FE Dataset).....	56
Table 4.11 RNN Classification Report (FE Dataset).....	58
Table 4.12 LSTM Classification Report (FE Dataset).....	60
Table 4.13 Macro and weighted average metrics across all classifier models.....	64
Table 4.14 Silhouette score across all class	70
Table 4.15 Impact of FE on Weighted Average Performance Metrics in Multiclass Malware Detection.....	72

[Page intentionally left blank]

LIST OF FIGURES

Figure 2.1 General GANs Architecture	13
Figure 2.2 LSTM Cell (Azati et al., 2024).....	17
Figure 2.3 Feature Enhancement Model (Wei et al., 2023)	18
Figure 3.1 Research Workflow	23
Figure 3.2 IoT-23 dataset capture diagram (Garcia et al., 2020)	25
Figure 3.3 Model of the FE-GAN-LSTM.....	29
Figure 3.4 Input Layer of Generator	30
Figure 3.5 CGAN-LSTM Generator Architecture	31
Figure 3.6 CGAN-LSTM Discriminator Architecture.....	33
Figure 4.1 Confusion matrix of CNN model on Original Dataset	50
Figure 4.2 Confusion matrix of RNN model on Original Dataset	52
Figure 4.3 Confusion matrix of LSTM model on Original Dataset.....	53
Figure 4.4 Loss Curve of CNN model on Original Dataset.....	54
Figure 4.5 Loss Curve of RNN model on Original Dataset.....	55
Figure 4.6 Loss Curve of LSTM model on Original Dataset.....	55
Figure 4.7 Confusion matrix of CNN model on FE Dataset.....	57
Figure 4.8 Confusion matrix of RNN model on FE Dataset.....	59
Figure 4.9 Confusion matrix of LSTM model on FE dataset	61
Figure 4.10 Loss curve of CNN model on Enhanced Dataset	61
Figure 4.11 Loss curve of RNN model on Enhanced Dataset	62
Figure 4.12 Loss curve of LSTM model on Enhanced Dataset	63
Figure 4.13 PCA Visualization of Original Dataset	67
Figure 4.14 PCA Visualization of Enhanced Dataset	69

[Page intentionally left blank]

CHAPTER 1

INTRODUCTION

In this chapter, it is explained about several things that underlie the research, including the background, problem formulation, research objectives, research benefits, research contributions, and limitations of the problem in the research. This research focuses on improving malware detection in IoT systems amid the growing cybersecurity challenges posed by the expansion of internet services and 5G networks. The study addresses the increasing threat of malicious traffic, with over 1.2 billion malware and potentially unwanted applications identified in 2024 alone. The research aims to implement a feature enhancement technique using a GAN-LSTM model to improve classification performance on imbalanced datasets, specifically targeting the IoT-23 dataset with 11 malware classes. The objectives include enhancing the quality of generated data and improving the model's performance compared to existing approaches. The study's scope encompasses testing the proposed method on a personal computer with specified hardware configurations. By contributing to the advancement of feature enhancement methods and GAN-LSTM performance, this research seeks to strengthen malware detection capabilities in the evolving landscape of IoT security.

1.1. Background

The development of the use of internet services has caused it to become an integral part of daily life (Shahin et al., 2024). The use is growing in the personal and professional realm which includes the fields of communication, data exchange, and retail activities. Large amounts of data are transmitted through one network access point to another through various hardware, software, and protocols (Lim et al., 2024). The large amount of data transmitted causes variations in the data traffic pattern (Zang et al., 2024). The development of these types of data and variations has led to the development of cyberattacks that can attack various systems, one of which is IoT.

Today's rapidly evolving 5G cellular network technology is leading to a boom in the use of IoT technology (Torre et al., 2024). IoT is a rapidly developing field of technology that connects objects or things (Ullah & Mahmoud, 2021). IoT

in general is a set of sensors that send signals that will be processed in a central processor for further processing. IoT networks and equipment can automate tasks, therefore, it requires minimal intervention (Mishra et al., 2024). In this hyper-connected world with the internet, traditional network architectures have stayed unchanged over the last few decades, resulting in various security concerns that might compromise IoT networks (Swathi et al., 2024). This necessitates the development of an effective malware detection model.

The development of IoT is directly proportional to the risks faced. One of the threats faced is malicious traffic. This is a serious threat to network security. This threat refers to all network traffic that invades, interferes, or steals data without permission (Wei et al., 2023a). Such network traffic activity is quite frequent nowadays and is arguably the most significant network security threat. AV-TEST (2024) reports that from year to year there has been a significant increase in the number of malwares that has been successfully identified. In 2024 alone, there will be more than 1.2 billion malware and potentially unwanted applications (PUAs). Moreover, compared to intrusion at the network layer, malware at application scale has a faster propagation nature, which leads to a higher risk. More and more network applications are also using encryption protocols such as Transport Layer Security Protocol (TLS) to protect the privacy and security of an application (Xu et al., 2021) in fact, it can make the security gap itself so that the application that is malicious is even more difficult to detect.

1.2. Research Questions

The research question that will be discussed in this study is as follows:

1. How to implement feature enhancement technique using GAN-LSTM model in imbalance dataset to improve the classification performance?
2. How to improve the quality of the generated data and the performance of the model compared to other models?

1.3. Research Objective

The objectives of this study are as follows:

1. Implementing feature enhancement technique using GAN-LSTM model in imbalance dataset to improve the classification performance.
2. Improving the quality of the generated data and the performance of the GAN-LSTM model compared to other models.

1.4. Research Benefits

The benefit of this study is to improve the quality of the imbalanced dataset using feature enhancement techniques in expectations of improving the performance of malware detection using deep learning. In addition, improving the performance of the GAN-LSTM model which is known to consume a lot of resources so that it can efficiently perform feature generation.

1.5. Research Contributions

This research is expected to contribute to several aspects such as:

1. Improvement of the dataset for improving classification performance using deep learning methods.
2. Improvement of the GAN-LSTM performance in feature enhancement.

1.6. Research Scope

This research has some limitations that need to be paid attention to:

1. Testing of the proposed method using the IoT-23 dataset consisting of 11 malware classes.
2. Testing using a personal computer device with specifications:
 - a. Processor : Intel Core i5-12400F
 - b. Memory : 32GB DDR4 3600mhz
 - c. GPU : NVIDIA RTX 3060 with 12GB VRAM
 - d. Internal Storage : 500GB of SSD Storage

[Page intentionally left blank]

CHAPTER 2

THEORETICAL BASIS

This chapter presents the foundational theories and concepts that underpin the research study. It lays out the critical scientific principles, methodologies, and existing knowledge relevant to the development and evaluation of the proposed model. Key topics include the theoretical framework of data preprocessing techniques, machine learning algorithms, and generative adversarial networks particularly Conditional GANs and their applications in cybersecurity domains such as the IoT-23 dataset. This theoretical exploration not only establishes the context for the research but also identifies gaps and challenges that the current study aims to address through innovative modeling approaches.

2.1. Related Works

2.1.1. Support Vector Machines

Support Vector Machines (SVMs) have been widely adopted in cybersecurity research, particularly for the detection and classification of malware in Internet of Things (IoT) environments. Previous studies consistently highlight the growing security risks posed by IoT malware, which exploit vulnerabilities inherent to devices primarily designed for functionality rather than protection. Shi et al. (2024) emphasize that the vast majority of IoT devices prioritize utility, neglecting critical security aspects, leaving them open to cyber threats. To address this, Shi and colleagues applied a one-class classification approach using SVMs, where the model is trained exclusively on benign traffic data. This strategy enables the detection of anomalous or malicious traffic by identifying deviations from the learned normal behavior. Their results demonstrated the effectiveness of this method, achieving an outstanding recall rate of 100%, which indicates that all malicious instances were successfully detected, coupled with precision rates exceeding 80% and 90% across several test datasets, signifying reliable accuracy in distinguishing malware from benign samples.

It is important to recognize some inherent challenges in applying SVMs to datasets typical of IoT network traffic, such as the IoT-23 dataset. According to Abdalgawad et al. (2022), SVMs can exhibit performance degradation when

handling imbalanced datasets, which are common in IoT security where benign traffic overwhelmingly outnumbers malicious traffic. This imbalance causes the SVM to bias towards the majority class, consequently reducing its ability to accurately classify the minority, often more critical, malicious class. Recognizing and mitigating this bias is crucial for developing robust malware detection systems. Techniques such as data resampling, adjusted class weights, or hybrid models combining SVMs with other algorithms have been proposed to enhance detection in such imbalanced contexts. Thus, while SVMs remain powerful and theoretically sound classifiers, their practical application to IoT malware detection demands careful consideration of dataset characteristics and method adaptations that address challenges like class imbalance to achieve optimal performance.

2.1.2. Synthetic Minority Oversampling Technique

Synthetic Minority Oversampling Technique (SMOTE) is one of the methods that can be used to overcome data imbalance (Dablain et al., 2023). SMOTE is able to create synthetic data on minority classes. Often minority classes due to their small number, are less representative of the class and cause low classification performance. SMOTE's ability to learn to create synthetic data in minority classes is able to improve classification performance. Study of the use of SMOTE with varying imbalance datasets such as the Pima Indian Diabetes Dataset, WPBC (Wisconsin Prognostic Breast Cancer), Ionosphere Dataset, Breast-cancer-wisconsin Dataset, dan WDBC (Wisconsin Diagnostic Breast Cancer) show improved classification performance using Random Forest classifier. The performance score obtained for the model used has an average Accuracy of 96.97%, OOB Error 4.43%, F-value 96.53%, and G-value: 97.06% (Wang et al., 2021).

However, in its implementation, SMOTE has several shortcomings such as the possibility of producing fake samples that misshape the representation of the minority class which may lead to model overgeneralization. SMOTE sometimes suffers from noise in the data which may introduce more distortion into the synthetic samples produced. SMOTE may also neglect some boundary instances that are critical in the definition of the decision boundary for different classes (Swana et al., 2022).

2.1.3. Generative Adversarial Networks for Malware Detection

Generative Adversarial Networks (GANs) have emerged as a powerful and versatile tool in the domain of malware detection, particularly within the challenging environment of Internet of Things (IoT) networks. GANs operate on the principle of adversarial learning, where two neural networks, the generator and the discriminator, compete against each other to improve performance. Since their introduction by Goodfellow et al. (2014), GANs have been employed in a variety of applications, extending far beyond their initial scope of image synthesis to areas such as cybersecurity. In IoT malware detection, GANs contribute not only as classifiers but also as powerful enablers of data augmentation, anomaly detection, and feature learning. This adaptability is important given the dynamic and evolving nature of malware threats that can be difficult to characterize with traditional static datasets. By simulating realistic attack scenarios, GANs help researchers and practitioners to build more robust malware detection systems that generalize better to unseen or obfuscated malicious activities.

Research by Shareef et al. (2024) highlights the practical application of GANs combined with optimization algorithms like the zebra optimization technique to enhance detection accuracy. Their hybrid approach achieved high performance metrics, including accuracy, precision, recall, and F1-score, all exceeding 92%, demonstrating the effectiveness of GANs in generating meaningful data representations and improving classification results. Beyond detection, GANs serve an indispensable role in addressing critical challenges such as class imbalance, which is pervasive in IoT datasets where benign traffic far outweighs malicious samples. Almasre & Subahi (2024) demonstrated the utility of GANs for synthetic data generation in network traffic datasets, so it can enrich the minority classes and enable more balanced and comprehensive model training. This enhancement significantly improves anomaly detection capabilities by providing diverse and realistic attack patterns. GANs not only improve malware detection efficacy but also contribute to the overall resilience and adaptability of IoT cybersecurity frameworks.

2.1.4. Temporal Pattern Learning on Generative Models

Temporal pattern learning is crucial for a model to learn how data values change and evolve across time. Generative models can benefit from this ability, enabling them to capture complex dependencies that evolve in a sequence. It can learn to generate spatiotemporal data accurately in terms of sequence generation. Strategies that are commonly used by researchers are combining generative models with recurrent or attention-based architecture models in the expectation of a much better modelling in spatial and temporal data. With that ability, generated data can mimic a realistic dataset and further improve IDS.

Integration of an LSTM network into generative models has been studied before, and it improves the temporal learning of the model. Graph convolutional networks (GCN) with LSTM networks inside GAN were used by Gao et al. (2021). GCN can capture spatial dependencies between nodes at each timestamp, while LSTM models capture temporal dependencies. Both combined make the generator able to generate realistic graph structure at future time steps, this ability addresses the challenges in handling evolving temporal patterns on network systems.

Shao et al. (2023) develops a model called Temporal-Topological Demand Prediction (TTDP) GAN. TTDP-GAN architecture incorporates LSTM and multi-head temporal self-attention to learn temporal dependencies at various scales. This design helps the model to capture short- and long-term temporal correlation in a spatiotemporal dataset. Adversarial training can train the generator to creates data that can reflect real data distributions based on temporal sequences.

Generative models combined with a temporal model like LSTM show that it is capable of producing high-quality synthetic data. It can learn complex spatial and temporal distributions. Some models may be able to learn spatial features, but when sequence and timing play a crucial role, the ability to model time-dependent behaviors remains limited. Based on these findings, this study aims to generate a model that can learn from spatiotemporal features on a dataset through combining GANs and LSTM. Another research points out that we can benefit from using LSTM. A paper by Altunay & Albayrak (2023) explain that LSTM is best used with sequential dataset. With the ability to remember past information, LSTM are able to identify complex pattern of the malware attacks. LSTM in this paper is used with

CNN and achieving accuracy of 92.9% with UNSW-NB15 dataset, and 99.8% with X-IIoTID dataset. It shows that LSTM can be used or combined with other models easily. But there are some drawbacks using it. Imbalance dataset can hinder the LSTM performance and also combined with high computational resource due to LSTM architecture that has multiple gates (input, output, and forget gates).

2.1.5. Improving Tabular Data Diversity using GANs

Researchers have widely used GANs for data augmentation. The data augmentation includes various domains such as image, audio, and network signal data, regardless of whether the data is high or low dimensional. Using the UNSW-NB15 dataset, research by Rahman et al. (2024) demonstrates the ability of GAN to create synthetic data that realistically mimics the original dataset. The resulting GAN dataset can replace or complement the main dataset. The dataset that has been augmented is classified using several ML models, including logistic regression (LR), random forest (RF), gaussian naïve bayes (GNB), and others. The classification results yielded values between 84% and 90% in evaluation metrics such as accuracy, precision, recall, and f1-score.

GAN-based augmentation has also been applied to address data imbalance issues. Lu et al. (2022) proposed an augmented data model to overcome data imbalance in sensor readings from robot anomalies. The original dataset exhibited 1:100 class ratio, was successfully adjusted to a more balanced ratio between 1:5 and 1:1. The classification was performed using models such as support vector machine (SVM), isolation forest, light gradient-boosting machine (LighGBM), and multi-layer perceptron (MLP). The results showed an increase in accuracy from 87% to 97%.

Existing studies show that GANs are capable of learning complex distributions of original data and can produce synthetic data that resembles real data. However, in its implementation, GANs still need improvement. GANs requires a large enough resource and is unstable, causing mode collapse (Zhang & Liu, 2022). Lack of temporal understanding of GANs leads to poor performance, resulting in poor quality of generated data (Lu et al., 2022).

2.1.6. Advancing Classification Performance Through Synthetic Feature Augmentation

Several feature enhancement (FE) models have been developed to improve the performance of IDS. The FE approach is carried out through an unsupervised and supervised model. Supervised Feature enhancement network (FENet) model developed by Cheng et al. (2025) able to directly learn labeled datasets through attention mechanisms on discriminatory spatial and channel features. The model is able to improve the performance of the classification by increasing the expressiveness of existing features, such as the number of pixels entered, and adding more channel attention to take into account so there is much more information entered into the calculation. However, in its application, this model is less capable of temporal features, and the existing dataset must go through an annotation process first to increase processing time and resource utilization. Another supervised model is channel-based feature enhancement through feature calibration and attention fusion (CEFC & CEAFF) developed by Zheng et al. (Zheng et al., 2025). That model can enhance adaptive features based on class labels from the dataset. Channel attention can increase the relevance of multi-scale features and fusion on diverse attacks. This model has a high computational load and has more complex tuning potential.

In addition to supervised models, there are unsupervised models that can also be used. Autoencoder (AE) is one of the unsupervised models developed by Li et al. (2024). AE is used in feature extraction where downstream is supervised fine-tuned. The model is able to retrieve nonlinear information from a dataset without a label but is resource-intensive and easier to enter an over or underfitting. Another feature generation model uses K-Means proposed by Wei et al. (2023). K-Means is used for clustering of existing features to form new large groups. The group will later be a representation of new features that correlate with a particular class. The newly created feature is then combined with the original feature to increase the difference between normal and abnormal traffic. This model is well applied to data with small dimensions but on complex distributions it is less effective and has the potential to stop at the local minima so that it fails to group features, especially on the imbalance dataset.

2.2. Theoretical Basis

2.2.1. Internet of Things (IoT)

The Internet of Things (IoT) refers to an interconnected network of physical devices, sensors, actuators, and embedded software systems that facilitate the collection, transmission, and exchange of data across the internet and other communication networks (Humayun et al., 2021). Fundamentally, IoT devices serve as smart things that are connected to a virtual world, that enables remote monitoring, control, and automation through internet-enabled infrastructures (Koohang et al., 2022). This connectivity transforms otherwise simple objects into smart devices capable of interacting with users and other systems, as a result enhancing the automation and efficiency of various processes.

IoT technology can provide unprecedented convenience and operational efficiency across multiple sectors. By enabling real-time connectivity and data exchange, IoT systems allow users to remotely control and monitor devices, resulting in improved resource management, energy efficiency, and service quality. For instance, IoT applications have revolutionized domains such as smart homes, healthcare, industrial automation, transportation, and agriculture by facilitating innovations like predictive maintenance, environmental monitoring, and seamless remote access (Allam et al., 2022).

The rapid increase in the use of IoT devices has been exceptional in these recent years. As of recent statistics, approximately 31 billion devices were connected through IoT networks globally, a number expected to soar to around 75 billion by 2025 according to Schiller et al. (2022). This exponential growth is proof to the widespread adoption of IoT technology, fueled by advancements in wireless communication, sensor technologies, and cloud computing.

Despite its benefits, the rapid expansion of IoT presents a concerning security challenges. The vast number of interconnected devices, often operating with limited computational resources and lacking standardized security protocols, increases the attack surface for potential cyber threats. Studies indicate that as many as 178 million IoT devices were exposed on public networks in 2022, many of which were vulnerable to unauthorized access and exploitation due to weak security configurations (Chaganti et al., 2022). This vulnerability is further confirmed by

the relatively low awareness of cybersecurity best practices among many users and organizations deploying IoT solutions.

As IoT devices integrate deeper into critical infrastructures and daily lives, ensuring their security becomes the most important focus. The potential consequences of compromised IoT devices range from privacy breaches and data theft to disruption of essential services and even physical harm in sectors like healthcare and transportation. Addressing these vulnerabilities requires not only great but also robust security frameworks, continuous monitoring, and advanced detection techniques to mitigate the risks associated with the interconnected IoT ecosystem.

2.2.2. Malware on IoT

Malicious software, commonly known as malware, covers a broad range of software designed to infiltrate, disrupt, or damage computer systems without the user's consent. Malware can perform harmful activities such as encrypting data to demand ransom (ransomware), stealing sensitive and personal information, covertly monitoring user behavior, and spying on communications (Moti et al., 2021). The rapid evolution of malware in terms of complexity and volume has been a significant concern in cybersecurity. According to recent projections by AVG, it is estimated that by the year 2024, there will be approximately 190,000 malware attacks occurring every second worldwide, highlighting the sheer scale and aggressive nature of modern cyber threats (Estenssoro, 2024).

This surge in malware attacks varies widely in type and intensity, presenting a growing threat to all networked systems, especially those with limited resources and weak defenses. One particularly vulnerable domain is the Internet of Things (IoT), where devices typically operate with constrained computing power, limited memory, and reduced security capabilities (Yuan et al., 2022). Despite their convenience and integral role in modern digital lifestyles, IoT devices are often inadequately protected due to these resource constraints and a widespread lack of digital security awareness among end-users (Wu et al., 2023).

The exploitation of IoT devices by malware can lead to various disruptions, including unauthorized access to private data, erosion of user privacy, degradation

of device performance, and overall reduction in user trust and experience (S. H. Khan et al., 2023). As IoT technologies increasingly permeate critical aspects of daily life, such disruptions have far-reaching consequences, affecting personal security, business operations, and even national infrastructure in some cases. The low computational resources of many IoT devices make traditional malware detection and mitigation techniques difficult to apply, necessitating specialized approaches tailored to the unique challenges of IoT ecosystems.

Consequently, the growing prevalence of IoT-targeted malware underscores the critical need for robust and early detection mechanisms. Early identification of malware not only minimizes the potential damage but also helps in maintaining the integrity, availability, and confidentiality of IoT networks. Effective countermeasures rely heavily on continuous monitoring, anomaly detection, and adaptive learning-based techniques to keep pace with the evolving malware landscape targeting IoT devices.

2.2.3. Generative Adversarial Networks

GANs is commonly used in image generation module. Because GANs is basically a neural network which in its application needs a binary data, in theory it can also generate tabular numerical data. The structure of Generative Adversarial Networks (GANs) has two components: a Generator (G) and a Discriminator (D).

