A LIGHTWEIGHT INTRUSION DETECTION FRAMEWORK USING FOCAL LOSS VARIATIONAL AUTOENCODER FOR INTERNET OF THINGS

SHAPLA KHANAM

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITI MALAYA
KUALA LUMPUR

2022

A LIGHTWEIGHT INTRUSION DETECTION FRAMEWORK USING FOCAL LOSS VARIATIONAL AUTOENCODER FOR INTERNET OF THINGS

SHAPLA KHANAM

THESIS SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITI MALAYA
KUALA LUMPUR

2022

UNIVERSITI MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: SHAPLA KHANAM

Registration/Matric No.: 17004551/2

Name of Degree: Doctor of Philosophy

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):

A Lightweight Intrusion Detection Framework using Focal Loss Variational Autoencoder for Internet of Things

Field of Study: Internet of Things

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date: 25/08/2022

Subscribed and solemnly declared before,

Witness's Signature Date: 25 August 2022

Name:
Designation:

A LIGHTWEIGHT INTRUSION DETECTION FRAMEWORK USING FOCAL LOSS VARIATIONAL AUTOENCODER FOR INTERNET OF THINGS

ABSTRACT

Internet of Things (IoT) generates imbalanced network traffic; thus, the connected objects in the IoT face security issues, including different and unknown attack types. Even though traditional learning-based techniques have been used for intrusion detection in IoT, the detection of low-frequency attacks is lacking due to the imbalanced nature of network traffic. For example, conventional learning-based techniques suffer from lower detection accuracy, higher False Positive Rate (FPR), and lower minority-class attacks detection rates. Moreover, due to the constrained nature of IoT, the conventional heavyweight intrusion detection models are not suitable for IoT. To overcome these issues, this research aims to establish and evaluate a lightweight intrusion-detection framework using Class-wise Focal Loss Variational Autoencoder (CFLVAE) for IoT. In establishing the proposed framework, a data generation model was developed using CFLVAE. Precisely, the CFLVAE model utilizes an efficient and cost-sensitive objective function called Class-wise Focal Loss (CFL) to train Variational AutoEncoder (VAE) to solve the data imbalance problem. Additionally, a highly imbalanced NSL-KDD intrusion dataset is employed to conduct extensive experimentation of the proposed model. Furthermore, a Lightweight Deep Neural Network (LDNN) model is established for intrusion detection in the IoT and trained using the balanced intrusion dataset created from the CFLVAE model to improve the intrusion detection performance. To maintain lightweight criteria, feature reduction using Mutual Information (MI) method and network compression using the Quantization technique are applied. The results demonstrate that the proposed CFLVAE with LDNN

(CFLVAE-LDNN) framework obtains promising performance in generating realistic new intrusion data samples and achieves superior intrusion detection performance. Specifically, the CFLVAE-LDNN achieves 88.08% overall intrusion detection accuracy and 3.77% false positive rate. It also achieved 79.25%, and 67.5% for Root to Local (R2L) and User to Root (U2R) low-frequency attacks detection rates, respectively. More significantly, low memory and CPU time consumption confirm that the proposed model is suitable for resource-constrained IoT. Overall, the proposed model benefits researchers and practitioners with intrusion detection in IoT.

Keywords: Internet of Things, Intrusion Detection, Data Imbalance, Focal Loss Variational Autoencoder, Deep Neural Network.

MODEL PENGESANAN PENCEROBOHAN RINGAN MENGGUNAKAN FOCAL LOSS VARIATIONAL AUTOENCODER UNTUK INTERNET BENDA

ABSTRAK

Internet Benda (IoT) menjana trafik rangkaian yang tidak seimbang; oleh itu, objek bersambungan dalam IoT menghadapi isu keselamatan termasuk jenis serangan yang berbeza dan tidak diketahui. Walaupun teknik berasaskan pembelajaran tradisional telah digunakan untuk pengesanan pencerobohan dalam IoT, pengesanan serangan yang mempunyai frekuensi/bilangan yang rendah adalah kurang disebabkan sifat trafik rangkaian yang tidak seimbang. Contohnya, teknik berasaskan pembelajaran konvensional mengalami ketepatan pengesanan yang lebih rendah, Kadar Positif Palsu (FPR) yang lebih tinggi dan kadar pengesanan serangan kelas minoriti yang lebih rendah. Selain itu, disebabkan sifat IoT yang terhad, model pengesanan pencerobohan wajaran berat konvensional tidak sesuai untuk IoT. Untuk mengatasi isu ini, penyelidikan ini bertujuan untuk mewujudkan dan menilai model pengesanan pencerobohan ringan menggunakan Class-wise Focal Loss Variational Autoencoder (CFLVAE) untuk IoT. Dalam mewujudkan model yang dicadangkan, model penjanaan data telah dibangunkan menggunakan CFLVAE. Tepatnya, model CFLVAE menggunakan fungsi objektif yang cekap dan sensitif kos yang dipanggil Class-wise Focal Loss (CFL) untuk melatih Variational AutoEncoder (VAE) untuk menyelesaikan masalah ketidakseimbangan data. Selain itu, set data pencerobohan NSL-KDD yang sangat tidak seimbang digunakan untuk menjalankan eksperimen yang meluas bagi model yang dicadangkan. Tambahan pula, model Rangkaian Neural Dalam Ringan (LDNN) ringan telah dibina untuk pengesanan pencerobohan dalam IoT dan dilatih menggunakan set data pencerobohan seimbang yang dihasilkan daripada model CFLVAE untuk meningkatkan prestasi pengesanan pencerobohan. Untuk mengekalkan kriteria

ringan, pengurangan ciri menggunakan kaedah Mutual Information (MI) dan pemampatan rangkaian menggunakan teknik Quantization digunakan. Keputusan menunjukkan bahawa rangka kerja CFLVAE dengan LDNN (CFLVAE-LDNN) yang dicadangkan memperoleh prestasi yang menjanjikan dalam menjana sampel data pencerobohan baharu yang realistik dan mencapai prestasi pengesanan pencerobohan yang unggul. Secara khusus, CFLVAE-LDNN mencapai 88.08% ketepatan pengesanan pencerobohan keseluruhan dan 3.77% kadar positif palsu. Ia juga mencapai 79.25%, dan 67.5% untuk kadar pengesanan serangan frekuensi rendah Root to Local (R2L) dan User to Root (U2R) masing-masing. Lebih ketara, memori yang rendah dan penggunaan masa CPU mengesahkan bahawa model yang dicadangkan sesuai untuk IoT yang dikekang oleh sumber. Secara keseluruhan, model yang dicadangkan memberi manfaat kepada penyelidik dan pengamal dengan pengesanan pencerobohan dalam IoT.

Kata kunci: Internet Benda, Pengesanan Pencerobohan, Ketidakseimbangan Data, Autoenkoder Variasi Kehilangan Fokus, Rangkaian Neural Dalam.

ACKNOWLEDGEMENTS

This Ph.D. has been an enriching and arduous journey, full of ups and downs, full of joy and tears, and incredibly full of rich experiences. Fortunately, I was surrounded by many supportive people who gave me the strength to endure difficult times. So, it is a great pleasure to thank them all for their positive and negative impacts during this quest.

Foremost, I would like to thank the Almighty for giving me the courage, patience, and strength to live in a foreign country alone to continue my education. Next, I cannot start without expressing all my gratitude to my loving family for their support, love, and comfort in the good and the bad and for bringing joy to my soul even though they are always far away. Hence, the utmost thanks go to my beloved parents who brought me to this earth and to all my elder brothers who helped me to grow and shared their experiences with me. My father always encouraged me to get a higher education and become a good human by contributing back to society. I appreciate my brother, who supported me financially throughout my education journey and supported me mentally when I felt down.