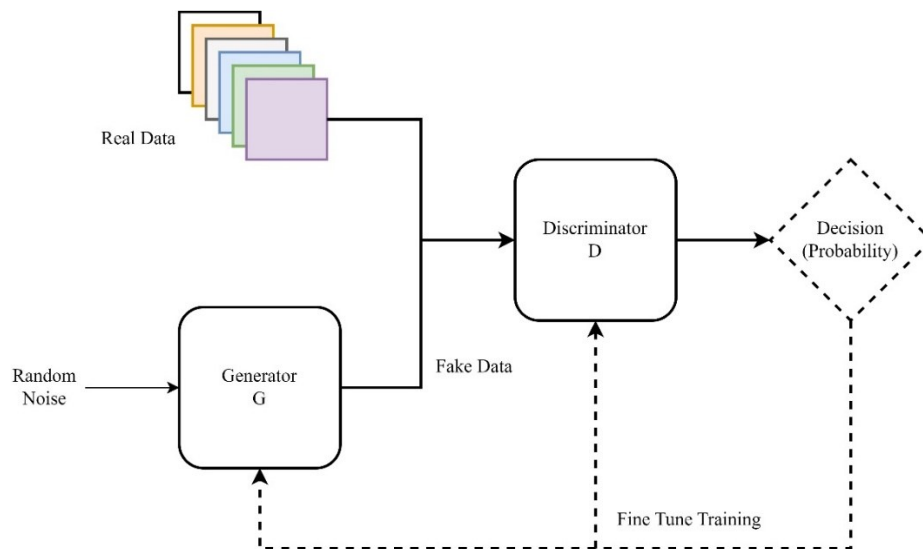


Figure 2.1 General GANs Architecture

From Figure 2.1 can be explained a short description of GAN components (Ruiz-Gándara & Gonzalez-Abril, 2024):

1. **Generator (G)** The generator performs the function of generating new data samples based on noise inputs. Random inputs (usually in the form of Gaussian noise) are taken and transformed into a data sample pertaining to the required training data. As the generator is usually deep neural network architecture, it uses a number of layers to learn from complex data such as images and also employs transposed convolutions to up sample the data.
2. **Discriminator (D)** The function of the discriminator is to identify which data samples is real (taken from the training dataset) and which one is synthetic and outputted by the generator. Guessed inputs exhibit the probability score that quantifies the degree of realism in the input, as an angle from zero or one. Similarly to the generator, the discriminator too is a deep neural network model which is most commonly designed to classify inputs into distinct groups through multiple convolutional and activation layers.

The training mechanism of GANs is formulated as a minimax game, a competitive process between two neural network models: the Generator (G) and the Discriminator (D) (Goodfellow et al., 2014). In this adversarial setup, the generator's goal is to create synthetic data samples that are indistinguishable from real data, and attempting to "fool" the discriminator by producing increasingly realistic outputs. Conversely, the discriminator functions as a binary classifier tasked with distinguishing between the genuine samples drawn from the real data distribution and the fake samples generated by the generator. This training dynamic can be understood through the concept of a zero-sum game where the generator tries to maximize the probability that the discriminator incorrectly identifies fake samples as real, while the discriminator strives to maximize its accuracy in correctly classifying both real and generated data.

The training loop alternates between these two objectives: the discriminator updates its parameters to become better at telling apart real from fake data, and the generator updates its parameters to produce more convincing fake data. This iterative "game" continues until an equilibrium is reached, ideally when the generator produces data so realistic that the discriminator can no longer reliably

distinguish real samples from generated ones, making its predictions essentially random.

Throughout this process, the GAN models engage in a feedback loop where the success of one model directly influences the improvement of the other. This dynamic pushes the generator to capture the underlying data distribution with high fidelity, resulting in synthetic outputs that appear very authentic. The minimax nature of this training provides a robust framework for learning complex data distributions without any explicit supervision on what the generated data should look like.

The GAN equation is as follows:

$$\min_G \max_D V(D, G) = A + B \quad (2.1)$$

where:

$$A = E_{x \sim p_{data}(x)} [\log(D(x))] \quad (2.2)$$

and

$$B = E_{z \sim p_z(z)} [1 - \log(D(G(z)))] \quad (2.3)$$

G is the generator, D is the discriminator, x are real data samples, and z is the noise drawn from the distribution D with the given standard deviation. $p_{data}(x)$ is the probability of the distribution of real data and $p_z(z)$ is the probability of random noise distribution.

$$Loss_D = -\log(D(x_{real})) - \log(1 - D(x_{synthetic})) \quad (2.4)$$

2.2.4. Long Short-Term Memory

Long Short-Term Memory (LSTM) networks represent a specialized architecture within the broader class of recurrent neural networks (RNNs), designed specifically to address the inherent limitations associated with learning and remembering information over extended sequences of data (M. Khan et al., 2021). While traditional or “vanilla” RNNs are capable of processing sequential information, they suffer from the well-known vanishing gradient problem, which

significantly hampers their ability to retain information from earlier time steps as sequences grow longer. This limitation reduces the effectiveness of vanilla RNNs when applied to tasks requiring long-range temporal dependencies.

To overcome this problem, LSTM networks introduce a unique structure composed of memory cells integrated with gating mechanisms that regulate the flow of information throughout the network. These gates typically called the input gate, forget gate, and output gate act as dynamic filters that control which information is admitted into the memory cell, which information is forgotten or retained, and which information is outputted to the next time-step. This refined control mechanism enables LSTMs to selectively remember or discard information, which gives the model the ability to capture and retain dependencies across much longer temporal windows as compared to vanilla RNNs.

Thanks to this architectural innovation, LSTMs have become highly effective at modeling long-term sequential data dependencies, making them invaluable for a variety of applications that hinge on understanding complex time series or sequences. For example, LSTMs are commonly employed in language modeling, where the context from words or phrases far apart in a sentence must be retained and utilized to predict future words accurately. Similarly, in time series forecasting, LSTMs can uncover patterns and trends that span wide temporal ranges, facilitating more reliable predictions over extended horizons. Beyond these, the architecture is also well-suited for domains like speech recognition, anomaly detection in network traffic, and other settings requiring the retention of information over long sequences (Malashin et al., 2024). The detailed LSTM structure is depicted in Figure 2.2, highlighting its memory and gating mechanisms responsible for this temporal learning capability (Chu et al., 2024).

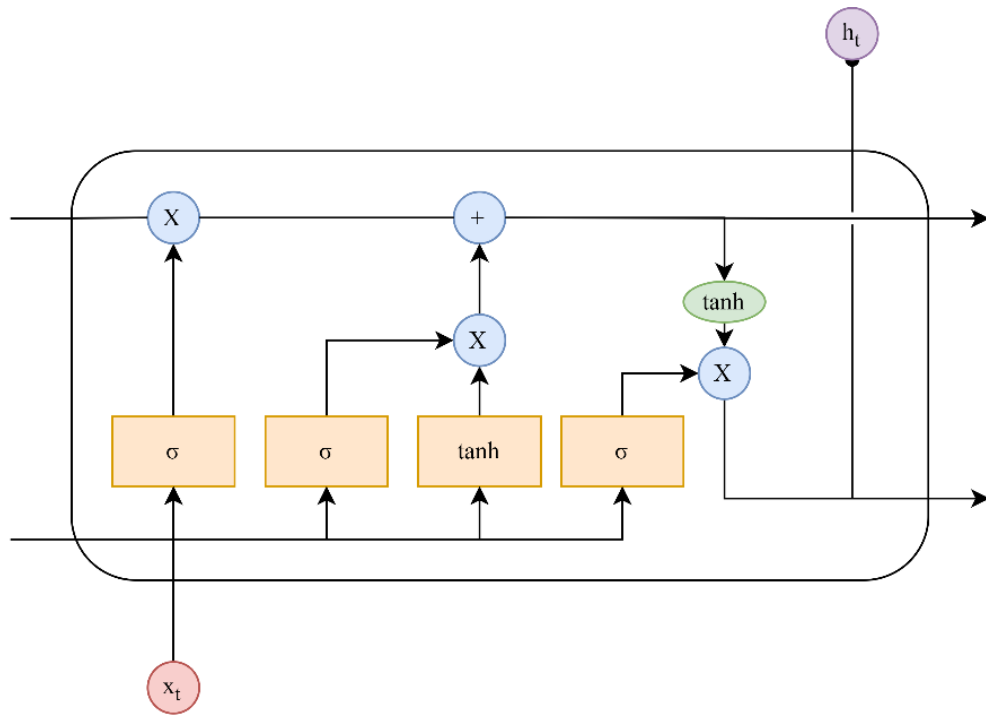


Figure 2.2 LSTM Cell (Azati et al., 2024).

The major elements of the internal working of an LSTM cell are:

1. Forget Gate: Controls the output of cell state and what information should be excluded.
2. Input Gate: Controls the cell state output and which new information is required.
3. Output Gate: Decides the information to output to cell state.

With that architecture, LSTMs are capable of solving not only short-term dependencies tasks but also long-term dependencies which made them popular for many tasks in ML and natural language processing (Sherstinsky, 2020).

2.2.5. Feature Enhancement

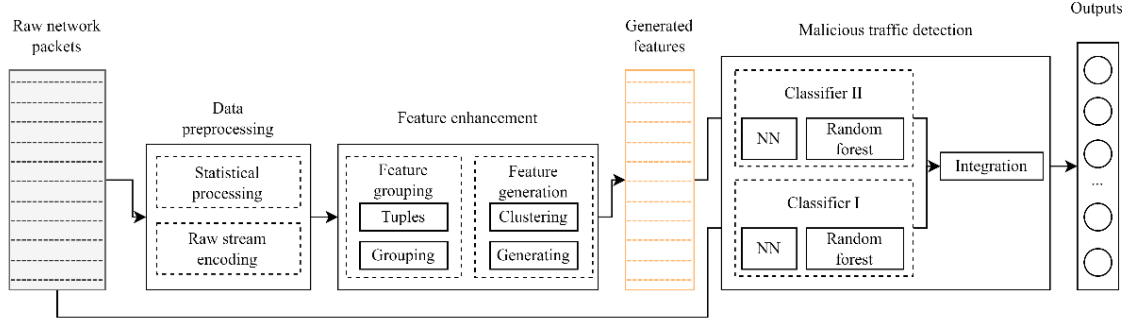


Figure 2.3 Feature Enhancement Model (Wei et al., 2023)

Figure 2.3 shows the feature enhancement model proposed by Wei et al. (2023c). Feature enhancement proposed above aims to help discriminate between normal traffic and malicious one in the context of network datasets. The process consists of two main phases: feature grouping and feature generation.

1. **Feature grouping:** This part deals with the separation of the raw features of network traffic into some smaller subgroups based on their Gaussian characteristics which include such statistical measures such as skewness, mean and standard deviation. The features are grouped such that those which have greater variation of the Gaussians are classified into different groups. This assists in determining the slight differences that exist between the two classes of traffic, normal and attacks.
2. **Feature generation:** Once the grouping has been done, the procedure uses the k-means clustering algorithm to create cluster features from the grouped features. Thus, the set of original features becomes the distance of those items from the clustering centers of these groups. This change augments the variety of the characteristics of the samples of traffic and increases the differences between the normal traffic and the attack traffic.

K-means clustering helps in feature generation. This method helps in natural grouping of raw feature data. this causes the dimensions of the data train to be reduced (D. Yang et al., 2023). The k-means equation is intended to minimize commonly used objective functions such as mean square error. the k-means equation can be seen below.

$$J = \sum_{i=1}^k \sum_{j=1}^n \|x_j - v_i\|^2 \quad (2.5)$$

The overall goal of the feature enhancement method is to magnify the subtle differences between normal and abnormal traffic, then improving the performance of classification models used for detecting malicious activities in the network.

2.2.6. Confusion Matrix

The confusion matrix is a fundamental tool used to summarize and evaluate the performance of classification models. It provides a comprehensive snapshot of how well a model categorizes data by breaking down predictions into four key outcomes: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) (Nguyen et al., 2023). True positives represent instances where the model correctly identifies positive cases, while true negatives correspond to cases correctly classified as negative. False positives indicate erroneous positive classifications, and false negatives are cases where the model fails to detect positive instances.

In practical terms, the confusion matrix is organized as a square matrix where the rows correspond to the actual class labels, and the columns represent the predicted class labels. This arrangement allows stakeholders to visualize the distribution of correct and incorrect predictions for each class distinctly. By quantifying these values, one can derive essential performance metrics such as accuracy, precision, recall, and F1-score, which provide deeper insights into the strengths and weaknesses of the classifier.

In the domain of malware analysis, the confusion matrix has important role in assessing the effectiveness of classification algorithms by visually representing how samples of malware are categorized into different classes. Specifically, when dealing with datasets like IoT malware, where the classifier needs to distinguish between multiple malware families or attack types, the confusion matrix illustrates which classes are being correctly recognized and which ones tend to be misclassified. For instance, in the case of classifying malware samples into 13 distinct categories, the confusion matrix can reveal if certain types of malwares are often confused with others, identifying potential weaknesses in the model (Massarelli et al., 2020).

The performance of the confusion matrix is measured in the equation below:

$$Accuracy = \frac{TP+TN}{TP+TN+PN+FN} \quad (2.6)$$

$$Precision = \frac{TP}{TP+FP} \quad (2.7)$$

$$Recall = \frac{TP}{TP+FN} \quad (2.8)$$

$$F1 - score = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (2.9)$$

2.2.7. Conditional GANs

Conditional Generative Adversarial Networks (CGANs) represent a powerful extension of the original Generative Adversarial Networks (GANs) framework by incorporating auxiliary information, commonly class labels, into both the generator and discriminator networks. This conditioning enables the generation of synthetic data samples that are not only realistic but also class-specific, facilitating targeted data augmentation for imbalanced datasets. Those abilities enable CGANs to enhance model training by improving representation across underrepresented classes, which is a persistent challenge in cybersecurity datasets.

The IoT-23 dataset, widely utilized for benchmarking intrusion detection systems (IDS) in Internet of Things (IoT) environments, comprises diverse labeled network traffic, including benign flows and numerous attack types. However, it suffers from inherent class imbalance, particularly for rare attack categories. This imbalance limits the effectiveness of conventional machine learning models, often resulting in biased detection performance skewed toward majority classes.

Recent studies have demonstrated that CGAN architecture provides an effective solution for augmenting minority class samples within IoT security datasets. By conditioning on attack class labels, CGANs synthesize high-fidelity artificial samples that preserve the statistical and semantic characteristics of underrepresented attacks, ultimately improving the robustness and accuracy of IDS models.

Alabsi et al. (2023) introduced a Conditional Tabular GAN (CTGAN)-based intrusion detection system that leveraged label conditioning to generate synthetic cybersecurity event data focusing on DDoS and DoS attacks within the IoT-23

dataset. Their results showed significant improvements in detection accuracy and class balance, that shows CGAN's efficacy in addressing data scarcity in IoT network attacks.

Almasre & Subahi (2024) proposed a CGAN framework designed to create realistic synthetic IoT network datasets by conditioning on traffic classes. Their approach tackled the issue of data imbalance and scarcity in IoT-23, which enable downstream models to achieve better generalization capabilities and more reliable detection of rare attacks.

More recently, hybrid architectures such as CE-GAN, combining CGAN with encoder-decoder models, have achieved superior performance in generating tabula IoT network traffic data that retain complex inter-feature dependencies and class distinctions. This advancement has further contributed to overcoming class imbalance problems in network intrusion detection tasks (Yang et al., 2025).

Building upon these insights, the CGAN-LSTM model proposed in this thesis conditions synthetic data generation on class labels from the IoT-23 dataset to augment minority attack categories. That model is able to create balanced, high-quality synthetic data, enhancing classifier training and detection performance in imbalanced IoT intrusion detection scenarios.

[Page intentionally left blank]

CHAPTER 3 RESEARCH METHODOLOGY

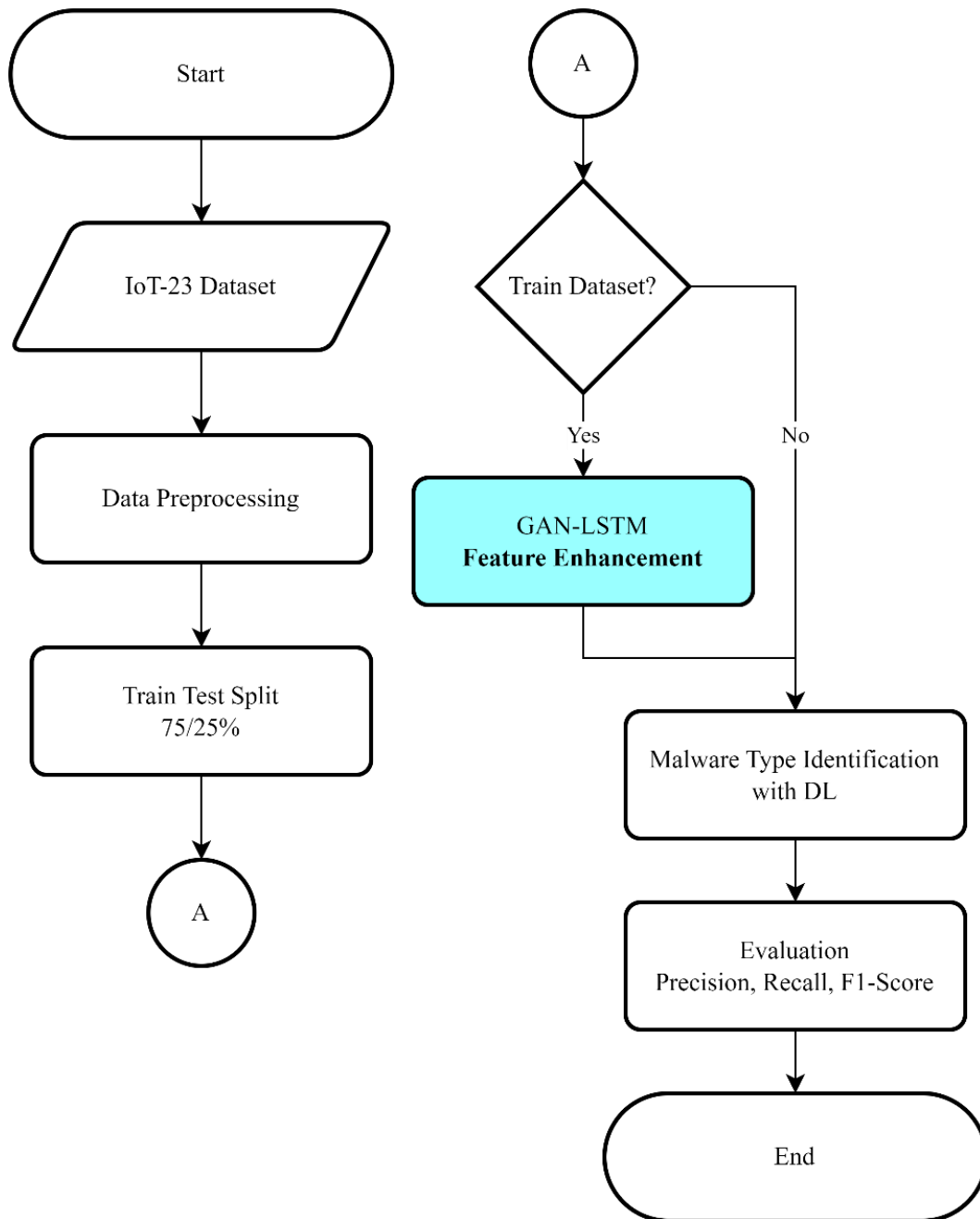


Figure 3.1 Research Workflow

In this chapter, the methodology and stages of the research are thoroughly explained, outlining the systematic approach employed to achieve the study's objectives. This research utilizes a **feature enhancement** strategy to perform malware analysis on the IoT-23 dataset, which contains labelled network traffic

data generated by various IoT devices under both normal and attack conditions. As depicted in Figure 3.1, the stage highlighted in blue represents the core proposed method of this study. The research process begins with the essential step of pre-processing the IoT-23 dataset, ensuring the data is clean, normalized, and properly formatted. Following this, the dataset is partitioned into training and testing subsets, where 75% of the data is allocated for training and 25% for testing. This split is useful to evaluate the model's ability to generalize and accurately classify unseen data in real-world scenarios.

The training dataset then undergoes a feature enhancement phase, where key attributes are transformed or emphasized to improve the performance of machine learning classifiers. Subsequently, multiple deep learning architectures namely Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory networks (LSTM) are employed to classify the enhanced training data. These models are chosen for their ability to capture complex temporal and spatial patterns in network traffic data that are indicative of malicious behavior. Finally, the classification results obtained from the enhanced dataset are rigorously compared against results from the original, untreated dataset. This comparative evaluation provides insights into the impact of feature enhancement on model accuracy and robustness, so it enable a comprehensive analysis of both the performance improvements and the overall effectiveness of the proposed methodology.

3.1. Dataset

The IoT-23 dataset is a comprehensive labeled dataset containing both benign and malicious network traffic generated by various IoT devices. This dataset is created by capturing network traffic as these devices operate under normal conditions as well as during simulated attack scenarios, which include different types of malware and network-based attacks targeting IoT devices. The network traffic is captured using packet capture (PCAP) tools and monitored in real-time by a powerful network analysis framework called Zeek. Zeek acts as a network security monitor that processes the raw packet data collected from the network and extracts high-level logs and metadata useful for cybersecurity research and analysis. Essentially, Zeek inspects all network activity passing through the monitored

environment, recording detailed connection and event logs while still keeping the original PCAP files of the traffic flows. The PCAP files constitute the raw packet data that forms the basis of the IoT-23 dataset. This dataset can help researchers to analyze both the detailed network communications and the derived metadata for anomaly detection or attack classification.

The architecture outlined in Figure 3.2 shows attackers launching various cyberattacks on a network containing multiple IoT devices, including cameras, smart taps, connected vehicles, and home automation systems. All traffic flowing through the network is captured and logged by Zeek, which segments this network data into structured PCAP files that represent the network traffic during both benign and attack phases. These labeled PCAP files provide a rich resource for training and evaluating machine learning models in the fields of intrusion detection and IoT security. The dataset published by Garcia et al. (2020) ensures that all captured traffic is preserved with detailed annotations for the type of attack and device involved. This dataset can give a valuable resource to create realistic and reproducible security research using open and publicly available data.

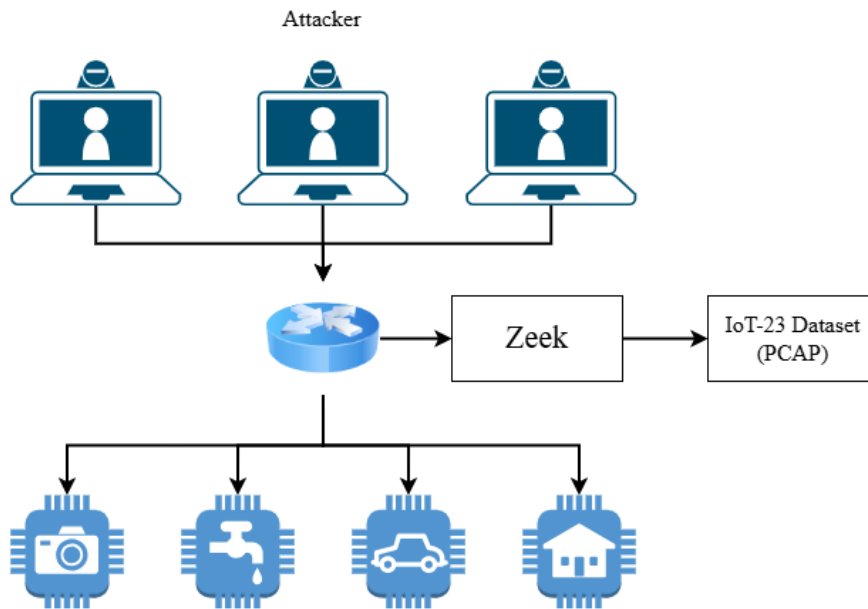


Figure 3.2 IoT-23 dataset capture diagram (Garcia et al., 2020)

The features selected from the IoT-23 dataset for this study are primarily numerical attributes that can represent various types of network traffic, including

attacks. These features encompass essential data points such as the duration of attacks, which provide important temporal context regarding how long malicious or benign events persist. The number of packets sent and received are some of the important feature, as it reveals the volume and flow of network communication during sessions, that helps to differentiate between normal and abnormal traffic patterns. Additionally, source and destination port numbers are selected to pinpoint where packets originate and terminate, allowing the model to recognize common or suspicious points of network activity. Furthermore, features indicating anomalies, such as missing bytes or unusual packet sizes, are included to capture irregularities that often accompany malicious behavior.

3.2. Data Preprocessing

The dataset used in this study is a network connection log dataset with Packet Capture (PCAP) format that records network traffic on packets. Then this dataset is extracted into the ‘conn.log.labeled’ format using the Zeek Network Analyzer tools. After extracting the data from the PCAP file using Zeek’s tools, the ‘conn.log.labeled’ dataset then enters the data pre-processing stage to prepare the data for further analysis. In this stage, the 23 subdatasets are made into one in H5 format.

3.2.1. One-Hot Encoding

One of the fundamental steps in the data preprocessing pipeline is the application of one-hot encoding, a technique crucial for converting categorical variables into a numerical representation that machine learning models can process. In the context of the IoT-23 dataset, categorical features namely ‘label’ inherently possess non-numeric labels that cannot be directly ingested by deep learning models, particularly LSTM networks which require numeric input sequences for training and inference (Gamal et al., 2024).

One-hot encoding addresses this limitation by transforming each unique category within a categorical feature into a distinct binary vector. Each category is represented as a vector in which all elements are 0 except for the position corresponding to the specific category, which is marked as 1. This transformation eliminates any implicit ordinal relationship that might be misinterpreted by numeric

encodings such as label encoding, so it can preserve the categorical nature of the data without imposing unintended hierarchies.

The importance of one-hot encoding in this study plays a vital role in enhancing model robustness and interpretability. By representing categorical variables in a higher-dimensional binary space, models can learn nuanced relationships and interactions across different categories without being hindered by scale differences or ordinality assumptions. Furthermore, this encoding guarantees compatibility with upstream normalization or scaling techniques, contributing to improved convergence during training and ultimately better predictive performance on IoT network traffic classification and anomaly detection tasks.

3.2.2. MinMax Scaler

Data preprocessing is a critical step in preparing the IoT-23 dataset for effective analysis and modeling, as it ensures that the features are on a comparable scale. One of the key preprocessing techniques applied is MinMax Scaler, which normalizes the feature values to a specific range, where in this study it is from -1 to +1. This normalization is essential because the IoT-23 dataset contains features with varying units and magnitudes, which, if left unchanged, could negatively impact the performance of machine learning algorithms.

MinMax Scaling works by transforming the original feature values linearly so that the minimum value of each feature maps to -1, and the maximum maps to +1. All intermediate values are proportionally adjusted within this range, keeping the original distribution's shape but ensuring that no feature dominates due to its scale. This is especially important for algorithms sensitive to the numeric range of inputs, such as neural networks and distance-based classifiers, which are often used in IoT anomaly detection tasks.

$$\mathbf{x}_{scaled} = \frac{\mathbf{x} - \mathbf{x}_{min}}{\mathbf{x}_{max} - \mathbf{x}_{min}} \times (\mathbf{1} - (-\mathbf{1})) + (-\mathbf{1}) \quad (3.1)$$

where \mathbf{x} is the original data point, \mathbf{x}_{min} and \mathbf{x}_{max} is the minimum and maximum value of the feature across the dataset, and \mathbf{x}_{scaled} is the transformed value in the range of -1 to 1.

By applying this scaling approach to the IoT-23 dataset, we ensure that all feature data is standardized within a uniform range, improving the training stability of models and accelerating convergence. Moreover, scaling to a symmetrical range around zero retains important information about the sign and relative position of values, which can be beneficial for certain activation functions and algorithms.

3.3. Train Test Split

Train-test splitting is a fundamental and widely adopted procedure in machine learning used to evaluate the generalizability and performance of predictive models. The core idea behind this method is to partition the available dataset into two separate subsets: one designed for training the model and the other reserved for testing its performance. The training dataset serves as the foundation on which the model learns patterns, relationships, and features present in the data, allowing it to adjust its internal parameters accordingly. By learning from this subset, the model creates a mapping from input features to their corresponding labels or outputs.

The testing dataset is essential in assessing the model's ability to generalize to unseen data that was not available during the training phase. This separation helps to simulate real-world scenarios where the model encounters new inputs and must make accurate predictions without prior exposure. Evaluating the model on the test dataset thus provides an unbiased estimate of its predictive performance and robustness. This practice helps detect issues such as overfitting, where a model performs well on training data but poorly on new data. The train-test split provides reliable assessment of machine learning models, guiding the selection and tuning of algorithms for optimal results.

The first step, the train-test split, also has a few of the following defined objectives (van der Goot, 2021):

1. **Model development:** The training set comprises the observations to train the model and thus, the patterns and relationships can be learned from data.
2. **Performance evaluation:** The test set provides unbiasedness towards evaluation in the sense that the model is being tested in data that was not covered during training.

3. Overfitting detection: When researchers perform evaluations on the training and test sets, it allows them to measure performance differences and thus, determine the extent to which a model is overfitted.
4. Generalization assessment: The test set also assists in evaluating the expected performance a model can yield in practice when it encounters new and unseen data.

In most cases, more data usually goes to the training set and 80-20 or even 70-30 ratios are followed (Khan, 2022). In this study, 75-25 train test split is used where the train data will be processed with FE-CGAN-LSTM model and test data will be used in the end of the model for comparing result in confusion matrix. The selection of that train test split is based on best distribution of train and test data itself and optimal for this dataset (Wakamiya et al., 2024).

3.4. Hybrid FE-GAN-LSTM

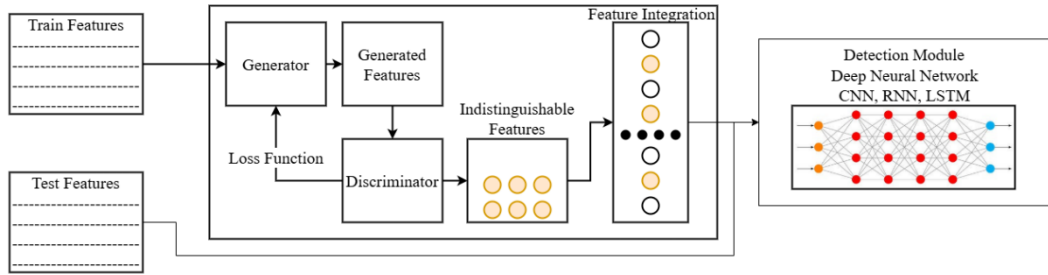


Figure 3.3 Model of the FE-GAN-LSTM

The GAN component functions as the central framework in the hybrid architecture designed for this study that integrates LSTM networks within its structural elements. This integration enables the GAN to use the LSTM's strong temporal learning capabilities while executing feature enhancement tasks on sequential data. The Generator components of the GAN incorporate LSTM networks, that allows the framework to process and learn from time-dependent features inherent in the datasets. The sequential nature of data is crucial for IoT network traffic, which often involves temporal dependencies and patterns. By embedding LSTMs, the GAN is expected to be better in modelling these sequences, improving the quality and representativeness of generated feature data, which helps in distinguishing between normal and malicious network behavior much more

effective. The overall architecture, including the GAN and embedded LSTM structures, is illustrated in Figure 3.3 of this study.

One of the primary advantages of employing LSTM networks within the GAN, especially when working with the IoT-23 dataset, lies in their ability to capture and maintain context over extended sequences of data. IoT-23 datasets are fundamentally sequential as they consist of time-series representations of network traffic activities. LSTMs' design includes memory cells that store information over long periods, that enable the model to retain relevant contextual data and temporal dependencies that are critical for generating realistic synthetic samples. This temporal preservation helps the GAN framework generate feature-enhanced data that is not only statistically consistent but contextually meaningful with respect to the sequence of events in network traffic. As a result, the inclusion of LSTMs in the GAN allows for better sequential data generation and more accurate modeling suited for intrusion detection and anomaly recognition tasks within IoT environments.

3.4.1. Generator

The Generator component, which contains LSTM layers, takes the input features (both train and test) and creates synthetic features through a generative process. The LSTM within the generator helps in learning temporal dependencies in the input data, enabling the generation of more realistic and temporally coherent features. The loss function provides feedback for improving both the generation quality and the temporal learning process.

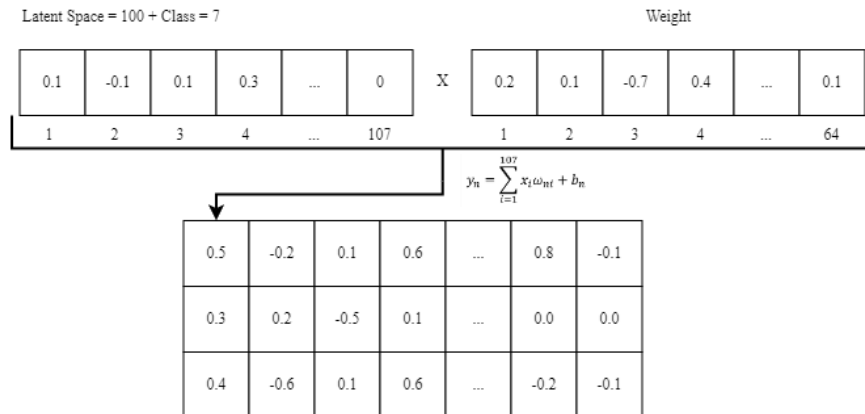


Figure 3.4 Input Layer of Generator

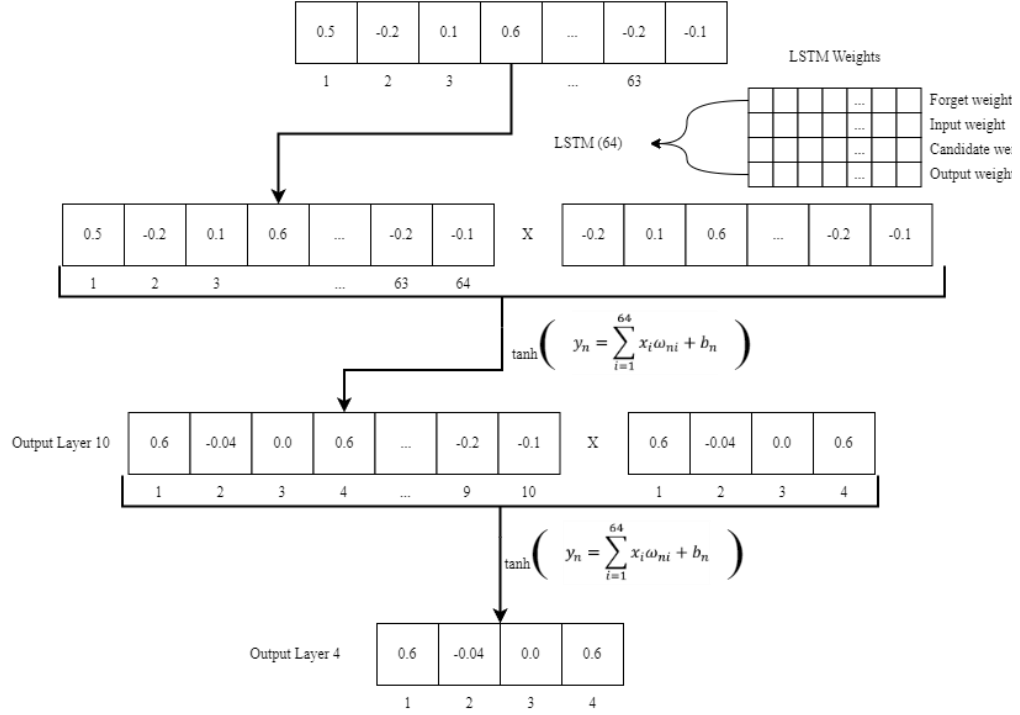


Figure 3.5 CGAN-LSTM Generator Architecture

Both Figure 3.4 and Figure 3.5 shows generator architecture of CGAN-LSTM model. In this model, generator try to generate features that closely resemble the real dataset. Using latent space and class label, generator generate random number that later multiplied by random weight of dense layer and added with bias. After that, it will generate 8x8 vector or 64 neurons. These data are then reshaped to create a timestep feature from 64 to 1,64, where it will be much easier for the LSTM layer to process it. In the LSTM layer, the data is then multiplied by some weights, such as forget, input, candidate, and output weights. It will generate another 1,64 neurons with its value. The data is then feed into the output layer where it dense into 10 features with activation function of tanh. The output layer is then reduced into 4 features as second output layer with activation function of tanh. The two outputs layer becomes the input of the discriminator model after concatenation. The discriminator model will determine if the data is real or fake by comparing it with the original dataset. All weight is then updated by the discriminator through discriminator loss function.

3.4.2. Discriminator

The Discriminator, also containing LSTM layers, evaluates the quality of features produced by the Generator. The LSTM within the discriminator helps in assessing the temporal consistency and authenticity of the generated features. This component produces “Indistinguishable Features” that meet certain quality criteria, which are then passed to the Feature Integration component before final processing by the Detection Module.

To mitigate the problem of mode collapse residing in the GAN-LSTM structures while generating signals from data sources with imbalanced datasets for instance IoT-23, a blend of hyperparameter search, architectural adaptation and suitable training methodologies is essential (Liu & Liu, 2021).

Hyperparameter Tuning is important as it helps viability issues from arising. Changing the learning rate for the generator and the discriminator in model training helps to ensure that neither model goes faster than the other when being trained. A good batch size range should be used too and other designs besides the one currently being used should be tried out since this will improve the robustness of the GAN.

The need for stability techniques is of importance because CGAN model relies on imbalanced data. With imbalanced data, methods like spectral normalization or gradient penalty significantly enhance the stability of the training by limiting the weight update of the discriminator. Further, changing the architecture to CGAN allows for eliminating diversity issues as generation of data gets controlled by images or labels (Gopali et al., 2021).

To prevent mode collapse more stable loss functions can be used by implementing alternative GAN types, for example, Wasserstein GAN (WGAN) or Least Squares GAN (LSGAN). In addition, the concept of progressive growing is addressed which allows for both the generator and the discriminator to be more complex as the training progresses. This type of growing approach contributes to improved system quality as well as enhancement on the output. Students are also directed to write components of working papers in appropriate formats (Ruiz-Gándara & Gonzalez-Abril, 2024).

SMOTE (Synthetic Minority Over-sampling Technique) can also be used to address the class imbalance problem by producing more instances of the under-

represented or minority classes. This means that appropriate input is fed into the GAN model which in turn is able to produce different kinds of outputs that are accurate. In the case of data imbalance, a training L2 regularization is beneficial as it helps manage complexity of the models and prevent overfitting (Chatterjee et al., 2025).

Monitoring of the performance of the different GAN for example is crucial since it enables users to evaluate the quality of the samples generated. By actively monitoring the GAN's performance, it is possible to prevent future collapse and instability. In order to save computational resources and to avoid degrading the quality of the model, changes can be made to set early stopping conditions when signs of instability are detected.

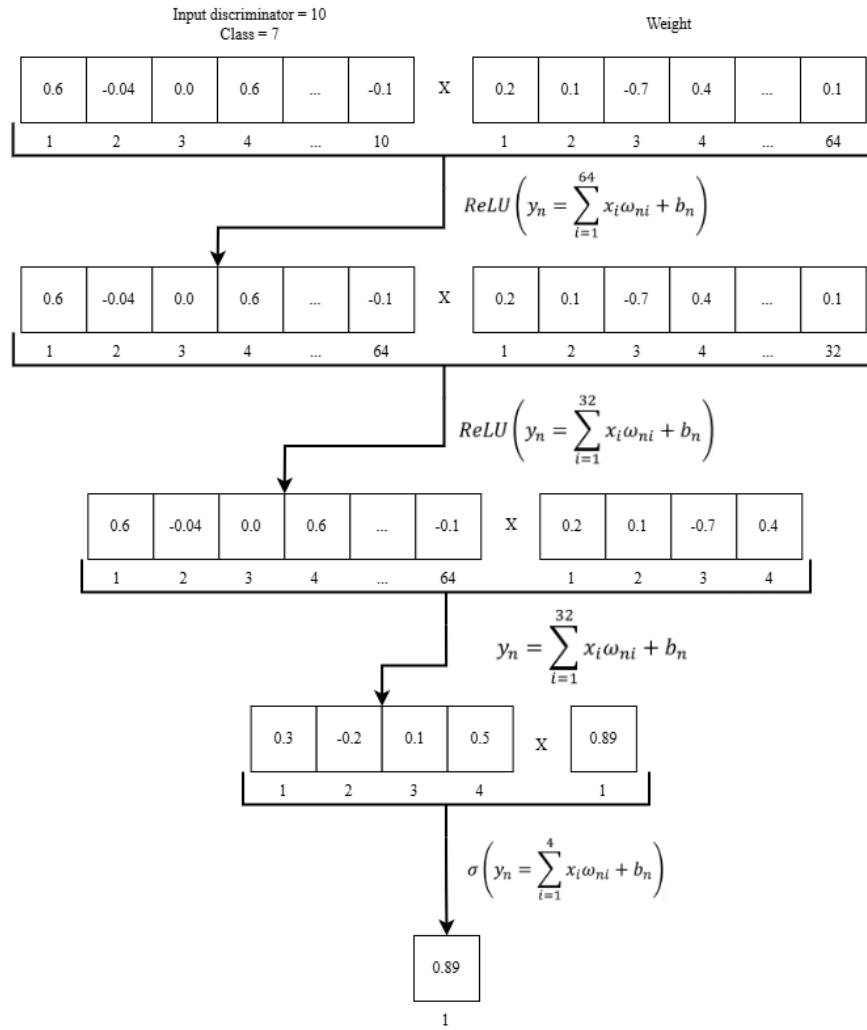


Figure 3.6 CGAN-LSTM Discriminator Architecture

Figure 3.6 shows the discriminator architecture of CGAN-LSTM model. The output of the generator model becomes the input for discriminator. The input is then multiplied by weight and added bias up to 64 neurons or vectors. Then the sum of it gets the ReLU function for easier decision of real or fake samples. The data is given repeated treatment for the 32 and 4-neuron layers until it is reduced to a single neuron using sigmoid activation, after which the loss is calculated. Loss near 1 represents uncertainty in the discriminator decision; otherwise, if it's close to 0, then the discriminator is certain that the dataset is real or fake.

3.5. Model Evaluation

The evaluation methodology employs stratified 10-fold cross-validation, a robust technique particularly suited for imbalanced datasets like IoT-23. This approach partitions the dataset into 10 folds while still maintain the original class distribution in each subset, ensuring representative sampling of minority attack classes. The stratification mitigates bias in performance estimates, given the inherent class imbalance in IoT security data where certain attack types may be underrepresented. Metrics like accuracy (A), precision (P), recall (R), F1-score (F) are computed across all 10 folds and averaged to produce final performance estimates. To help this evaluation, this study employs 3 classifiers namely CNN, RNN, and LSTM with visual approach using PCA evaluation.

3.5.1. Visual Evaluation using Principal Component Analysis

Principal Component Analysis (PCA) is a statistical technique that uses linear multivariate methods (Sadeghi et al., 2024). PCA reduces the dimensionality of variables into much less principal components (PCs) groups. The basis of the PCs rearrangement data is the interrelationship between multiple variables that are captured in a covariance matrix. After the dimensional reduction by PCA is done, the data can be observed visually. Visualization of the data was done to inspect the dataset feature clustering which is important in understanding the structure of the real dataset compared to augmented dataset.