Naturally, I would like to express my gratitude to all my supervisors, including Dr. Ismail Ahmedy, Associate Professor Dr. Yamani Idna Idris, and Dr. Mohamed Hisham Jaward, for their guidance and support. Special Thanks to Dr. Mohamed Hisham Jaward for showing me the right research path. Moreover, I would like to thank the "IIRG" for publication funding. I would also like to thank the jury members for my defenses at several stages throughout this journey. I would also like to express my gratitude to all the examiners who agreed to read and review my thesis. I thank them in advance for all the attention they are willing to give to my work, as well as their valuable feedback. It is my utmost honor to have all these experts review my work.

Last but not least, I would like to thank all my dearest friends for their support, true

friendship and love, listening, positive and negative feedback, and support in difficult times. I am grateful to all my loved ones, especially those who helped me during this Ph.D. and whom I had the privilege of meeting during this journey.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

6LoWPAN	:	IPv6 over Low Power Wireless Personal Area Network.
ABE	:	Attribute-Based Encryption.
ADASYN	:	Adaptive Synthetic Sampling Approach.
AE	:	AutoEncoder.
AES	:	Advanced Encryption Standard.
AH	:	Authentication Header.
AKC	:	Asymmetric Key Cryptography.
AMQP	:	Advanced Message Queuing Protocol.
BLE	:	Bluetooth Low Energy.
CE	:	Cross Entropy.
CFL	:	Class-wise Focal Loss.
CIA	:	Confidentiality, Integrity and Availability.
CNN	:	Convolutional Neural Networks.
CoAP	:	Constrained Application Protocol.
CPA	:	Chosen Plaintext Attack.
CS	:	Compressive Sensing.
CVAE	:	Conditional Variational AutoEncoders.
DDoS	:	Distributed Denial of Service.
DES	:	Data Encryption Standard.
DH	:	Diffie-Hellman.
DL	:	Deep Learning.
DNN	:	Deep Neural Networks.
DODAG	:	Destination Oriented Directed Acyclic Graph.
DoS	:	Denial of Service.
DSA	:	Digital Signature Algorithms.
DT	:	Decision Tree.
DTLS	:	Datagram Transport Layer Security.
ECC	:	Elliptic-Curve Cryptography.
ECDH	:	Elliptic Curve Diffie-Hellman exchange.
ESP	:	Encapsulation Security Payload.
FL	:	Focal Loss.
FPR	:	False Positive Rate.
GAN	:	Generative Adversarial Networks.
HKC	:	Hybrid Key Cryptography.
IDEA	:	International Data Encryption Algorithm.
IDS	:	Intrusion Detection System.
IoT	:	Internet of Things.

IPS	:	Intrusion Prevention Systems.
IPv6	:	Internet Protocol Version 6.
KDC	:	Key Distribution Centre.
KNN	:	K-Nearest Neighbour.
LDNN	:	Lightweight Deep Neural Network.
LLN	:	Low Power and Lossy Network.
LSTM	:	Long Short-Term Memory.
MD5	:	Message Digest-5.
MitM	:	Man-in-the-Middle.
ML	:	Machine Learning.
MQTT	:	Message Queuing Telemetry Transport.
NB	:	Naïve Bayes.
PCA	:	Principle Component Analysis.
PHI	:	Protected Health Information.
PKC	:	Public-Key Cryptography.
PTQ	:	Post Training Quantization.
QAT	:	Quantization Aware Training.
R2L	:	Remote to Local.
RF	:	Random Forest.
RFID	:	Radio-Frequency Identification.
RNN	:	Recurrent Neural Network.
ROS	:	Random Over Sampling.
RPK	:	Raw Public Key.
RPL	:	Routing Protocol for Low Power and Lossy Networks.
RSA	:	Rivest–Shamir–Adleman.
SF	:	Selective Forwarding.
SLA	:	Service Level Agreement.
SMOTE	:	Synthetic Minority Oversampling Technique.
SVM	:	Support Vector Machine.
TCP	:	Transmission Control Protocol.
TEA	:	Tiny Encryption Algorithm.
U2R	:	User to Root.
UDP	:	User Datagram Protocol.
VAE	:	Variational AutoEncoders.
WiFi	:	Wireless Fidelity.
WLAN	:	Wireless Local Area Network.
WSN	:	Wireless Sensor Network.
XMPP	:	Extensible Messaging and Presence Protocol.

CHAPTER 1: INTRODUCTION

1.1 Background

With the constant growth and extensive application of the Internet of Things (IoT), big data, cloud computing, smart applications, and advanced network technologies, billions of devices are connected to the internet. The application of IoT has already been witnessed in all walks of life (Alaba et al., 2017). However, due to the constrained nature of IoT on memory, processor, power and information transmission, it suffers from significant security risks. Furthermore, because several IoT nodes gather and store an enormous volume of users' sensitive data, IoT has become an ultimate target for cyber adversaries (Alaba et al., 2017; Khanam et al., 2020). For instance, one of the leading aluminum companies named Norsk Hydro, was invaded on March 18, 2019, by LockerGoga (a variant of ransomware) (Briggs, 2019). The ransomware caused a shutdown of automated production lines of the aluminum company in Europe and the USA. Therefore, it is crucial to detect cyber-attacks on time to safeguard the network and its devices.

An Intrusion Detection System (IDS) is used to prevent and protect network devices from such security threats and vulnerabilities. Due to recent developments, IDS can identify and detect the attack types using Machine Learning (ML) and Deep Learning (DL) algorithms (Ahmad et al., 2021; H. Liu & Lang, 2019). ML approaches include Support Vector Machine (SVM), K-Nearest Neighbour (KNN), Decision Tree (DT), Random Forest (RF), Naïve Bayes (NB) (Chang et al., 2017; Jianhong, 2015; Zaman & Lung, 2018), and others. The efficiency of such learning methods has been applied and verified using several publicly available datasets, such as KDD99, NSL-KDD, UNSWNB15, and Kyoto (Janarthanan & Zargari, 2017; Moustafa & Slay, 2015; Protić, 2018), and they achieved significant intrusion detection performance. Also, DL approaches include DNN, Convolutional Neural

Networks (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), AutoEncoder (AE), Variational AutoEncoders (VAE) (Althubiti et al., 2018; Doersch, 2016; Hochreiter & Schmidhuber, 1997; Sajjad et al., 2019; Sak et al., 2014; Socher et al., 2011; Sutskever, 2013; Vaiyapuri & Binbusayyis, 2020; Y. Xiao et al., 2019; Y. Yang et al., 2020), along with others. ML and DL approaches have gained much research attention for their increasing popularity in high detection accuracy (Moustafa & Slay, 2015; Protić, 2018; Tavallaee et al., 2009a).

Notwithstanding the significant overall accuracy achieved by shallow ML and DL algorithms, current intrusion detection approaches for IoT still suffer from a high False Positive Rate (FPR) and inferior intrusion detection rates because of the imbalanced nature of real-network datasets (Chawla, 2009; H. He & Garcia, 2009; H. He & Ma, 2013; Jiang et al., 2020; Napierala & Stefanowski, 2016; Zuech et al., 2021). Due to the high dimension of real network data traffic, which is also imbalanced, the conventional deep learning algorithms still suffer from inefficiency in learning and classifying network attacks. Data imbalance refers to the state in which the class distribution is disproportional among multiple samples. The real network traffic is always imbalanced. For instance, the NSL-KDD dataset (Tavallaee et al., 2009a) contains five intrusion classes with imbalanced network traffic (I.e., benign traffic with 13449 samples, Denial of Service (DoS) attacks with 9234 samples, Probe with 2289 samples, Remote to Local (R2L) with 209 and User to Root (U2R) with 11 samples). When Many researchers considered the NSL-KDD dataset for intrusion detection in IoT because it contains diverse useful information to mark malicious network traffic (Dhanabal & Shantharajah, 2015; K. Singh et al., 2021; Su et al., 2020; W. Xu et al., 2021).