PCA works by calculating eigenvalue decomposition from the sample covariance matrix where the variance of the PCs is determined by the eigenvalue and the PCs direction is determined by eigenvectors (Robert Frost, 2022). $\mathbf{X} \in$

$\mathbb{R}^{n \times d}$ are the dataset matrix with n is the number of samples and d is the number of features. Covariance matrix Σ is defined as:

$$\Sigma = \frac{1}{n-1} X^T X \quad (3.2)$$

Eigenvalue decomposition:

$$\Sigma \mathbf{v}_i = \lambda_i \mathbf{v}_i \quad (3.3)$$

Where \mathbf{v}_i is the i -th PCs, λ_i are the corresponding eigenvalue, and the eigenvectors with the biggest eigenvalue represent the PC. The original data is then projected into lower-dimensional space once the PC is obtained using:

$$\mathbf{Z} = \mathbf{X} \mathbf{V}_k \quad (3.4)$$

Where $\mathbf{V}_k \in \mathbb{R}^{d \times k}$ is the matrix of the top k eigenvectors, and $\mathbf{Z} \in \mathbb{R}^{n \times k}$ is the transformed data.

In this research, PCA is used to create 2D projections for both real and augmented datasets. Those projections are created to visualize the clustering behavior. Good cluster is when a data point is closer inside their class but distanced from other classes it can also means good class separability (Cui et al., 2021). PCA analysis is then supported by Silhouette Score to further verify the class cluster cohesion.

3.5.2. Silhouette Score

Silhouette score is one of the data quality measurement metrics where this score will assess a certain data cluster, which will be compared to other clusters. Therefore, this equation can be used to evaluate the quality of the GAN-LSTM data.

The silhouette score for a single data point is calculated using the equation:

$$S_i = \frac{b_i - a_i}{\max(b_i, a_i)} \quad (3.5)$$

where a_i intra-cluster distance:

$$a_i = \frac{1}{|C_i|-1} \sum_{j \in C_i, j \neq i} d(i, j) \quad (3.6)$$

and b_i nearest-cluster distance:

$$b_i = \frac{1}{|C_k|-1} \sum_{j \in C_k} d(i, j) \quad (3.7)$$

Equation a_i Calculating the average distance from a point i with all the points within the cluster itself, while the equation b_i calculate the average distance between points i with data points in other clusters (Paramasivam et al., 2023).

For the assessment of the quality of the GAN-LSTM results using silhouette scores, the values produced can vary but there are criteria that can be used as a reference (Bousmina et al., 2023):

1. The silhouette score resulting from the equation has a value of -1 to 1.
2. A score close to a value of 1 indicates that a data point is close to the cluster it is currently occupying and has different characteristics from the neighboring cluster.
3. A score close to 0 indicates that a data point is at the boundary of characteristics between 2 different clusters.
4. A negative score indicates that a data point is not in the appropriate cluster.

In addition to being seen as a comparison of data similarity, silhouette score can be used to see if the GAN data is comparable to the raw dataset from IoT-23 (Bourechak et al., 2023). If we focus on the IoT-23 dataset and similar use scenarios, A silhouette score greater than 0.5 would be deemed reasonable and scores greater than 0.7 would suggest that there is low overlapping of clusters formed with high quality generated data.

3.5.3. Malware Identification

In this study, Deep Neural Networks (DNNs) were employed as the primary models for classifying the datasets, leveraging the unique strengths of each architecture to optimize classification performance. Specifically, three types of DNNs were utilized: Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory Networks (LSTMs).

CNNs were chosen due to their well-established ability to detect and learn intricate patterns within data, particularly excelling in identifying spatial hierarchies in images and similarly structured data formats. Beyond image processing, CNNs have proven highly effective in handling network traffic data represented in binary format, such as those derived from malware samples. Their inherent feature extraction capabilities enable the network to capture localized dependencies and

features, which contribute to achieving a relatively high level of classification accuracy in cybersecurity tasks (Mutambik, 2024). This makes CNNs suitable for scenarios where the data exhibits spatial correlations or structured patterns.

RNNs, on the other hand, were selected for their aptitude in modeling temporal or sequential dependencies within data. Unlike CNNs, RNNs are designed to process sequential inputs by maintaining a dynamic internal state that reflects previous inputs, allowing the network to capture the context and behavioral patterns over time. This characteristic is particularly advantageous when analyzing malware behavior that unfolds across a time sequence, providing insights into how certain malicious activities develop and persist (Alqahtani et al., 2023). By capturing these temporal relationships, RNNs enhance the model's ability to detect subtle anomalies or patterns indicative of malware.

Building upon the RNN architecture, LSTMs represent an advanced variant specifically designed to overcome the limitations of standard RNNs, such as the vanishing gradient problem. LSTMs incorporate memory cells and gate mechanisms that enable them to retain information over extended time intervals while selectively forgetting irrelevant data. This sophisticated design empowers LSTMs to identify dependencies between data points separated by longer time spans, making them particularly effective in processing sequential datasets with long-range correlations. In the context of IoT-23, which comprises sequential network traffic data, LSTMs are exceptionally well-suited due to their ability to recognize complex temporal patterns and correlations across time, which improve detection accuracy for cyber threats (Bensaoud & Kalita, 2024).

CHAPTER 4

RESULTS AND DISCUSSIONS

This chapter presents comprehensive evaluation results from the proposed CGAN-LSTM model for enhancing feature in IoT malware detection that has been carried out. The analysis on both real and generated data are carried out with classifier performance and quality assessment of the generated data using measures like silhouette score and PCA. Research models are compared with other feature enhancement models by metrics like accuracy, recall, precision, and F1-score. Limitations are also discussed which was observed during the development and testing process.

4.1. Experiment Environment

The test environment includes the hardware and software components used in this system. Details of the hardware and software specifications used during the software test can be found in Table 4.1.

Table 4.1 Experiment environment

Parts	Specification
Processor	Intel Core i5-12400F
Graphic Processing Unit	RTX 3060 12GB VRAM
Memory	32GB Dual Channel Memory
Storage	500GB NVMe SSD
Operating System	Windows 11
Environment	Python 3.10

4.2. Dataset

The IoT-23 dataset comprises of 23 distinct features, each representing a different data capture session. These scenarios are categorized into 20 malicious traffic captures and 3 benign, that reflect realistic network activities encountered in IoT deployments. The malicious captures cover a broad spectrum of attack types that pose significant threats to IoT security, including botnet communications, scanning activities, information theft, and command and control (C&C) communications used by attackers to remotely manage compromised devices.

The dataset includes traffic generated by various IoT gadgets, ranging from smart cameras to sensors, simulating practical attack scenarios across diverse device functionalities. The malicious traffic traces encompass multiple network protocols such as HTTP and DNS, among others, depending on the specific attack vectors employed. This diversity in protocol usage reflects the complex and diverse nature of contemporary cyber threats targeting IoT ecosystems.

In addition to its richness in attack diversity, the IoT-23 dataset provides detailed labeled annotations, that enables supervised learning approaches to classify and detect malicious behavior accurately. These labeled scenarios help the evaluation of advanced machine learning and deep learning models, including the CGAN-LSTM model investigated in this research, which relies on comprehensive and robust data representations to generate and detect attack patterns.

4.2.1. Dataset Features

Analyzing the features inside the IoT-23 dataset are important especially when determining which feature to include in the model. Not all features equally contribute to creating an effective and understandable machine learning model

Table 4.2 Features Contained in IoT-23 Dataset (Zeghida et al., 2024).

Feature Name	Description (Perbaiki Format Penulisan)
ts	Time Flow
id.orig_h	Source Address (IP)
duration	The Flow total duration
orig_bytes	number of payload from originator(in bytes)
resp_bytes	number of payload from responder(in bytes)
missed_bytes	the total missed bytes within a flow
orig_pkts	number of packets sent by originator
orig_ip_bytes	number of IP level sent by originator (in bytes)
resp_pkts	number of packets sent by responder
resp_ip_bytes	number of IP level sent by responder (in bytes)
proto_icmp	Transaction protocol ICMP
proto_tcp	Transaction protocol TCP

proto_udp	Transaction protocol UDP
conn_state_OTH	Connection state(no SYN)
conn_state_REJ	Connection state(attempt rejected)
conn_state_RSTO	Connection state*connection Established, originator aborted)
conn_state_RSTOS0	Connection state(originator send SYN followed by RST)
conn_state_RSTR	Connection state(responder sent RST)
conn_state_RSTRH	Connection state(responder send SYN followed by RST)
conn_state_S0	Connection state (connection attempt seen)
conn_state_S1	Connection state (connection established)
conn_state_S2	Connection state (connection established and closed by originator)
conn_state_S3	Connection state (connection established and closed by responder)
conn_state_SF	Connection state (normal establishment and termination)
conn_state_SH	Connection state (Originator sent a SYN followed by a FIN)
conn_state_SHR	Connection state (Responder sent a SYN ACK and FIN)
Label	Label (Malicious/Benign)

Analyzing the features inside the IoT-23 dataset are important especially when determining which feature to include in the model. Not all features equally contribute to creating an effective and understandable machine learning model.

Table 4.2 lists the overall features within the IoT-23 dataset. From the full list of features, this study selects the most informative and relevant ones for modeling network behavior, while discarding others to minimize noise, complexity, and redundancy. The selected features are as follows:

1. id.orig_p and id.resp_p: These represent the source and destination port numbers. Port numbers provide critical information about the communication endpoints and help in identifying the types of services and protocols involved. Since certain ports are commonly targeted or used in attacks, their inclusion helps the model learn patterns related to network service usage and potential malicious access points.

2. `duration`: Measures the total time length of each network flow. This feature is crucial because the length of a connection can reveal behavioral patterns; malicious traffic may exhibit extremely short or unusually long durations compared to normal flows.
3. `orig_bytes` and `resp_bytes`: These two features capture the amount of data, in bytes, sent by the originator and responder respectively. By considering both, this study evaluates the volume and directionality of communication, which helps distinguish between normal and anomalous traffic patterns such as data exfiltration or flooding.
4. `missed_bytes`: Represents bytes lost or missed during transmission, which may indicate network issues or deliberate evasion tactics in malicious traffic.
5. `orig_pkts` and `resp_pkts`: These indicate how many packets are sent by each side of the connection. Packet counts reflect the granularity of traffic exchanges, that complement byte-based measures and provide insights into packet-level behavior under different attack types.
6. `orig_ip_bytes` and `resp_ip_bytes`: These features extend the byte count to the IP layer, including payload and headers. They provide a fuller picture of data transmission size and network overhead, useful for spotting subtle anomalies.
7. `proto`: this indicates what protocol is being used to transmit the data. It can provide information about what protocol is used by the normal and malicious traffic.

From all the columns available in the IoT-23 dataset, this study carefully selects only the most meaningful features that capture the essence of network traffic behavior for the purpose of intrusion detection. The chosen features represent the fundamental quantitative characteristics of network flows, encompassing both the size and volume of data exchanged as well as the timing of communication sessions. These features provide crucial insights into the behavior of both parties in a network conversation: the originator and the responder. Selecting these specific metrics enables the model to detect anomalies in the flow duration and asymmetric patterns in bytes and packets sent or received, which are often indicative of malicious

activity such as data exfiltration, denial-of-service attacks, or command-and-control communications. The focus on these numerical, protocol-agnostic features also simplifies the feature space by avoiding more complex or categorical attributes like connection states or protocol flags, which may introduce noise and complicate interpretability. Ultimately, this targeted feature selection strikes a crucial balance between capturing meaningful behavioral patterns in network traffic and maintaining model efficiency, robustness, and transparency all essential for effective IoT intrusion detection systems operating on diverse and evolving traffic patterns.

4.2.2. Class Label

The IoT-23 dataset includes a diverse range of malware types and attack vectors targeting Internet of Things (IoT) devices, each exhibiting distinct malicious behaviors and impacts.

Table 4.3 Name and total of attacks inside IoT-23 dataset

No.	Type of Malware	Num of Attack
1	PartOfAHorizontalPortScan	825,933
2	Okiru	262,689
3	Benign	197,817
4	DDoS	138,776
5	C&C	15,103
6	Attack	3,915
7	benign	1,950
8	C&C-HeartBeat	350
9	C&C-FileDownload	43
10	C&C-Torii	30
11	FileDownload	14
12	C&C-HeartBeat-FileDownload	9
13	C&C-Mirai	1

From Table 4.3 can be explained below:

1. PartOfAHorizontalPortScan (825,933 attacks): This is the most frequent malware-like behavior recorded in the dataset. Horizontal port scanning is a

reconnaissance technique used by attackers to identify open ports across multiple devices within a network. In the context of IoT, such scans can reveal vulnerable devices or services that can be exploited. Although not inherently malicious on its own, persistent port scanning often precedes more targeted attacks such as exploitation or intrusion.

2. Okiru (262,689 attacks): Okiru is a Trojan malware typically designed to compromise IoT devices by exploiting known vulnerabilities. It often establishes persistence on the device and connects it to a command and control (C&C) infrastructure for further exploitation or launching distributed attacks.
3. Benign (197,817 + 1,950 attacks): These entries represent network traffic linked to normal, non-malicious activity of IoT devices. Such data serves as the baseline for distinguishing legitimate operations from malicious activities.
4. DDoS (138,776 attacks): Distributed Denial of Service (DDoS) attacks overwhelm IoT devices or their network infrastructure with massive traffic to cause service outages or degrade performance. Due to the resource constraints of many IoT devices, they are particularly susceptible to disruption from DDoS attacks.
5. C&C (Command and Control) Variants (Total 15,539 across several types): This includes generic C&C, HeartBeat signals, FileDownload commands, Torii variant, and Mirai. C&C malware enables attackers to remotely control compromised IoT devices, often to create coordinated attacks, update malware payloads, or steal data.
6. Attack (3,915 attacks): This label may refer to generic or unspecified attack instances captured within the dataset that do not fall into the above categories but indicate suspicious or malicious activity.

C&C datasets are combined because of the nature of the dataset. Some C&C variants have so little data, so it has to be combined with the main dataset inside the same class.

4.3. Data Preprocessing

The preprocessing procedure begins with addressing placeholder values represented by the symbol ‘-’, converted to NaN (Not a Number). This transformation is essential to standardize the treatment of missing or undefined entries, facilitating consistent handling during subsequent data cleaning.

Missing values denoted by NaN are substituted differently based on the feature type to maintain data integrity. Numerical columns are assigned to a value of 0, representing a neutral baseline or absence of activity. Categorical columns receive the value ‘Unknown’ to explicitly encode the absence of category information, enabling models to distinguish missing categories without introducing bias.

The normalization of the class label column is one of the important step. This involves the removal of unique id and inconsistencies (such as (empty), -, or repeated terms like “Malicious”) to unify labels into coherent categories. The label naming is standardized programmatically to ensure consistent class representation, as demonstrated by the replacement of lowercase “benign” and empty strings with the capitalized label “Benign”. This process prevents fragmentation of classes that could otherwise hinder classification accuracy and clarity.

In accordance with dataset provenance guidance, multiple related subclasses within the Command and Control (C&C) category specifically ‘C&CHearBeat’, ‘C&CTorii’, ‘C&CFileDownload’, ‘C&CHearBeatFileDownload’, and ‘C&CMirai’ are merged into a single big ‘C&C’ class. This join process aligns labeling with domain knowledge and simplifies model training by consolidating conceptually similar instances.

The dataset is sorted based on the ‘ts’ column to maintain the sequence nature of the dataset. This sorting verifies that the progression of network events follows the exact timeline in which they occurred, keeping the chronological context critical for effective modeling. By arranging the data chronologically, the temporal dynamics of network traffic, including the progression and patterns of normal and anomalous events are maintained. This chronological ordering allows the LSTM component to learn temporal correlations and sequence dependencies

accurately, while the CGAN leverages this sequence-aware input to synthesize realistic data that mirrors time-evolving behaviors.

The feature selection narrows down to a targeted set of relevant quantitative attributes that are crucial for capturing distinctive network flow behavior. These include ‘id.orig_p’ and ‘id.resp_p’, which represent the originator and responder port numbers, respectively, giving insight into communication channels used. The ‘duration’ feature records the total connection time, providing temporal context of the flow. Byte counts are covered by ‘orig_bytes’ and ‘resp_bytes’, indicating the volume of data sent and received, while ‘missed_bytes’ captures anomalous or missing data in transmission streams, potentially signaling network irregularities or attacks. Packet counts such as ‘orig_pkts’ and ‘resp_pkts’ quantify the number of packets exchanged, and IP-layer byte counts ‘orig_ip_bytes’ and ‘resp_ip_bytes’ offer a deeper measure of the data load at the IP protocol level.

To ensure feature values are on comparable scales and suitable for model input, all features undergo Min-Max scaler process to normalize the numerical values into a uniform range between -1 and +1. This process prevents dominance of features with large magnitude and stabilizes optimization during training. The target classes are then converted through one-hot encoding into seven distinct binary vectors, that makes the categorical labels can be processed by machine learning algorithms. Table 4.4 presents a sample of the original raw dataset prior to preprocessing, while Table 4.5 illustrates the dataset after undergoing systematic preprocessing transformations.

Table 4.4 Original Dataset

ts	uid	id.orig_h	id.orig_p
1.53E+09	CwxSC...	192.168.100.103	43763
1.53E+09	C3GBT...	192.168.100.103	41101
1.53E+09	CC6vK...	192.168.100.103	43763
1.53E+09	CDe43c...	192.168.100.103	60905
1.53E+09	CJaDcG...	192.168.100.103	44301

id.resp_h	id.resp_p	proto	service
154.8.94.65	14336	udp	Unknown
111.40.23.49	23	tcp	Unknown
196.170.198.141	11764	udp	Unknown
131.174.215.147	23	tcp	Unknown
91.42.47.63	23	tcp	Unknown
Duration	orig_bytes	resp_bytes	conn_state
0	0	0	S0
0	0	0	S0
0	0	0	S0
2.998796	0	0	S0
0	0	0	S0
local_orig	local_resp	missed_bytes	history
Unknown	Unknown	0	D
Unknown	Unknown	0	S
Unknown	Unknown	0	D
Unknown	Unknown	0	S
Unknown	Unknown	0	S
orig_pkts	orig_ip_bytes	resp_pkts	resp_ip_bytes
1	40	0	0
1	60	0	0
1	40	0	0
3	180	0	0
1	60	0	0

Table 4.5 Dataset after Preprocessing

id.orig_p	id.resp_p	duration	orig_bytes		
0.335561151	-0.562493324	-1	-1		
0.254322118	-0.999298085	-1	-1		
0.335561151	-0.640985733	-1	-1		
0.858701457	-0.999298085	-0.999923927	-1		
0.351979858	-0.999298085	-1	-1		
resp_bytes	missed_bytes	orig_pkts	orig_ip_bytes		
-1	-1	-0.999999997	-0.9999999958		
-1	-1	-0.999999997	-0.9999999937		
-1	-1	-0.999999997	-0.9999999958		
-1	-1	-0.9999999909	-0.9999999812		
-1	-1	-0.999999997	-0.9999999937		
resp_pkts	resp_ip_bytes	proto_icmp	proto_tcp	proto_udp	
-1	-1	-1	-1	1	
-1	-1	-1	1	-1	
-1	-1	-1	-1	1	
-1	-1	-1	1	-1	
-1	-1	-1	-1	1	

4.4. Performance Behavior on Original and Feature Enhanced Datasets

In this study, both the original and enhanced datasets undergo rigorous evaluation using various neural network classifiers, specifically Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) networks. The performance of these classifiers on the original dataset establishes a critical baseline, serving to highlight inherent limitations such as data imbalance, noise, and insufficient feature representation that could hinder the models' predictive capabilities. Imbalanced datasets often cause classifiers to be biased toward majority classes, while noisy data and inadequate feature representation can further degrade performance by confusing the learning process. By benchmarking against these baseline results, the study exposes

the challenges faced by traditional approaches when handling real-world IoT traffic data.

Subsequently, the enhanced dataset by synthetic feature generation through the CGAN-LSTM model is evaluated using the same classifiers to assess improvements in detection performance. Key metrics such as accuracy, recall, precision, and F1-score are carefully observed as indicators of how well the models detect malware, especially in minority classes that typically suffer from poor representation in original datasets. The use of CGAN combined with LSTM aims to generate realistic, temporally consistent synthetic data that enriches the feature set, by doing so handling the class imbalance and improving classifier robustness. Furthermore, the stability and training dynamics of each model are assessed by analyzing loss curves during training epochs. This comprehensive evaluation helps determine whether the CGAN-LSTM enhancement can enhance classifier performance and generalization in identifying diverse malware behaviors within IoT network data. The additional features incorporated into the dataset through augmentation are summarized in Table 4.6. It provides a clear overview of the expanded feature space.

Table 4.6 Generated CGAN Features

f14	f15	f16	f17
-0.46124	0.228559	0.146804	-0.17757
0.002825	0.263761	0.472883	-0.09463
-0.37858	0.246763	0.173844	-0.21616
-0.04649	0.251798	0.496857	-0.12977
-0.01084	0.25594	0.471659	-0.09308

4.4.1. Performance on Original Dataset

The initial experiment was conducted using the original IoT-23 dataset to evaluate the classification capabilities of three deep learning models: CNN, RNN, and LSTM. The classification performance metrics, summarized in Table 4.7, Table 4.8, and Table 4.9, demonstrate generally strong results with high overall accuracy across the models. A deeper analysis reveals critical shortcomings in detecting

certain classes, as evidenced by extremely low or zero values in precision, recall, and F1-score for specific minority categories.

Table 4.7 CNN Classification Report (Original Dataset)

Label	P. (%)	R. (%)	F1. (%)	Support
Attack	0	0	0	783
Benign	96.83	40.18	56.79	39953
C&C	98.47	43.64	60.48	3107
DDoS	98.42	99.83	99.12	27755
FileDownload	0	0	0	3
Okiru	99.93	100	99.96	52538
PortScan	86.23	99.52	92.40	165187

The CNN classifier's performance on the original IoT-23 dataset is initially assessed through the classification report that can be seen on Table 4.7, which reveals notable variability across different attack categories. The model achieves excellent precision and recall for certain classes such as DDoS (Precision: 98.42%, Recall: 99.83%, F1-score: 99.12%) and Okiru (Precision: 99.93%, Recall: 100%, F1-score: 99.96%), indicating its strong capability to correctly identify these attack types. The 'PortScan' class shows a high recall of 99.52% and an F1-score of 92.40%, although precision is somewhat lower at 86.23%, suggesting occasional false positives. The model struggles with the 'Attack' and 'FileDownload' classes, both showing zero precision, recall, and F1-scores, which points to complete failure in detecting these categories. 'Benign' and 'C&C' classes reflect moderate performance with precision near 96.83% and 98.47% respectively, but low recall (40.18% and 43.64%), indicating the model misses a large portion of these samples. This performance distribution is confirmed by the confusion matrix that can be seen on Figure 4.1, which visually reveals this classification distribution.