Two common approaches to dealing with such data imbalance problems are the data-driven approach and the algorithm-driven approach (Chawla et al., 2002; Hamad et al.,

2020; H. He et al., 2008; Kotsiantis et al., 2006; More, 2016; H. M. Nguyen et al., 2012). The data-driven methods focus on handling the distribution of the classes by bringing them to an equilibrium state by either oversampling (copying and adding minority samples) or under-sampling (removing majority class samples) (More, 2016; H. M. Nguyen et al., 2012). The data augmentation/generation algorithm is a well-known technique that solves the sample imbalance issue, and a lot of research has been proposed on this area since (Chawla et al., 2002; H. He et al., 2008) years ago. The research on data imbalance issue in intrusion detection on IoT is still a hot research topic. This is because the learning algorithms showed promising intrusion detection performance as they learn from a large volume of data. The performance degrades notably in the case of learning from imbalanced class samples (J. Lee & Park, 2021; Leevy et al., 2021; Parsaei et al., 2016).

Sampling techniques work by simply replicating or synthesizing the observed data. The most common data oversampling algorithm that uses the data-oriented approach is the Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al., 2002). SMOTE works by feature space interpolation and generates synthetic data for the minority class. Some recent advancements in VAE (Osada et al., 2017; X. Xu et al., 2020; Y. Yang et al., 2019), Conditional Variational AutoEncoders (CVAE) (Lopez-Martin et al., 2017), and Generative Adversarial Networks (GAN) (Creswell et al., 2018; Goodfellow et al., 2020) algorithms are applied to solve data imbalance issue by generating synthetic samples. Remarkably, the VAE (Doersch, 2016) and CVAE (Lopez-Martin et al., 2017) have proven proficiency in the intricate representation of visual data, e.g., images and motion pictures. The algorithm-oriented approach highly depends on the cost sensitivity of learning algorithms. The cost matrices can be customized for better learning of miss-classified samples using cost-sensitive learning (Chawla et al., 2004; H. He & Garcia, 2009; Kingma et al., 2014; Kingma & Welling, 2013). The represented cost matrix is used to reduce the

probability of miss-classification.

1.2 Motivation

Many research works on IoT security issues, and challenges have been published. Some focused on classifying security attacks, and some studied autonomic and cryptographic countermeasures to those security issues. There are several review studies on the security of IoT. Some of these studies focused on security challenges, whereas others focused on security solutions based on different techniques and methodologies. The survey by Hassija et al. (Hassija et al., 2019) provided several IoT security challenges and further discussed fog, edge computing, blockchain, and machine learning technologies as the various means of scaling up IoT security. Another survey (Sharma et al., 2020) focused on physical layer security, protocols and handover defenses for mobile IoT. The authors compared the existing security measures for mobile IoT applications. A systematic review study (Liao et al., 2020) investigated hardware and software-based security measures for IoT mobile computing devices. The study (A. Kim et al., 2020) examined insider IoT threats based on various data sources such as IoT deployment environments and IoT architectures. The authors compared different data sources from different IoT layers and investigated the limitations on the potential utilization of the data sources and methodologies.

Some researchers surveyed recent developments in learning-based intrusion detection systems for IoT. For instance, the studies (Al-Garadi et al., 2020; Hussain et al., 2020) reviewed machine and deep learning-based security solutions for IoT and identified the limitations of each method. The authors (Hussain et al., 2020) also provided future research challenges and directions. The existing surveys and reviews on IoT security focused on security challenges and discussed measures, which only focused on a specific type of methodology. Other studies focused on security challenges concerning mobile-IoT, location-based or commercial IoT. Others yet focused on a specific type of security

countermeasures such as either learning-based or cryptographic-based measures. However, the existing studies have not considered providing a comprehensive and up-to-date analysis of intrusion in IoT and security countermeasures within the current trends in encryption methods, learning-based and autonomic approaches.