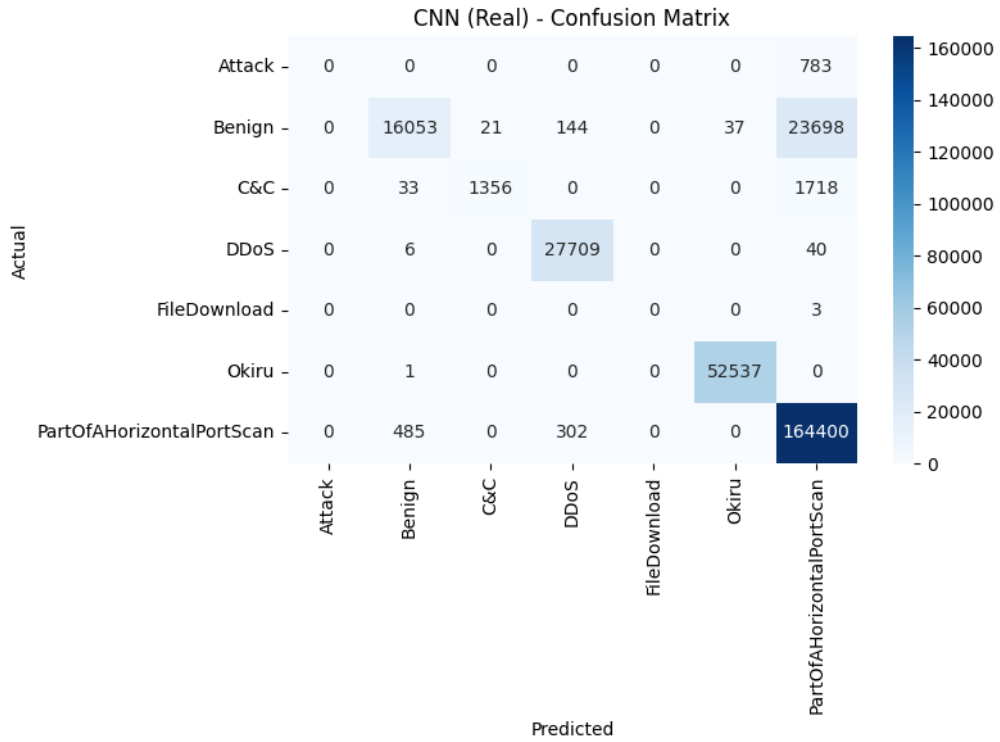


Figure 4.1 Confusion matrix of CNN model on Original Dataset

In the confusion matrix a great performance can be seen for DDoS and Okiru class which has 27,709 TP, 52,537 TP respectively. Despite the great performance from those classes, this matrix also shows a substantial number of misclassifications. The ‘Attack’ class has 783 data points, had no TPs but was entirely misclassified as ‘PortScan’ (783 FN). The benign class missed 23,698 samples that were incorrectly predicted as ‘PortScan’ and had additional false negatives incorrectly labeled as C&C and DDoS. The ‘PortScan’ class had 485 samples misclassified as benign and 302 as DDoS, contributing to FP for those classes. This distribution of FN and FP across key classes aligns with the recall and precision values seen in the classification report, confirming the model’s struggle to differentiate between certain overlapping traffic patterns, especially misclassifying attacks as port scans or benign traffic. The confusion matrix also shows minimal errors for classes like ‘DDoS’ and ‘Okiru’, supporting their high classification metrics.

Table 4.8 RNN Classification Report (Original Dataset)

Label	P. (%)	R. (%)	F1. (%)	Support
Attack	0	0	0	783
Benign	90.10	42.09	58.91	39953
C&C	87.37	98.39	92.55	3107
DDoS	98.53	99.83	99.18	27755
FileDownload	0	0	0	3
Okiru	99.91	100	99.95	52538
PortScan	87.61	99.7	93.27	165187

The result of RNN model in Table 4.8 shows a similar result to CNN where it is great at detecting classes such as ‘DDoS’ and ‘Okiru’ but shows poor performance in other classes like failing to detect the ‘Attack’ and ‘FileDownload’ classes. For the ‘Benign’ class, the RNN shows relatively high precision (90.10%) but low recall (42.09%), indicating it predicts benign instances with reasonable accuracy but misses many true benign samples, lowering overall detection effectiveness for this class. In contrast, the RNN classification report are improved when detecting ‘C&C’ attacks compared to CNN, with a recall of 98.39% and F1-score of 92.55%, showing strong recognition capability. The performance on dominant attack classes such as ‘DDoS’ and ‘Okiru’ remains robust, with near-perfect recall and F1-scores exceeding 99%, reflecting high sensitivity and accuracy. The ‘PortScan’ class also demonstrates a good balance, with precision at 87.61%, recall at 99.7%, and an F1-score of 93.27%, although some false positive predictions are present.

The result of the classification report is proved further with the visualization of predicted and actual class inside confusion matrix in Figure 4.2. Misclassification is a common occurrence across all classes. True Positives dominate along the diagonal for classes like ‘DDoS’ (27,709), ‘Okiru’ (52,537), ‘C&C’ (3,057), and ‘PortScan’ (164,696), confirming effective detection for these categories. However, the matrix also exposes significant misclassifications: all 783 samples of the ‘Attack’ class were incorrectly labeled as ‘PortScan’, aligning with the zero-

performance metrics from the classification report. The ‘Benign’ class confuses the model, with 22,424 benign samples misclassified as ‘PortScan’, 441 as ‘C&C’, and 221 as ‘DDoS’, explaining the low recall. False positives are high in the ‘PortScan’ category due to 300 benign and 191 DDoS samples being predicted as ‘PortScan’.

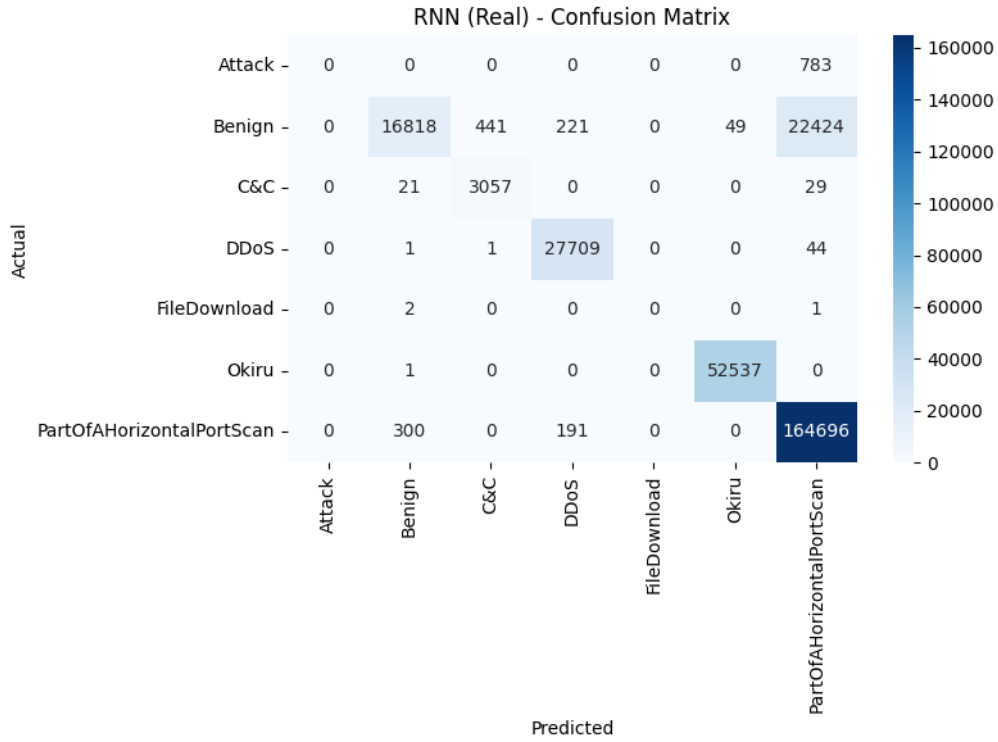


Figure 4.2 Confusion matrix of RNN model on Original Dataset

The LSTM classifier provides a comparable performance to the RNN model showing improvements in certain classes but poor in others that can be seen on Table 4.9. The LSTM fails to detect the rare ‘Attack’ and ‘FileDownload’ classes, similar to previous models. The ‘Benign’ class shows moderate performance with a precision of 96.09%, a recall of 40.19%, and an F1-score of 56.67%, indicating the classifier is relatively accurate when labeling benign traffic but still misses numerous instances (false negatives). The ‘C&C’ class sees marginal improvement in precision (97.91%) but retains a low recall (43.64%), suggesting more false negatives compared to its precision. Major attack categories like ‘DDoS’ and ‘Okiru’ maintain strong performance, with recall near or at 100% and F1-scores nearing 99%, confirming the LSTM robustness to these prevalent threats. The

‘PortScan’ class also performs well, with an F1-score of 92.44% supported by high recall (99.48%) and solid precision (86.33%), despite some classification noise.

Table 4.9 LSTM Classification Report (Original Dataset)

Label	P. (%)	R. (%)	F1. (%)	Support
Attack	0	0	0	783
Benign	96.09	40.19	56.67	39953
C&C	97.91	43.64	60.37	3107
DDoS	97.94	99.83	98.88	27755
FileDownload	0	0	0	3
Okiru	99.9	100	99.95	52538
PortScan	86.33	99.48	92.44	165187

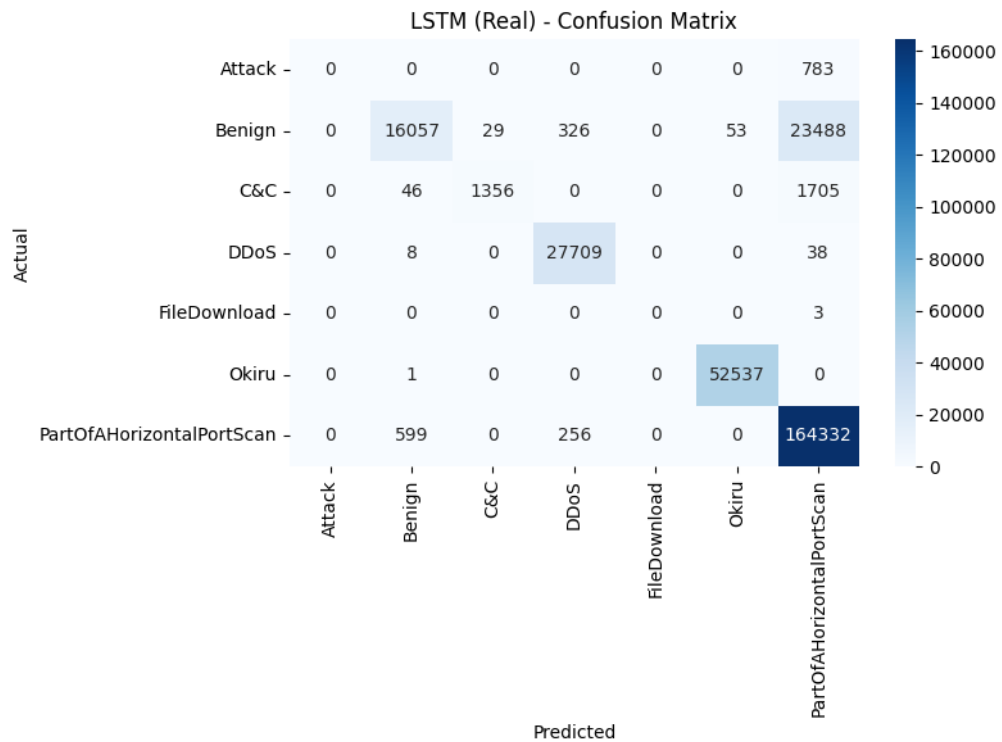


Figure 4.3 Confusion matrix of LSTM model on Original Dataset

The confusion matrix on Figure 4.3 is the representation of the LSTM model on the original dataset. The diagonal highlights correct identifications for critical classes, such as 27,709 ‘DDoS’, 52,537 ‘Okiru’, 1,356 ‘C&C’, and 164,332 ‘PortScan’ samples. However, the rare ‘Attack’ and ‘FileDownload’ classes remain

undetected, with all 783 and 3 instances, respectively, misclassified predominantly as ‘PortScan’. The ‘Benign’ class again confuses the model, where 23,488 ‘Benign’ samples are wrongly predicted as ‘PortScan’, 326 as ‘DDoS’, and 29 as ‘C&C’, explaining the low recall. There are also notable false negative contributions in the ‘PortScan’ category, including ‘599’ benign and 256 ‘DDoS’ samples mistakenly categorized, suggesting some difficulty in clearly discriminating against these classes.

This failure confirms the findings in the literature, which stated that imbalanced datasets skew model learning towards majority classes, resulting in biased decision boundaries and poor minority class detection (Alfares & Banimelhem, 2024). The original IoT-23 dataset’s imbalance ratio spans around 59,000:1 down to more balanced classes, directly correspond to poor model performance.

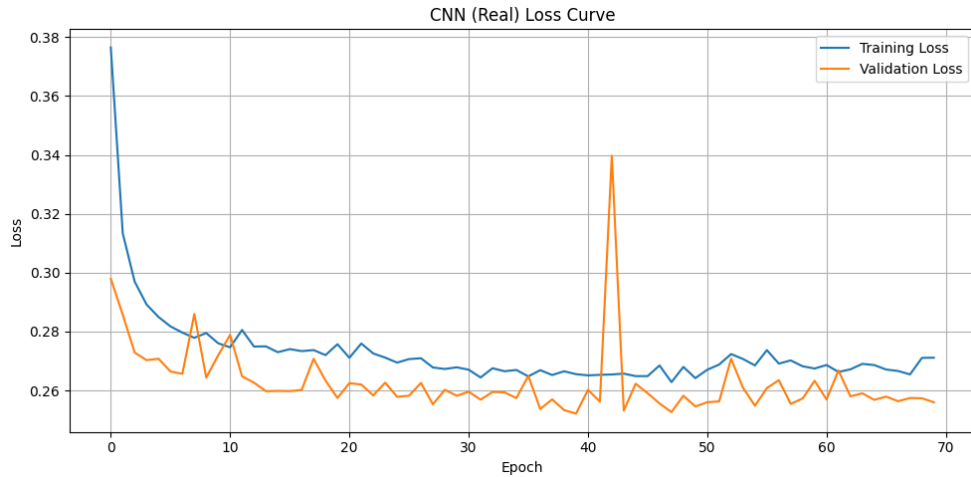


Figure 4.4 Loss Curve of CNN model on Original Dataset

The CNN model’s loss curve Figure 4.4 exhibits relatively stable learning dynamics through the epochs. The training loss steadily decreases and approaches a plateau, while the validation loss is generally lower and closely follows the training curve, indicating good generalization. However, a slight increase in validation loss near the middle epoch suggests minor overfitting, where the model begins to fit noise or specific training details that do not generalize well. This overfitting can limit the model’s ability to classify samples accurately beyond the training set.

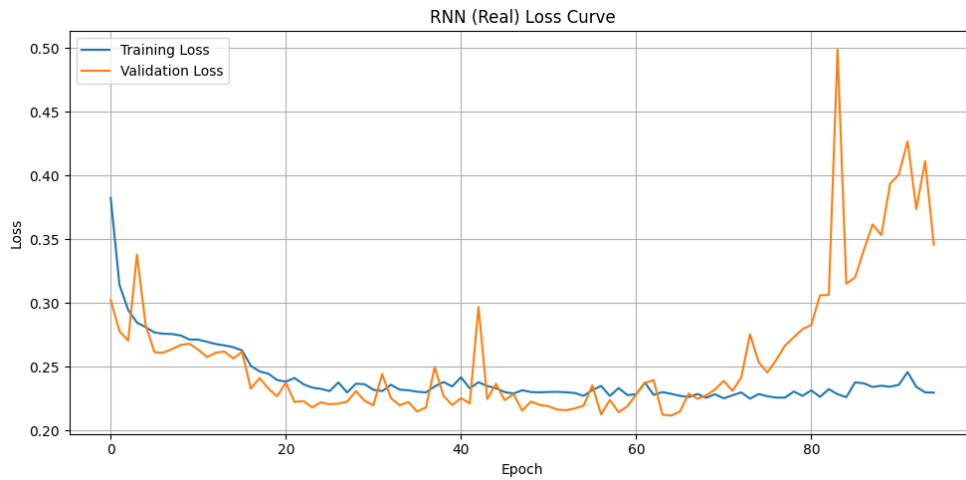


Figure 4.5 Loss Curve of RNN model on Original Dataset

The RNN’s loss curve Figure 4.5 highlights a more pronounced overfitting tendency. Initial epochs show declining training and validation loss, but from about epoch 60 onward, the validation loss rises sharply while the training loss remains low and flat. This divergence implies that while the model fits the training data well, it struggles to generalize. It results in the overfitting because of the noise and outliers.

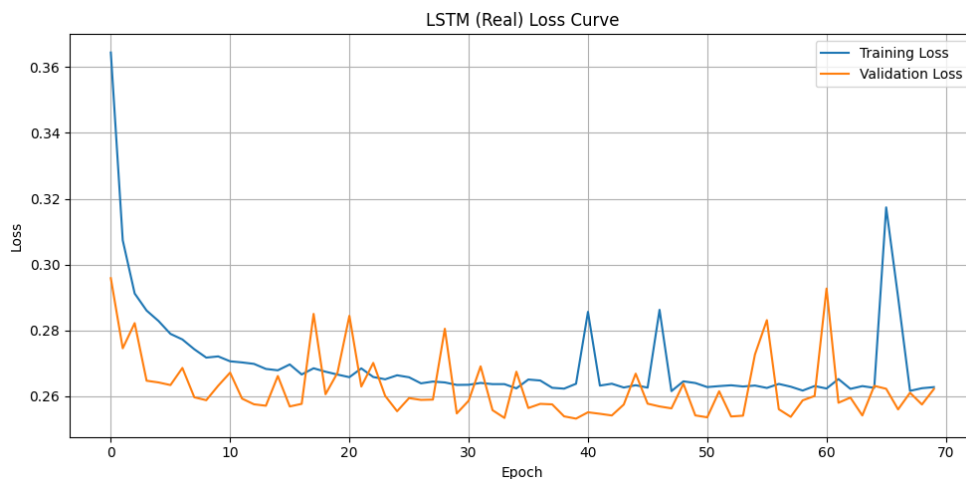


Figure 4.6 Loss Curve of LSTM model on Original Dataset

The LSTM model shows less overfitting compared to the RNN, with training and validation losses progressing more concordantly. There are minor oscillations present, typical in sequential models fine-tuning their parameters, yet both losses stay close, showing stable and effective learning. Despite this, there is

slight variability in the validation loss. It can be a signals that the dataset is less regularized or it needs data balancing to further enhance generalizability.

4.4.2. Performance on FE Dataset

CGAN-LSTM model was implemented to generate synthetic features conditioned on class labels. These synthetically generated features were then integrated into the original dataset, producing a more expressive and balanced feature representation for each class. This augmentation method addresses the inherent class imbalance issues present in the original data, which is evident from the classification metrics. Table 4.10, Table 4.11, and Table 4.12 report the classification results when the models are trained on the augmented dataset enriched with CGAN-LSTM synthetic features.

Table 4.10 CNN Classification Report (FE Dataset)

Class	P. (%)	R. (%)	F1. (%)	Support
Attack	100	100	100	783
Benign	99.98	100	99.99	39953
C&C	100	99.87	99.94	3107
DDoS	100	99.98	99.99	27755
FileDownload	100	100	100	3
Okiru	100	100	100	52538
PortScan	100	100	100	165187

The CNN classifier's performance shows a remarkable improvement after applying CGAN-LSTM feature enhancement on the IoT-23 dataset. According to the classification report, precision, recall, and F1-scores for all classes have achieved near-perfect or perfect values, specifically 100% for classes including 'Attack', 'Benign', 'C&C', 'DDoS', 'FileDownload', 'Okiru', and 'PortScan'. This reflects the significant positive impact of the feature enhancement in enabling the CNN model to distinguish between different types of network traffic or attack categories with extremely high accuracy. Classes like 'FileDownload' and 'Attack' which had 0 scores previously, now demonstrate perfect detection capability,

showing that the synthetic data and enhanced features addressed previous class imbalance or insufficient feature representation.

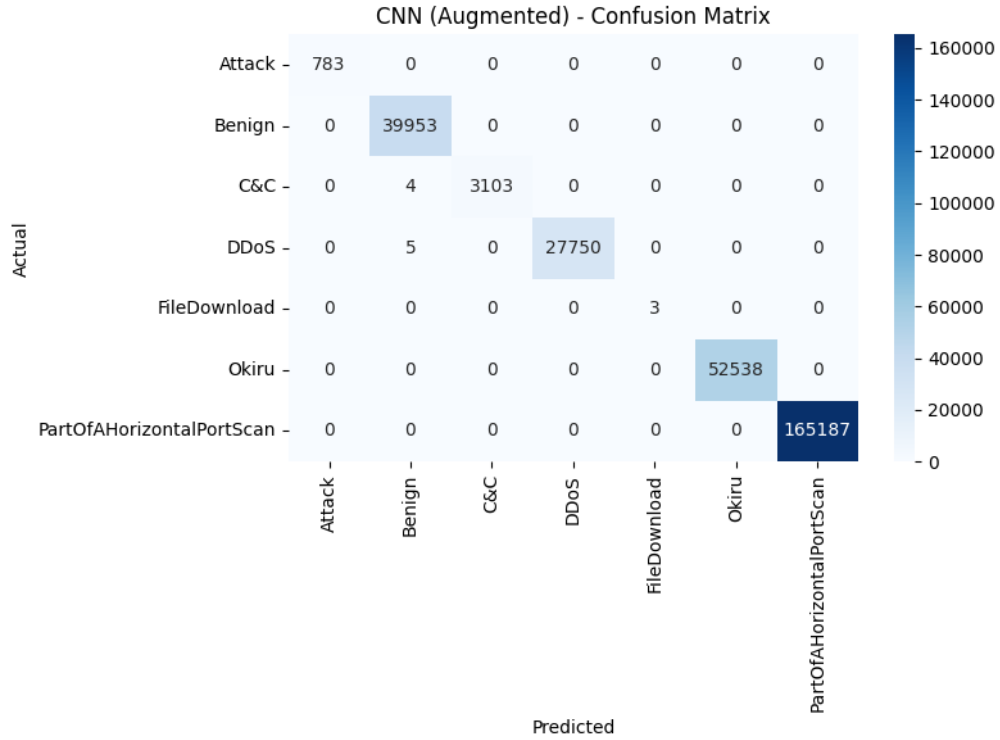


Figure 4.7 Confusion matrix of CNN model on FE Dataset

The confusion matrix for the CNN model on the CGAN-LSTM feature-enhanced dataset reveals a strong and clear dominance along its diagonal, which corresponds precisely to the true positives for each respective class. The 'Attack' class shows an exact count of 783 instances correctly identified as attacks. This number perfectly matches the actual number of attack records present in the dataset, signifying that the model did not miss any attack instance. The 'Benign' class exhibits a true positive count of 39,953, again exactly matching the total benign instances in the dataset. This confirms that all benign traffic was properly recognized, avoiding any misclassification as malicious or other types of network traffic. The "Okiru" malware class, with 52,538 true positives, the model successfully identified all occurrences of this attack type. Each class listed in the confusion matrix follows this strong diagonal pattern, showing that the model's predictions are largely in perfect alignment with the true labels, indicating near-zero false negatives and false positives overall.

Table 4.11 RNN Classification Report (FE Dataset)

Class	P. (%)	R. (%)	F1. (%)	Support
Attack	100	99.99	100	783
Benign	99.98	99.99	99.99	39953
C&C	99.84	99.97	99.90	3107
DDoS	100	99.98	99.99	27755
FileDownload	0	0	0	3
Okiru	100	100	100	52538
PortScan	100	100	100	165187

Classification report of RNN classifier (Table 4.10) shows increased performance across almost all classes on the enhanced feature dataset compared to the original dataset. It shows that CGAN-LSTM feature enhancement can improve the classification accuracy. Major classes such as ‘Attack’, ‘Benign’, ‘C&C’, ‘DDoS’, ‘Okiru’, and ‘PortScan’ achieves precision, recall, and F1 scores approaching or reaching 100%. These results indicate the model can identify and classify these networks traffic types and attack categories with minimal error. Enhanced dataset as a result from feature enhancement model can show a much more generalized feature than original dataset. The ‘Attack’ class is classified with 100% precision and recall, resulting in a perfect F1 score of 100%, signifying that the model successfully identifies all attack instances without misclassifying benign ones as attacks, or vice versa. The ‘Benign’ class similarly shows a precision of 99.98%, recall of 99.99%, and an F1 score of 99.99%, demonstrating strong reliability in distinguishing normal from malicious traffic. The high precision and recall for ‘C&C’, ‘DDoS’, ‘Okiru’, and ‘PortScan’ further underscore the model’s impressive detection capabilities. Regardless of the strong result, RNN fails to detect ‘FileDownload’ class due to limited number of sample.

To further analyze the classification report, the classification result in details can be seen in the confusion matrix in Figure 4.8.

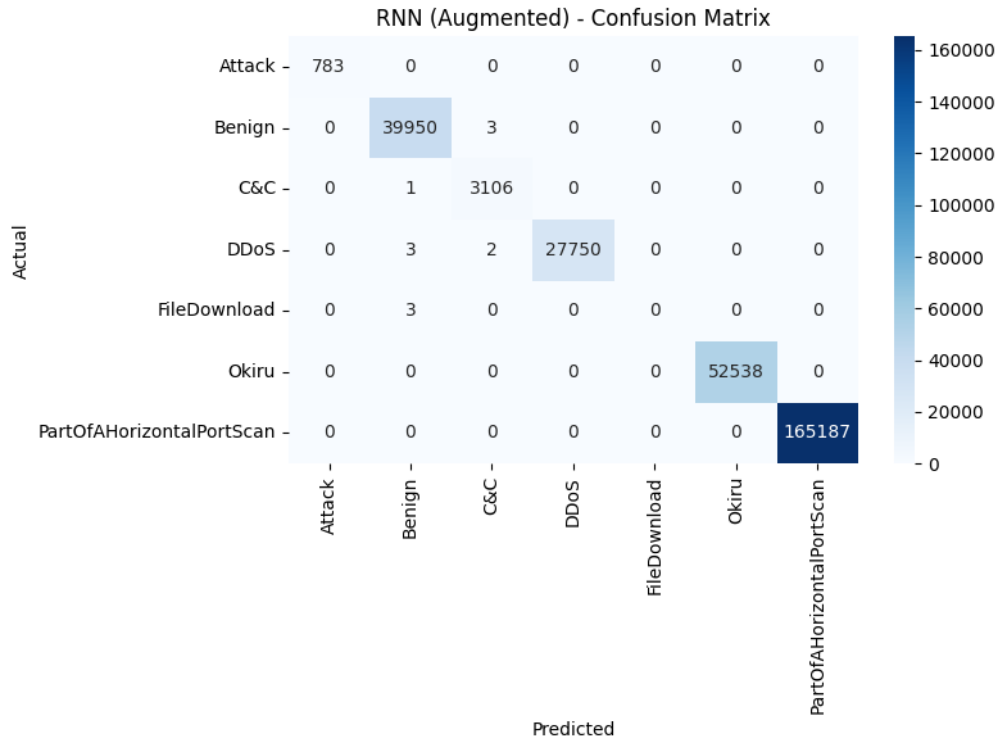


Figure 4.8 Confusion matrix of RNN model on FE Dataset

The classification report of RNN is further analyzed with confusion matrix. It shows that all classes can be well classified in each their respective class. The model correctly classifies 738 data points in the ‘Attack’ class. The ‘Benign’ class shows great performance where the model only misclassified 3 data points into ‘C&C’ class. Other classes like ‘C&C’, ‘DDoS’, ‘Okiru’, and ‘PortScan’ also exhibit very high numbers of true positives, confirming that the model able to distinguishes these classes in the enhanced feature space. The ‘FileDownload’ class is an exception because none of the three true instances of ‘FileDownload’ are correctly identified where all are misclassified as Benign traffic. There are some confusions of the model when handling ‘Benign’ and ‘FileDownload’ traffic where the model had trouble differentiating the normal and malicious traffic. The confusion matrix confirmed the RNN model’s strong classification following the CGAN-LSTM feature enhancement, with a high accuracy in detecting multiple attack types and benign traffic.

Table 4.12 LSTM Classification Report (FE Dataset)

Class	P. (%)	R. (%)	F1. (%)	Support
Attack	100	100	100	783
Benign	99.98	100	99.99	39953
C&C	99.97	99.87	99.92	3107
DDoS	100	99.98	99.99	27755
FileDownload	100	100	100	3
Okiru	100	100	100	52538
PortScan	100	100	100	165187

The LSTM model in Table 4.12 demonstrates remarkably strong performance on the enhanced dataset, achieving high accuracy and robustness across all key evaluation metrics. This is evident in its ability to reach precision, recall, and F1 scores approaching 99.99% or even a perfect 100% in most classes, which underscores its effectiveness in classifying diverse network traffic types with minimal error. One of the crucial factors behind this strong performance is the inherent design of the LSTM architecture, which makes it especially suited to handle the temporal and spatial characteristics present in the dataset. Unlike other models, the LSTM is built to capture sequential dependencies and temporal patterns in data, enabling it to understand the complex time-series behavior of network traffic and cyber-attacks. This temporal sensitivity allows the model to maintain contextual information over periods, thus enhancing its discriminatory power between benign and malicious activities.

Figure 4.9 LSTM confusion matrix further confirms the classification report claims where only several data points are misclassified. The matrix shows that every one of the 783 samples labeled as ‘Attack’ is correctly identified, with zero misclassifications. In the ‘Benign’ class, the 39,953 instances correctly classified, highlighting the model’s strong ability to distinguish normal network traffic from malicious activity.

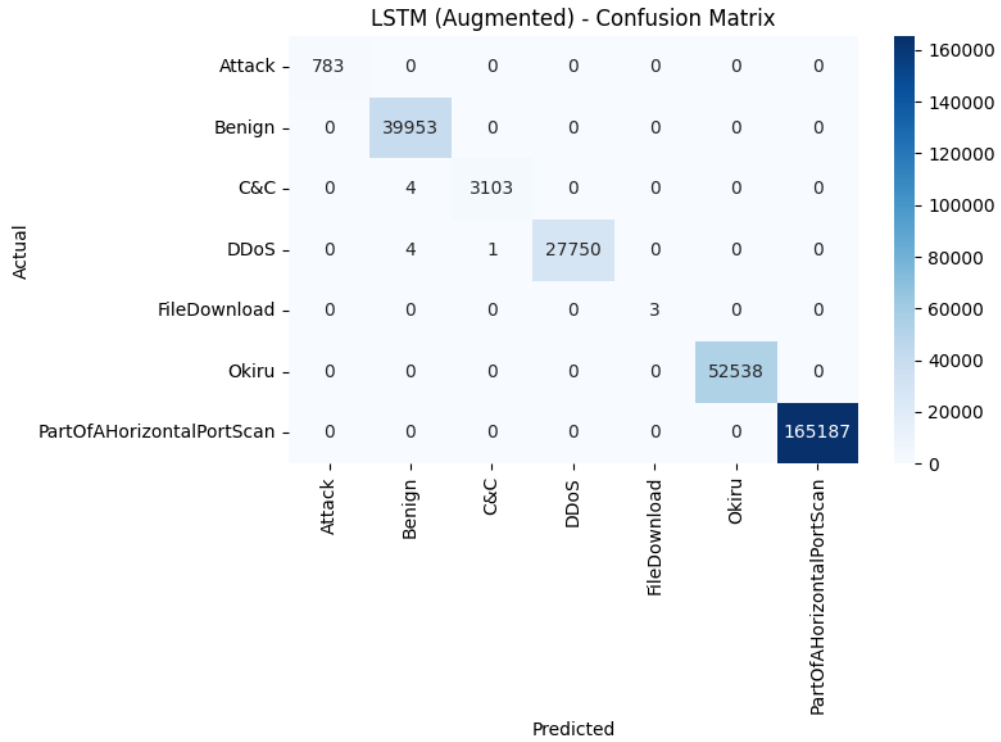


Figure 4.9 Confusion matrix of LSTM model on FE dataset

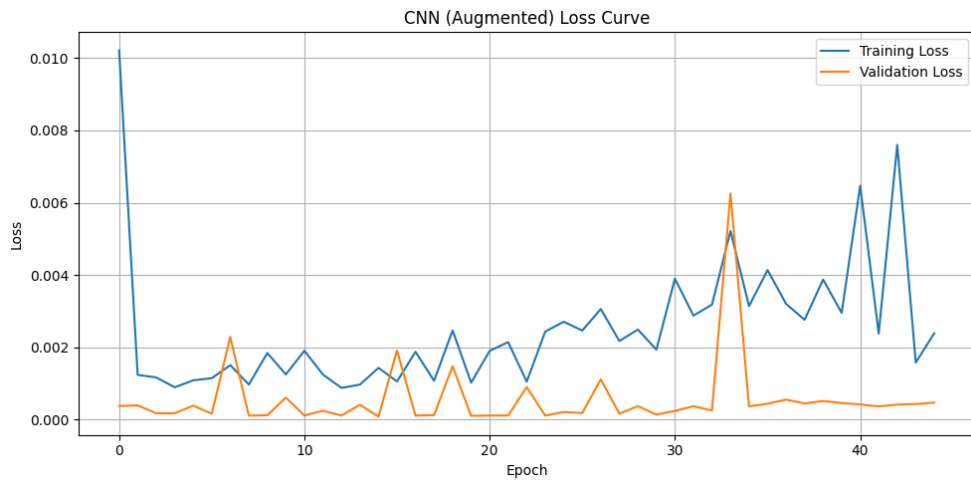


Figure 4.10 Loss curve of CNN model on Enhanced Dataset

Figure 4.10, Figure 4.12, Figure 4.11, presents the loss curves for the Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM) models respectively which evaluated on the feature enhanced dataset. These loss graphs provide critical insights into the

learning dynamics, model stability, and the underlying quality of the synthetic data produced by the CGAN-LSTM augmentation process.

The CNN model in Figure 4.10 demonstrates an initially steep decline in training loss within the first few epochs, indicating effective early learning and rapid convergence. Despite this, the training loss exhibits oscillations after the initial phase, with fluctuations that suggest some instability in fitting, potentially caused by variations in the enhanced data's distribution or overfitting to certain data patterns. The validation loss remains consistently low and relatively smooth throughout training. It shows that the CNN model generalizes reasonably well to unseen data, reaffirming the enhanced dataset's capacity to maintain the structural integrity and variability needed for robust model training.

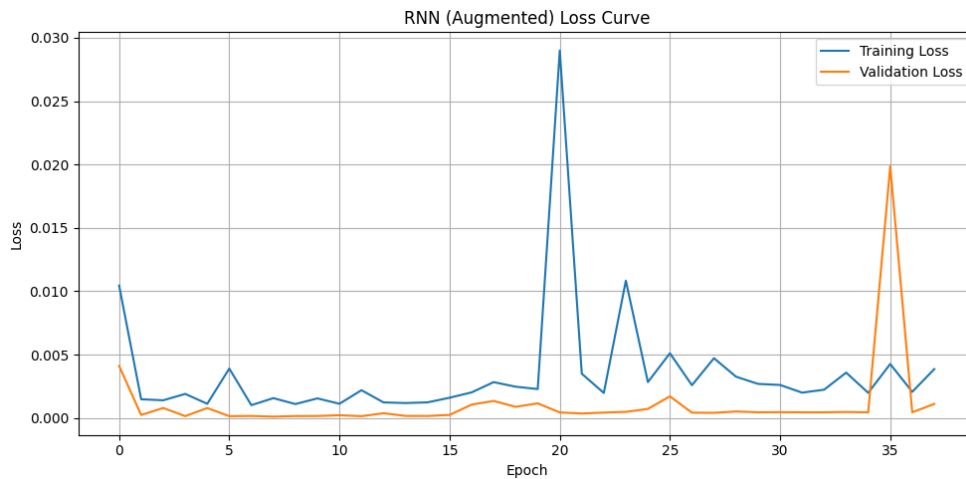


Figure 4.11 Loss curve of RNN model on Enhanced Dataset

The RNN's loss curve in Figure 4.11 indicates more volatility in training. The training loss shows pronounced spikes, particularly near epochs 20 and 35, suggesting periods where the model struggles to fit the data or experiences disruptions in the optimization trajectory. The validation loss curve remains fairly flat and low but includes a sharp peak around epoch 35, potentially reflecting moments where the model overfits transient artifacts or noise introduced during augmentation. This instability may highlight sensitivity in RNN architectures to subtle distributional shifts in augmented sequential data or suggest that further tuning of augmentation parameters is necessary to improve data fidelity and model robustness.

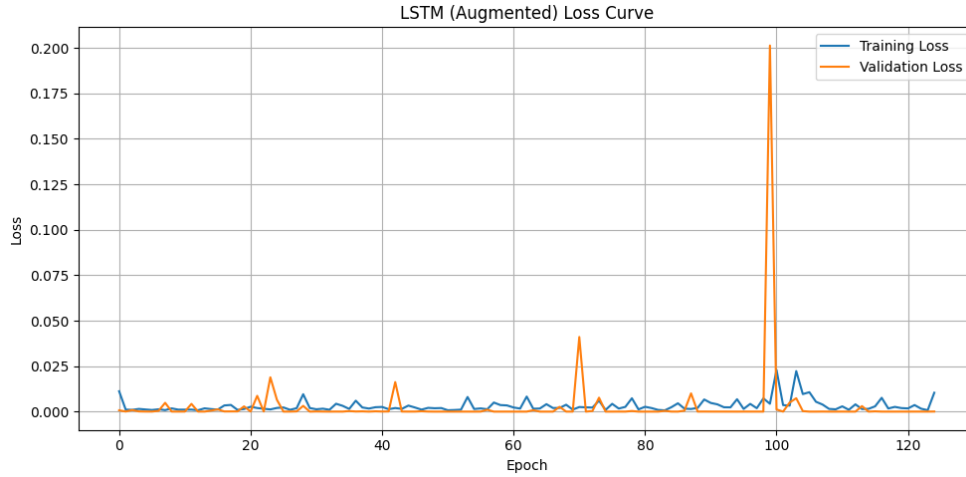


Figure 4.12 Loss curve of LSTM model on Enhanced Dataset

The LSTM model in Figure 4.12 exhibits the most stable loss progression among the three models. Both training and validation losses remain consistently near zero for the majority of the epochs, indicating strong learning capability and generalization. However, a spike in validation loss around epoch 100 signals a possible overfitting event or outlier behavior not reflected in training, underscoring the need for vigilant monitoring and potentially early stopping or regularization to prevent degradation in model performance. The overall low loss values across training and validation phases suggest that the LSTM, given its architectural design tailored for sequential dependencies, can leverage the augmented dataset for learning long-term temporal patterns with minimal overfitting, further evidence of the quality of the generated synthetic data.

In summary, the comparative loss analysis highlights that dataset quality and structure significantly influence model training dynamics. The marked stability in the LSTM model's loss curve validates the effectiveness of the CGAN-LSTM augmentation approach in preserving meaningful temporal characteristics needed for sequential classification tasks. On the other hand, the fluctuations in CNN and particularly RNN losses point towards areas where augmentation refinement or model hyperparameter tuning could further optimize generalization and performance. These insights are invaluable not only for interpreting the augmented data's realism and utility but also for guiding iterative improvements in data

generation and deep learning model selection strategies in IoT network security contexts.

The CGAN-LSTM model is designed to enhance the feature and temporal pattern of the data by giving a richer feature to represent a class. Aside from improving, this model can also give latent inconsistencies, which reduce the overall performance of the classifier. It can be shown on the RNN model where the loss curve shows sharp spikes and unstable convergence, which may be caused by noise and temporal inconsistencies inside the enhanced feature. On the other hand, CNN shows moderate instability, which suggests the model cannot capture the spatial or local features provided in the synthetic dataset, or the dataset may present data that cannot represent each class. The best result of the classifier is LSTM, which shows a more stable loss curve. Both training and validation losses are low and closely aligned. The LSTM model can benefit from its capability to extract temporal continuity and sequential dependencies. It is suggested that the generated data can appear structurally meaningful to this model.

Table 4.13 Macro and weighted average metrics across all classifier models

Model	With FE	A. (%)	P. (%)	R. (%)	F1 (%)
CNN	No	90.77	68.70	54.95	58.65
RNN		90.67	68.92	54.79	58.49
LSTM		91.02	68.70	55.24	58.95
CNN	(Weighted Avg.)	90.77	91.29	89.73	58.39
RNN		90.67	67.36	62.86	63.41
LSTM		91.02	68.31	54.74	58.33
CNN	Yes	99.99	87.11	86.41	86.64
RNN		99.99	95.69	95.70	95.69
LSTM		99.99	97.12	97.13	97.13
CNN	(Weighted Avg.)	99.99	99.99	99.99	99.99
RNN		99.99	99.99	99.99	99.99
LSTM		99.99	99.99	99.99	99.99

Table 4.13 presents a comprehensive summary of the macro average metrics for classifier performance across all tested models (CNN, RNN, and LSTM) on both the original and feature enhanced datasets. These results are obtained through 10-fold stratified k-fold cross validation to ensure the stability of the results. The original dataset achieves accuracy score of around 90-91%, precision score of around 68-69%, recall score below 55%, and f1-score of 58-64%. The scores indicate that the model was accurate but struggle to be consistent across all classes. Weighted average is higher because of the performance of majority classes that are high, making the minority class negligible. The score difference from macro and weighted average emphasize the class imbalance problem settled inside the dataset.

The results from enhanced dataset show significant improvement from the baseline result of the non-FE dataset. All models can achieve accuracy, precision, recall, and f1-score of 99.99% in the weighted average metrics after the implementation of FE. FE model are capable of enhancing the dataset because it can generate realistic and diverse feature making the classifier able to learn and generalize the complex pattern that may underrepresented in the original dataset, so it can address the class imbalance. But, the macro average score still lower than the weighted average because the failure to detect extreme minority class of 'FileDownload'.

4.5. Quality of The FE Dataset

CGAN-LSTM, as a hybrid generative model, is capable of producing diverse synthetic features that can significantly enhance the richness and balance of datasets, particularly those suffering from class imbalance or limited representation. However, the effectiveness of the generated dataset depends critically on the quality and relevance of the synthetic data produced. It is essential that CGAN-LSTM not only generates data samples that vary broadly but also ensures these samples are meaningful and reflect the true underlying distribution of the original dataset. This means maintaining the characteristics and statistical properties of each class, so that the augmented data remains representative of real-world patterns and does not introduce distortions that could mislead the learning process.

Model must keep class identity where each generated feature must be accurately aligned with its intended class to avoid confusing the classifier during training. At the same time, measures must be taken to prevent overfitting in the GAN training process. Overfitting in generative models can lead to repeated or overly similar synthetic samples, reducing the diversity advantage and potentially causing the model to memorize existing data points rather than generalize new, plausible features. Hamed et al. (2025) stated that evaluating the quality of generated data involves verifying that the synthetic features are sufficiently similar to real data to support improved classification performance, yet sufficiently novel to enrich the feature space. The goal is to create augmented datasets that aid classifiers in learning more robust and generalized decision boundaries, as a result improving detection accuracy especially in minority or rare classes.

4.5.1. Visual Analysis using PCA

Visual analysis using Principal Component Analysis (PCA) is a widely adopted technique for evaluating the distribution of real and augmented datasets. PCA plays a critical role in data analysis by reducing the dimensionality of data while retaining most of the variance present in the original features. In this context, PCA reduces the dataset from potentially dozens or hundreds of features down to just two principal components, which can be easily visualized on a two-dimensional plot. This reduction simplifies complex, high-dimensional data into a form that allows clearer visual inspection and intuitive understanding of the data's structure.

The key importance of PCA lies in its ability to capture the most significant patterns and variations within the dataset. By transforming the original correlated features into a set of linearly uncorrelated principal components, PCA highlights the directions in which the data varies the most. This feature is especially valuable when comparing real and synthetic (augmented) datasets, as it allows researchers to check whether the synthetic data covers the same feature space similar to the real data. If the synthetic data distribution aligns well on the principal component plot, it suggests that the augmentation process preserves the underlying patterns and statistical properties of the original dataset.

Moreover, PCA's ability to reveal clustering tendencies in a reduced dimension makes it an excellent tool for assessing the effectiveness of data augmentation techniques such as CGAN-LSTM. For instance, tightly grouped points in the PCA scatter plot reflect compact and well-separated clusters, which indicate good class separability. Conversely, overlapping or dispersed points may suggest poor cluster definition or insufficient augmentation quality. PCA can serve as a diagnostic tool that provides visual confirmation of improvements or limitations in the synthetic data generation process. The resulting plot can be used to complement quantitative metrics like silhouette scores.

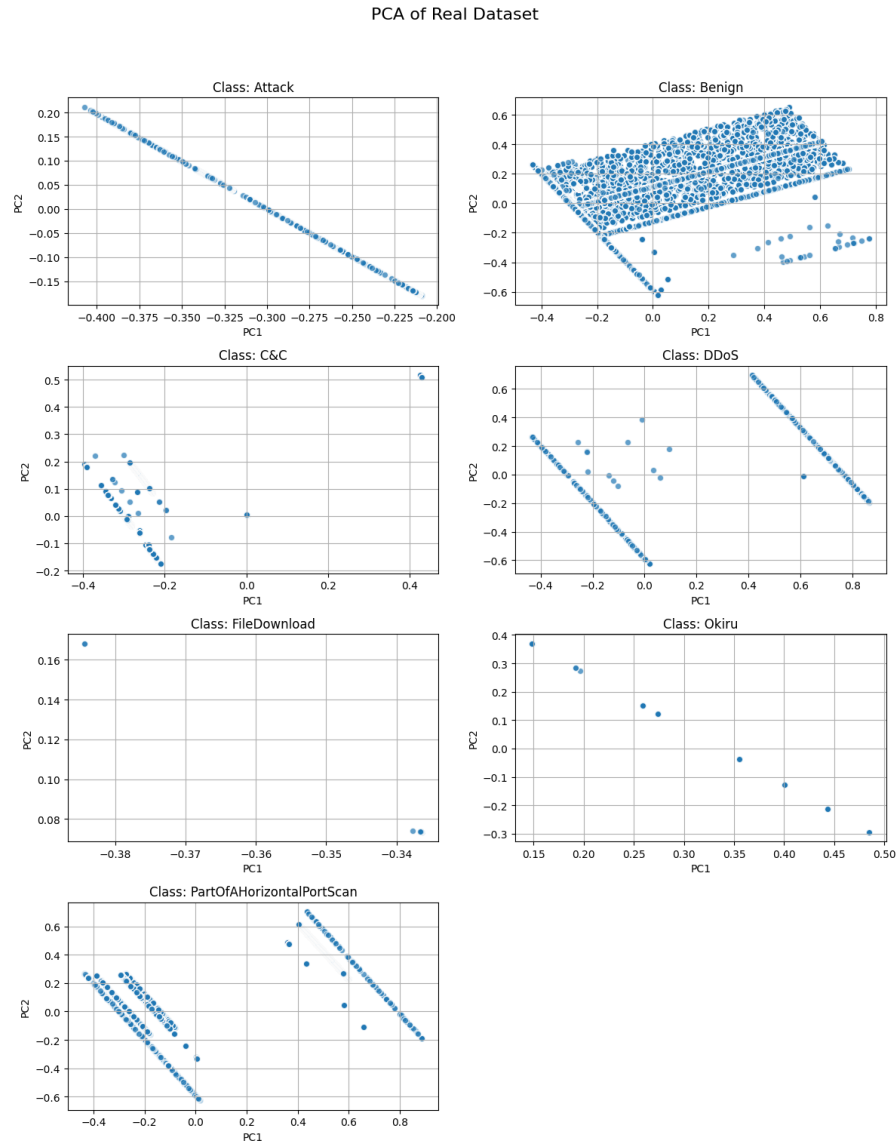


Figure 4.13 PCA Visualization of Original Dataset

Figure 4.13 shows the PCA projection of the original dataset. Original dataset showed that some classes are poorly clustered. 'Benign' class shows a scattered dot across the figure. It's suggested that this class has high variance and thus can overlap with other classes. It can lead to false positive behavior in the classification process. 'Attack' class shows a tight and linear cluster that indicates a limited intra class variance. It can be positive for classifier performance, but it can lead to domination of larger classes if the class is imbalanced. 'C&C', 'DDoS', and 'PartOfAHorizontalPortScan' appears to have multimodal clustering where the data points are divided into subgroups even in the same class. This shows an intra-class inconsistency which leads to a reduced accuracy and makes the classifier ignore the minor classes. 'FileDownload' and 'Okiru' represent a sparse distribution across the figure. This indicates underrepresentation in the dataset. These conditions can hinder the performance of the classifier where it limits the classifier to learn due to low examples. Decision boundaries will incline towards neighboring class which were more dominant.

The PCA analysis further confirms that the original dataset have class imbalances that can affect negatively classifier performance, poor intra-class clustering that leads to misclassification, and overlapping cluster that results in false positives. Class imbalances makes some of the classes like 'FileDownload', 'Okiru', and 'DDoS' nearly invisible in the PCs spaces. This will result in a less sufficient exposure to that class leading to difficulty of the models to learn meaningful pattern inside the class. This cause the model to misclassify or may entirely missing these rare classes. On the other hand, poor intra-class clustering weakens the ability of classifier to create interpretation of each class. It will increase confusion and false categorization especially when handling the data that looks similar to other classes or borderline data. Another issue is data overlap, where data in different classes overlapping with another class which create a further more confusion especially in 'Benign' class. It is dangerously fatal when models cannot detect abnormal behavior over the normal one which further increases the damage done by the malware.

PCA of Augmented Dataset

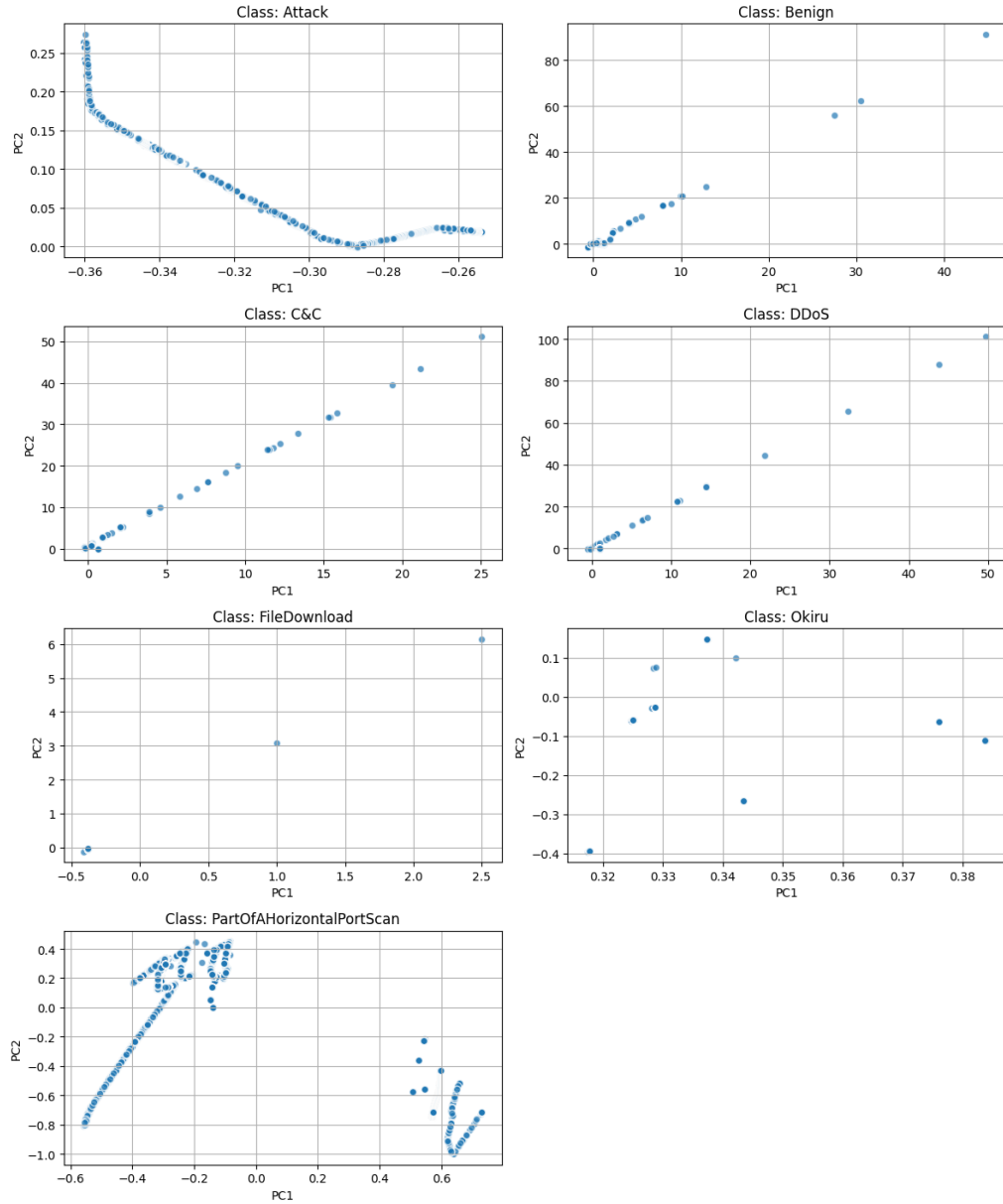


Figure 4.14 PCA Visualization of Enhanced Dataset

Figure 4.14 is the PCA visualization of the augmented dataset. The PCA analysis of the augmented dataset gives insight of how CGAN-LSTM affects the dataset. Almost every class shows an improvement in the overall structure and quality of the dataset. Cluster are more visible compared to the original dataset. PCA spaces shows clearer class boundaries, reduced overlap, and more compact intra-class clustering. This indicates that augmentation makes the dataset more

reliable, has clear distribution, and affects positively towards the performance of the classifiers.

Improvement of the dataset can be seen through the clustering of each class. ‘Benign’, ‘C&C’, and ‘DDoS’ classes is no longer scattered over the PCA spaces which form tighter and more centralized clusters. It shows how CGAN-LSTM able to preserve the class cluster by enriching the dataset. This augmentation allows classifiers to learn more about the underlying pattern and distribution of these classes. It contributes to the increase in classifier performance across all models. The CGAN-LSTM model can also improve the class visibility which is shown by the inter-class clustering that is more visible across all class. The augmented dataset shows reduction in the region that overlaps one another especially ‘Benign’ and the other attack types. In the original dataset, ‘Benign’ class often share feature to another class in the same space, which in the augmented dataset it was more distinct and well-partitioned on the projection

4.5.2. Cluster Analysis using Silhouette Score

Silhouette Score is widely used to evaluate clustering validity by measuring how well the data point placed inside a cluster is compared to other clusters. Silhouette Score ranges between +1 means well-clustered and -1 means poorly clustered. This part is exploring the silhouette scores for each class/cluster before and after applying FE-CGAN-LSTM. This analysis aims to understand how the synthetic features affect the compactness and separability of each class. It is crucial to understand this in terms of the data quality generated by CGAN-LSTM.

Table 4.14 Silhouette score across all class

Class	Dataset		Improvement (Δ)
	Original	Enhanced	
Attack	-0.2957	0.6703	0.9660
Benign	-0.4484	-0.2717	0.1767
C&C	-0.1324	0.3339	0.4663
DDoS	-0.0254	0.2720	0.2974
FileDownload	0.7045	-0.5429	-1.2474
Okiru	0.5541	0.6536	0.0995
PortScan	-0.4311	-0.0970	0.3341

Table 4.14 presents a comprehensive overview of the increases in silhouette scores across almost all the analysed classes. This result indicates how effective the CGAN-LSTM approach is in enhancing the quality of clustering. From these results, it is evident that CGAN-LSTM significantly improves both the compactness within clusters and the separability between different classes. This means that the data points belonging to the same class are grouped more tightly together, while distinct classes are more clearly differentiated from each other.

The ‘Attack’ class shows a particularly remarkable improvement. Initially, this class exhibited poor clustering quality, as reflected by a negative silhouette score which indicated overlapping or poorly defined clusters. After applying the CGAN-LSTM augmentation, the Silhouette score shifted positively, suggesting a much clearer and well-defined cluster structure for this category. Similarly, the ‘C&C’ class and the ‘PortScan’ class both demonstrate notable gains in their silhouette scores. However, it is important to highlight that the ‘PortScan’ class, despite showing improvement, continues to reside in the negative silhouette score range, indicating that while the clustering is better, it is still not ideal.

The synthetic features introduced through CGAN-LSTM augment the dataset by adding variance that broadly aligns with the inherent characteristics of the original data. However, this augmentation is not uniformly precise across all classes, which explains the disparity in improvements observed. For example, the ‘DDoS’ class sees a moderate increase in cluster quality, with Silhouette scores rising from negative values into the positive range. This improvement points out the ability of the augmentation to better isolate class samples, likely due to enhanced representation of their periodic traffic patterns and volume behavior, which are key characteristics in identifying DDoS attacks.

On the other hand, the ‘Benign’ class, the normal, non-attack traffic, shows only modest improvement and unfortunately remains within the negative silhouette score area. Despite this, the synthetic augmentation plays a critical role in reducing the false positive rate within this class, which is crucial for practical deployment where minimizing false alarms is paramount. Lastly, the ‘Okiru’ class, which is already well-clustered in the original dataset, experiences reinforcement of its

existing cluster patterns through the augmentation process, without any distortion or degradation to its cluster structure.

4.6. Performance of FE and Non-FE Model

This section discusses the effect of FE models compared with non-FE models. Table 4.15 presents the impact of feature enhancement (FE) on the weighted average performance metrics across various models used for multiclass malware detection on IoT datasets. It compares accuracy, precision, recall, and F1-score for different models with and without the application of feature enhancement, illustrating how enhancing the feature set can improve detection performance. The table spans different datasets and model architectures. Models that incorporate feature enhancement tend to achieve higher classification metrics, particularly for more complex scenarios with increased numbers of classes.

Table 4.15 Impact of FE on Weighted Average Performance Metrics in Multiclass Malware Detection

Model	Dataset	NoC.*	A. (%)	P. (%)	R. (%)	F1. (%)
ANN (Jamal et al., 2022)	ToN-IoT	9	97.08	98.45	98.41	96.55
Hybrid CNN-RNN (Alanzi & Alzahrani, 2024)	IoT-23	2	99.70	99.20	99.10	99.20
DEMD-IoT (Nobakht et al., 2023)	IoT-23	2	99.90	99.83	99.97	99.90
FE-MDTM (Wei et al., 2023)	IoT-23	4	99.70	99.70	99.70	99.70
CGAN-LSTM	IoT-23	7	99.99	99.99	99.99	99.99

*NoC: Number of trained classes

In this study, the impact of FE on malware detection is highlighted by examining various models and datasets, illustrating how feature enhancement significantly influences classification efficacy and granularity. The Artificial Neural Network (ANN) model, employed on the ToN-IoT dataset with nine classes

(one benign and eight attack types), achieves moderately high weighted average metrics, accuracy at 97.08%, precision at 98.45%, recall at 98.42%, and F1-score at 96.55%. Although ANN shows solid capability in detecting malicious activities broadly, its performance is hindered by difficulty in correctly classifying minority classes, which leads to misclassification errors, especially for less common attack types. This limitation further emphasize the challenge of complex multiclass scenarios without advanced feature treatment.

Hybrid CNN-RNN model applied to the IoT-23 dataset simplifies the classification into two categories, benign and malicious. The performance improves significantly, with accuracy nearing 99.7% and similarly high precision, recall, and F1-score values. The model benefits from the complementary nature of CNN and RNN architectures to extract spatial and temporal features; however, its binary classification restricts its practical application in identifying specific attack types, limiting actionable insights.

The DEMD-IoT model further utilizes a deep CNN for malware detection, also on the IoT-23 dataset with two major classes. Despite not incorporating feature enhancement, this model attains excellent results with accuracy and F1-score near 99.9%, indicative of the power of deep feature extraction in distinguishing benign from malicious traffic. Nonetheless, like the Hybrid CNN-RNN, the lack of feature enhancement and limited class granularity reduces its usefulness for nuanced threat identification.

The FE-MDTM model explicitly integrates a feature enhancement process before classification using general neural networks and random forest algorithms on a 4-class IoT-23 dataset, 'Benign', 'PortScan', 'C&C', and 'DDoS'. This enhancement step refines and augments the feature space, capturing more discriminative patterns intrinsic to each class, consequently improving classification robustness. The model achieves a balanced 99.7% across accuracy, precision, recall, and F1-score, demonstrating that feature enhancement elevates performance when distinguishing among multiple attack types rather than simple binary splits.

The CGAN-LSTM model proposed in this study represents the peak of performance by combining Conditional Generative Adversarial Networks (CGAN)

for data augmentation and Long Short-Term Memory (LSTM) networks for sequential pattern learning. Utilizing seven classes from the IoT-23 dataset, Benign, ‘Attack’, ‘PortScan’, ‘C&C’, ‘Okiru’, ‘FileDownload’, and ‘DDoS’, the model achieves near-perfect weighted average scores of 99.99% across all metrics. The feature enhancement gives positive effect to the dataset by enriching the input representations, allowing the LSTM to capture complex temporal and spatial dependencies in the data. The CGAN component further amplifies the benefit by generating synthetic yet realistic samples, addressing class imbalance, and improving minority class detection. This combination showcases how feature enhancement synergizes with advanced modelling to drastically improve the accuracy and reliability of multiclass malware detection in IoT environments.

4.7. Limitations

The CGAN-LSTM model, despite its innovative integration of conditional generation and sequential learning, faces several notable challenges that can impact its performance and usability in practice. One of the primary difficulties is the vanishing gradients problem, which occurs during the adversarial training phase. As the discriminator network becomes highly proficient at distinguishing real from synthetic data, the gradient signals that the generator relies on to improve become increasingly weak or vanish altogether. This phenomenon can hinder the generator’s learning progress, causing delays in convergence or even failure to generate high-quality synthetic samples. The delicate balance between generator and discriminator requires meticulous hyperparameter tuning and potentially the use of advanced training techniques such as gradient penalty or modified loss functions to alleviate this issue.

Furthermore, the CGAN-LSTM model is susceptible to mode collapse, a scenario where the generator produces a limited variety of outputs rather than adequately capturing the diverse distribution of the underlying data. This happens particularly in complex datasets like IoT-23, which contains multiple classes with highly imbalanced representation. Mode collapse limits the diversity and richness of the synthetic samples generated, thus constraining the model’s capacity to genuinely augment minority classes. This reduction in variability can bias the

downstream classifiers trained on the augmented dataset, potentially leading to overfitting synthetic examples and diminishing generalization to real-world data. Techniques such as mini-batch discrimination, feature matching, or incorporating multiple generators have been proposed to address this issue, but no universal solution fully eliminates the problem.

Another challenge lies in the inconsistency occasionally observed in the data generated. Due to the adversarial dynamics and the inherent instability in GAN training, the synthetic outputs may sometimes exhibit discrepancies or noise that do not conform to the statistical properties of the original dataset. These inconsistencies can manifest as unrealistic feature values, anomalous correlations, or distributional shifts, which may inadvertently mislead the classifiers trained on this data. In high-stakes domains like cybersecurity, the introduction of such noisy or biased synthetic data can degrade model reliability and compromise detection accuracy, particularly for subtle or rare attack patterns. Ensuring high fidelity in synthetic data production often demands additional regularization, validation mechanisms, or post-generation filtering steps.

[Page intentionally left blank]

CHAPTER 5

CONCLUSIONS AND SUGGESTIONS

5.1. Conclusion

5.1.1. Implementation of CGAN-LSTM to Enhance Dataset and Improve Classifier Performance

CGAN-LSTM can be implemented to enhance the dataset and improve the classifier performance for up to 20% from baseline model where the average score across all macro-averaged metrics are Accuracy (99.99%), Precision (93.30%), Recall (93.08%), and F1-score (93.15%). These results confirm that the CGAN-LSTM model can perform feature enhancement to boost the classification performance of Intrusion Detection System.

5.1.2. Quality of Generated Data and Model Performance Compared to Other Models

Performance of all classifiers significantly increased from the baseline model. Compared to other model, this model achieved 99.99% score across all metrics. Although there was great performance showed by the model, minority class like 'FileDownload' still failed to be detected, it suggest that the model failed to generalize underrepresented classes. CGAN-LSTM generated data shows great quality but need to be analysed further because of the potential of mode collapse.

5.2. Suggestion

To further enhance the performance and robustness of CGAN-LSTM-based feature enhancement, future research should explore fine-tuning techniques to prevent issues such as mode collapse, where the generator fails to produce sufficient diversity in output. Strategies may include adjusting learning rates, experimenting with different loss functions, applying instance noise or label smoothing, and incorporating class-wise loss monitoring during training.

The quality of synthetic data can be improved by enforcing distributional constraints through techniques such as feature matching or auxiliary classifiers. For underrepresented classes, conditional sampling or oversampling in the latent space may help ensure balanced class representation.

[Page intentionally left blank]

REFERENCES

- Abdalgawad, N., Sajun, A., Kaddoura, Y., Zualkernan, I. A., & Aloul, F. (2022). Generative Deep Learning to Detect Cyberattacks for the IoT-23 Dataset. *IEEE Access*, 10, 6430–6441. <https://doi.org/10.1109/ACCESS.2021.3140015>
- Alabsi, B. A., Anbar, M., & Rihan, S. D. A. (2023). Conditional Tabular Generative Adversarial Based Intrusion Detection System for Detecting Ddos and Dos Attacks on the Internet of Things Networks. *Sensors* 2023, Vol. 23, Page 5644, 23(12), 5644. <https://doi.org/10.3390/S23125644>
- Alanzi, S. M., & Alzahrani, Dr. A. J. (2024). IoT Malware Detection Using Hybrid Deep Learning Algorithms. *International Journal of Computer Science and Network Security*, 24(12), 1–17. <https://doi.org/10.22937/IJCSNS.2024.24.12.1>
- Alfares, H., & Banimelhem, O. (2024). Comparative Analysis of Machine Learning Techniques for Handling Imbalance in IoT-23 Dataset for Intrusion Detection Systems. *2024 11th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, 112–119. <https://doi.org/10.1109/IOTSMS62296.2024.10710296>
- Allam, Z., Bibri, S. E., Jones, D. S., Chabaud, D., & Moreno, C. (2022). Unpacking the ‘15-Minute City’ via 6G, IoT, and Digital Twins: Towards a New Narrative for Increasing Urban Efficiency, Resilience, and Sustainability. *Sensors*, 22(4). <https://doi.org/10.3390/s22041369>
- Almasre, M., & Subahi, A. (2024). Create a Realistic IoT Dataset Using Conditional Generative Adversarial Network. *Journal of Sensor and Actuator Networks*, 13(5). <https://doi.org/10.3390/jsan13050062>
- Alqahtani, A., Azzony, S., Alsharafi, L., & Alaseri, M. (2023). Web-Based Malware Detection System Using Convolutional Neural Network. *Digital*, 3(3), 273–285. <https://doi.org/10.3390/digital3030017>
- Altunay, H. C., & Albayrak, Z. (2023). A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks. *Engineering Science and Technology, an International Journal*, 38, 101322. <https://doi.org/10.1016/J.JESTCH.2022.101322>
- AV-TEST. (2024). *Malware*. <https://www.av-test.org/en/statistics/malware/>
- Azati, Y., Wang, X., Quddus, M., & Zhang, X. (2024). Graph convolutional LSTM algorithm for real-time crash prediction on mountainous freeways. *International Journal of Transportation Science and Technology*. <https://doi.org/10.1016/J.IJTST.2024.07.002>

- Bensaoud, A., & Kalita, J. (2024). CNN-LSTM and transfer learning models for malware classification based on opcodes and API calls. *Knowledge-Based Systems*, 290, 111543. <https://doi.org/10.1016/j.knosys.2024.111543>
- Bourechak, A., Zedadra, O., Kouahla, M. N., Guerrieri, A., Seridi, H., & Fortino, G. (2023). At the Confluence of Artificial Intelligence and Edge Computing in IoT-Based Applications: A Review and New Perspectives. *Sensors*, 23(3). <https://doi.org/10.3390/s23031639>
- Bousmina, A., Selmi, M., Ben Rhaïem, M. A., & Farah, I. R. (2023). A Hybrid Approach Based on GAN and CNN-LSTM for Aerial Activity Recognition. *Remote Sensing*, 15(14). <https://doi.org/10.3390/rs15143626>
- Chaganti, R., Ravi, V., & Pham, T. D. (2022). Deep learning based cross architecture internet of things malware detection and classification. *Computers & Security*, 120, 102779. <https://doi.org/https://doi.org/10.1016/j.cose.2022.102779>
- Chatterjee, S., Hazra, D., & Byun, Y.-C. (2025). GAN-based synthetic time-series data generation for improving prediction of demand for electric vehicles. *Expert Systems with Applications*, 264, 125838. <https://doi.org/https://doi.org/10.1016/j.eswa.2024.125838>
- Cheng, G., Lang, C., Wu, M., Xie, X., Yao, X., & Han, J. (2025). Feature Enhancement Network for Object Detection in Optical Remote Sensing Images. *Journal of Remote Sensing*, 2021. <https://doi.org/10.34133/2021/9805389>
- Chu, C., Hastak, A., & Chen, F. (2024). LSTM-QGAN: Scalable NISQ Generative Adversarial Network. *ArXiv Preprint ArXiv:2409.02212*.
- Cui, Z., Jing, X., Zhao, P., Zhang, W., & Chen, J. (2021). A New Subspace Clustering Strategy for AI-Based Data Analysis in IoT System. *IEEE Internet of Things Journal*, 8(16), 12540–12549. <https://doi.org/10.1109/IIOT.2021.3056578>
- Dablain, D., Krawczyk, B., & Chawla, N. V. (2023). DeepSMOTE: Fusing Deep Learning and SMOTE for Imbalanced Data. *IEEE Transactions on Neural Networks and Learning Systems*, 34(9), 6390–6404. <https://doi.org/10.1109/TNNLS.2021.3136503>
- Estenssoro, J. V. (2024). *Malware and Virus Statistics 2024: The Trends You Need to Know About*. <https://www.avg.com/en/signal/malware-statistics>
- Gamal, M., Elhamahmy, M., Taha, S., & Elmahdy, H. (2024a). Improving intrusion detection using LSTM-RNN to protect drones' networks. *Egyptian Informatics Journal*, 27, 100501. <https://doi.org/https://doi.org/10.1016/j.eij.2024.100501>

- Gamal, M., Elhamahmy, M., Taha, S., & Elmahdy, H. (2024b). Improving intrusion detection using LSTM-RNN to protect drones' networks. *Egyptian Informatics Journal*, 27, 100501. <https://doi.org/https://doi.org/10.1016/j.eij.2024.100501>
- Gao, N., Xue, H., Shao, W., Zhao, S., Qin, K. K., Prabowo, A., Rahaman, M. S., & Salim, F. D. (2021). Generative Adversarial Networks for Spatio-temporal Data: A Survey. *ACM Transactions on Intelligent Systems and Technology*, 13(2), 26. <https://doi.org/10.1145/3474838>
- Garcia, S., Parmisano, A., & Erquiaga, M. J. (2020). *IoT-23: A labeled dataset with malicious and benign IoT network traffic*. Zenodo. <https://doi.org/10.5281/zenodo.4743746>
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, 2672–2680.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems* (Vol. 27). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf
- Gopali, S., Abri, F., Siامي-Namini, S., & Namin, A. S. (2021). *A Comparative Study of Detecting Anomalies in Time Series Data Using LSTM and TCN Models*. <http://arxiv.org/abs/2112.09293>
- Hamed, S. Kh., Ab Aziz, M. J., & Yaakub, M. R. (2025). A data augmentation approach based on various GAN models to address class imbalance in fine-grained multimodal fake news datasets. *Computing*, 107(1), 52. <https://doi.org/10.1007/s00607-025-01413-2>
- Humayun, M., Jhanjhi, N. Z., Alsayat, A., & Ponnusamy, V. (2021). Internet of things and ransomware: Evolution, mitigation and prevention. *Egyptian Informatics Journal*, 22(1), 105–117. <https://doi.org/https://doi.org/10.1016/j.eij.2020.05.003>
- Jamal, A., Hayat, M. F., & Nasir, M. (2022). Malware detection and classification in IoT network using ANN. *Mehran University Research Journal Of Engineering & Technology*, 41(1), 80–91. <https://search.informit.org/doi/10.3316/informit.263296849285942>
- Khan, A. (2022). *Balanced Split: A new train-test data splitting strategy for imbalanced datasets*. <https://doi.org/10.48550/arXiv.2212.11116>

- Khan, M., Mehran, M. T., Haq, Z. U., Ullah, Z., Naqvi, S. R., Ihsan, M., & Abbass, H. (2021). Applications of artificial intelligence in COVID-19 pandemic: A comprehensive review. *Expert Systems with Applications*, 185, 115695. <https://doi.org/10.1016/J.ESWA.2021.115695>
- Khan, S. H., Alahmadi, T. J., Ullah, W., Iqbal, J., Rahim, A., Alkahtani, H. K., Alghamdi, W., & Almagrabi, A. O. (2023). A new deep boosted CNN and ensemble learning based IoT malware detection. *Computers & Security*, 133, 103385. <https://doi.org/10.1016/J.COSE.2023.103385>
- Koohang, A., Sargent, C. S., Nord, J. H., & Paliszkiewicz, J. (2022). Internet of Things (IoT): From awareness to continued use. *International Journal of Information Management*, 62, 102442. <https://doi.org/https://doi.org/10.1016/j.ijinfomgt.2021.102442>
- Li, J., Chen, H., Shahizan, M. O., & Yusuf, L. M. (2024). Enhancing IoT security: A comparative study of feature reduction techniques for intrusion detection system. *Intelligent Systems with Applications*, 23, 200407. <https://doi.org/10.1016/J.ISWA.2024.200407>
- Lim, W., Yong, K. S. C., Lau, B. T., & Tan, C. C. L. (2024). Future of generative adversarial networks (GAN) for anomaly detection in network security: A review. *Computers & Security*, 139, 103733. <https://doi.org/https://doi.org/10.1016/j.cose.2024.103733>
- Liu, X., & Liu, J. (2021). Malicious traffic detection combined deep neural network with hierarchical attention mechanism. *Scientific Reports*, 11(1), 12363. <https://doi.org/10.1038/s41598-021-91805-z>
- Lu, H., Du, M., Qian, K., He, X., & Wang, K. (2022). GAN-Based Data Augmentation Strategy for Sensor Anomaly Detection in Industrial Robots. *IEEE Sensors Journal*, 22(18), 17464–17474. <https://doi.org/10.1109/JSEN.2021.3069452>
- Malashin, I., Tynchenko, V., Gantimurov, A., Nelyub, V., & Borodulin, A. (2024). Applications of Long Short-Term Memory (LSTM) Networks in Polymeric Sciences: A Review. *Polymers*, 16(18). <https://doi.org/10.3390/polym16182607>
- Massarelli, L., Aniello, L., Ciccotelli, C., Querzoni, L., Ucci, D., & Baldoni, R. (2020). AndroDFA: Android Malware Classification Based on Resource Consumption. *Information*, 11(6). <https://doi.org/10.3390/info11060326>
- Mishra, A. K., Paliwal, S., & Srivastava, G. (2024). Anomaly detection using deep convolutional generative adversarial networks in the internet of things. *ISA Transactions*, 145, 493–504. <https://doi.org/https://doi.org/10.1016/j.isatra.2023.12.005>

- Moti, Z., Hashemi, S., Karimipour, H., Dehghantanha, A., Jahromi, A. N., Abdi, L., & Alavi, F. (2021). Generative adversarial network to detect unseen Internet of Things malware. *Ad Hoc Networks*, 122, 102591. <https://doi.org/10.1016/J.ADHOC.2021.102591>
- Mutambik, I. (2024). Enhancing IoT Security Using GA-HDLAD: A Hybrid Deep Learning Approach for Anomaly Detection. *Applied Sciences*, 14(21). <https://doi.org/10.3390/app14219848>
- Nguyen, C.-D., Khoa, N. H., Doan, K. N.-D., & Cam, N. T. (2023). Android Malware Category and Family Classification Using Static Analysis. *2023 International Conference on Information Networking (ICOIN)*, 162–167. <https://doi.org/10.1109/ICOIN56518.2023.10049039>
- Nobakht, M., Javidan, R., & Pourebrahimi, A. (2023). DEMD-IoT: a deep ensemble model for IoT malware detection using CNNs and network traffic. *Evolving Systems*, 14(3), 461–477. <https://doi.org/10.1007/s12530-022-09471-z>
- Paramasivam, K., Sindha, M. M. R., & Balakrishnan, S. B. (2023). KNN-Based Machine Learning Classifier Used on Deep Learned Spatial Motion Features for Human Action Recognition. *Entropy*, 25(6). <https://doi.org/10.3390/e25060844>
- Rahman, S., Pal, S., Mittal, S., Chawla, T., & Karmakar, C. (2024). SYN-GAN: A robust intrusion detection system using GAN-based synthetic data for IoT security. *Internet of Things*, 26, 101212. <https://doi.org/10.1016/J.IOT.2024.101212>
- Robert Frost, H. (2022). Eigenvectors from Eigenvalues Sparse Principal Component Analysis. *Journal of Computational and Graphical Statistics*, 31(2), 486–501. <https://doi.org/10.1080/10618600.2021.1987254>;REQUESTEDJOURNAL:JOURNAL:UCGS20;WGROU:STRING:PUBLICATION
- Ruiz-Gándara, A., & Gonzalez-Abril, L. (2024). Generative Adversarial Networks in Business and Social Science. *Applied Sciences*, 14(17). <https://doi.org/10.3390/app14177438>
- Sadeghi, M., Casey, P., Carranza, E. J. M., & Lynch, E. P. (2024). Principal components analysis and K-means clustering of till geochemical data: Mapping and targeting of prospective areas for lithium exploration in Västernorrland Region, Sweden. *Ore Geology Reviews*, 167, 106002. <https://doi.org/10.1016/J.OREGEOREV.2024.106002>
- Schiller, E., Aidoo, A., Fuhrer, J., Stahl, J., Ziörjen, M., & Stiller, B. (2022). Landscape of IoT security. *Computer Science Review*, 44, 100467. <https://doi.org/https://doi.org/10.1016/j.cosrev.2022.100467>

- Shahin, M., Maghanaki, M., Hosseinzadeh, A., & Chen, F. F. (2024). Advancing Network Security in Industrial IoT: A Deep Dive into AI-Enabled Intrusion Detection Systems. *Advanced Engineering Informatics*, 62, 102685. <https://doi.org/https://doi.org/10.1016/j.aei.2024.102685>
- Shao, F., Shao, H., Wang, D., Lam, W. H. K., & Cao, S. (2023). A generative model for vehicular travel time distribution prediction considering spatial and temporal correlations. *Physica A: Statistical Mechanics and Its Applications*, 621, 128769. <https://doi.org/10.1016/J.PHYSA.2023.128769>
- Shareef, S. K. K., Chaitanya, R. K., Chennupalli, S., Chokkakula, D., Kiran, K. V. D., Pamula, U., & Vatambeti, R. (2024). Enhanced botnet detection in IoT networks using zebra optimization and dual-channel GAN classification. *Scientific Reports*, 14(1), 17148. <https://doi.org/10.1038/s41598-024-67865-2>
- Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena*, 404, 132306. <https://doi.org/10.1016/J.PHYSD.2019.132306>
- Shi, T., McCann, R. A., Huang, Y., Wang, W., & Kong, J. (2024). Malware Detection for Internet of Things Using One-Class Classification. *Sensors*, 24(13). <https://doi.org/10.3390/s24134122>
- Swana, E. F., Doorsamy, W., & Bokoro, P. (2022). Tomek Link and SMOTE Approaches for Machine Fault Classification with an Imbalanced Dataset. *Sensors*, 22(9). <https://doi.org/10.3390/s22093246>
- Swathi, B., Kolisetty, S. S., Sivanarayana, G. V., & Battula, S. R. (2024). Efficientnetv2-RegNet: an effective deep learning framework for secure SDN based IOT network. *Cluster Computing*, 27(8), 10653–10670. <https://doi.org/10.1007/s10586-024-04498-0>
- Torre, D., Chennamaneni, A., Jo, J., Vyas, G., & Sabrsula, B. (2024). Towards Enhancing Privacy-Preservation of a Federated Learning CNN Intrusion Detection System in IoT: Method and Empirical Study. *ACM Trans. Softw. Eng. Methodol.* <https://doi.org/10.1145/3695998>
- Ullah, I., & Mahmoud, Q. H. (2021). A Framework for Anomaly Detection in IoT Networks Using Conditional Generative Adversarial Networks. *IEEE Access*, 9, 165907–165931. <https://doi.org/10.1109/ACCESS.2021.3132127>
- van der Goot, R. (2021). We Need to Talk About train-dev-test Splits. In M.-F. Moens, X. Huang, L. Specia, & S. W. Yih (Eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 4485–4494). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2021.emnlp-main.368>

- Wakamiya, S., Li, C.-T., Huang, C.-T., Tsoulos, I. G., Ahmad, I., Wan, Z., Ahmad, A., & Sajid Ullah, S. (2024). A Hybrid Optimization Model for Efficient Detection and Classification of Malware in the Internet of Things. *Mathematics* 2024, Vol. 12, Page 1437, 12(10), 1437. <https://doi.org/10.3390/MATH12101437>
- Wang, S., Dai, Y., Shen, J., & Xuan, J. (2021). Research on expansion and classification of imbalanced data based on SMOTE algorithm. *Scientific Reports*, 11(1), 24039. <https://doi.org/10.1038/s41598-021-03430-5>
- Wei, N., Yin, L., Zhou, X., Ruan, C., Wei, Y., Luo, X., Chang, Y., & Li, Z. (2023a). A feature enhancement-based model for the malicious traffic detection with small-scale imbalanced dataset. *Information Sciences*, 647, 119512. <https://doi.org/https://doi.org/10.1016/j.ins.2023.119512>
- Wei, N., Yin, L., Zhou, X., Ruan, C., Wei, Y., Luo, X., Chang, Y., & Li, Z. (2023b). A feature enhancement-based model for the malicious traffic detection with small-scale imbalanced dataset. *Information Sciences*, 647, 119512. <https://doi.org/https://doi.org/10.1016/j.ins.2023.119512>
- Wei, N., Yin, L., Zhou, X., Ruan, C., Wei, Y., Luo, X., Chang, Y., & Li, Z. (2023c). A feature enhancement-based model for the malicious traffic detection with small-scale imbalanced dataset. *Information Sciences*, 647, 119512. <https://doi.org/https://doi.org/10.1016/j.ins.2023.119512>
- Wu, C. Y., Ban, T., Cheng, S. M., Takahashi, T., & Inoue, D. (2023). IoT malware classification based on reinterpreted function-call graphs. *Computers & Security*, 125, 103060. <https://doi.org/10.1016/J.COSE.2022.103060>
- Xu, B., He, G., & Zhu, H. (2021). ME-Box: A reliable method to detect malicious encrypted traffic. *Journal of Information Security and Applications*, 59, 102823. <https://doi.org/https://doi.org/10.1016/j.jisa.2021.102823>
- Yang, D., Wang, X., Zhu, N., Li, S., & Hou, N. (2023). MJ-GAN: Generative Adversarial Network with Multi-Grained Feature Extraction and Joint Attention Fusion for Infrared and Visible Image Fusion. *Sensors*, 23(14). <https://doi.org/10.3390/s23146322>
- Yang, Y., Liu, X., Wang, D., Sui, Q., Yang, C., Li, H., Li, Y., & Luan, T. (2025). A CE-GAN based approach to address data imbalance in network intrusion detection systems. *Scientific Reports*, 15(1), 1–19. <https://doi.org/10.1038/S41598-025-90815-5>;SUBJMETA=1042,117,639,705;KWRD=COMPUTATIONAL+SCIENCE,COMPUTER+SCIENCE
- Yuan, B., Wang, J., Wu, P., & Qing, X. (2022). IoT Malware Classification Based on Lightweight Convolutional Neural Networks. *IEEE Internet of Things Journal*, 9(5), 3770–3783. <https://doi.org/10.1109/JIOT.2021.3100063>

- Zang, X., Wang, T., Zhang, X., Gong, J., Gao, P., & Zhang, G. (2024). Encrypted malicious traffic detection based on natural language processing and deep learning. *Computer Networks*, 250, 110598. <https://doi.org/https://doi.org/10.1016/j.comnet.2024.110598>
- Zeghida, H., Boulaiche, M., Chikh, R., Bamhdi, A. M., Barros, A. L. B., Zeghida, D., & Patel, A. (2024). Enhancing IoT cyber attacks intrusion detection through GAN-based data augmentation and hybrid deep learning models for MQTT network protocol cyber attacks. *Cluster Computing*, 28(1), 58. <https://doi.org/10.1007/s10586-024-04752-5>
- Zhang, Y., & Liu, Q. (2022). On IoT intrusion detection based on data augmentation for enhancing learning on unbalanced samples. *Future Generation Computer Systems*, 133, 213–227. <https://doi.org/10.1016/J.FUTURE.2022.03.007>
- Zheng, Q., Chen, S., Wang, G., Li, L., Peng, S., & Yao, Z. (2025). An accurate and efficient self-distillation method with channel-based feature enhancement via feature calibration and attention fusion for Internet of Things. *Future Generation Computer Systems*, 169, 107816. <https://doi.org/10.1016/J.FUTURE.2025.107816>

AUTHOR BIOGRAPHY



Gregorius Edo holds a bachelor's degree in chemical engineering from Universitas Negeri Jember (UNEJ) and is currently pursuing a master's degree in informatics engineering at Institut Teknologi Sepuluh Nopember (ITS). His academic and professional interests include machine learning, network security, and network design.

Edo is passionate about server architecture, secure network topology, and the implementation of robust security measures in wired and wireless systems. He is a young entrepreneur in the field of IT infrastructure and currently manages an IT consultancy specializing in network solutions. He also serves as a network engineer at Universitas Katolik Widya Mandala Surabaya, where he contributes to the design and optimization of campus-wide network systems.