Secondly, the traditional intrusion detection methods suffer from the following drawbacks. Due to the data imbalance problem, the majority attack class overrides the learning algorithms, and the minority attack classes may not be learned effectively, hence leading to a high FPR, low minority class detection rate, and low overall detection accuracy. This issue could be solved by developing an appropriate data oversampling algorithm. In oversampling approach, new samples are created from the existing data points and added to the low-frequency classes in order to balance the data set. However, most conventional oversampling methods are trained with widely used cost-sensitive learning Cross Entropy (CE) loss function. By utilizing the CE loss function, the majority class samples overwhelm the loss curve, which cannot enhance the quality and diversity of the synthesized minority class intrusion data samples. CE loss function passes equal weights to each data instance, and this leads the model to oversee minority class samples. When training a model with an imbalanced dataset, the accumulation of the small losses over the majority class samples can overwhelm the overall loss. This leads to degenerated models (Fernando & Tsokos, 2021).

Recently, a technique called Focal Loss (FL) has emerged to enhance the power of CE as an alternative cost-sensitive learning to amplify the efficiency of learning algorithms (Aljohani et al., 2021; T.-Y. Lin et al., 2017; Pasupa et al., 2020; Tian et al., 2018). The idea behind FL loss is it down-weights the correctly predicted records and assigns large weight to misclassified records. Hence, the model is able to learn all the class samples more efficiently (T.-Y. Lin et al., 2017; Yun et al., 2019). FL was also implemented in

intrusion detection very recently (Z. Cheng & Chai, 2020; Mulyanto et al., 2021). The authors evaluated the FL objective function and achieved a noteworthy enhancement in the performance of intrusion detection. Likewise, the FL loss performed much better in learning from data than traditional CE loss in computer vision and IoT applications (Awalgaonkar et al., 2020; T.-Y. Lin et al., 2017; Mulyanto et al., 2021).

This research explored the use of the Class-wise Focal Loss (CFL) objective function instead of the conventional reconstruction CE loss in order to develop an appropriate data generation method. With the CFL loss function, this research focuses on the minority class samples more to learn a better representation of data for each class. In the learning process, the class-wise cost-sensitive approach including oversampling approaches, aim to modify and re-weight the minority class samples. As a result, the Variational AutoEncoder (VAE) is able to generate minority class samples as close to the original input, which will, in turn, lead to better performance of our intrusion detector and reduce the FPR of the minority and unknown attacks. Using the FL loss, the generative model learns a better representation of minority class samples and generates high-quality, diverse, and realistic synthetic data to solve the data imbalance problem.

1.3 Problem Statement

The existing research on IoT security focused on security challenges and discussed measures, which only focused on a specific type of intrusion detection system which does not provide the readers with a comprehensive idea (Hassija et al., 2019; A. Kim et al., 2020). Other studies focused on security challenges concerning mobile-IoT, location-based or commercial IoT (Sharma et al., 2020). Others yet focused on a specific type of security countermeasures such as either learning-based or cryptography-based measures (Al-Garadi et al., 2020; Hussain et al., 2020; Liao et al., 2020). However, the existing studies have not considered current IoT attack categories such as multi-layer attacks, and security

measures of IoT in terms of its characteristics and diversities. There is a need to undertake a holistic investigation of autonomic, learning-based, and encryption-based IoT security countermeasures. This research aims at providing a comprehensive and up-to-date analysis of security countermeasures within the current trends in cryptography or encryption methods, learning-based strategies and autonomic approaches. The study also aims to provide useful comprehensive insights and opens a research gateway for future researchers who are interested in IoT security challenges and solutions.

Furthermore, although the existing intrusion detection approaches succeeded with satisfactory performance, they yet suffer from inferior detection rates, and high false-positive rates in low-frequent, minority and unknown attack classes. The majority of network traffic in a real environment is uneven, which means the attack traffic is considerably lower compared to normal network traffic. This leads to a class imbalance problem (R. George & Roy, 2022), and imbalanced class degrades classification accuracy and escalates the FPR of the training model. Some recent research has focused on addressing the data imbalance problem to improve detection accuracy. Many oversampling methods exist, such as Random Over Sampling (ROS) (Hayaty et al., 2020), Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al., 2002), and Adaptive Synthetic Sampling Approach (ADASYN) (H. He et al., 2008), and some recent developments such as Generative Adversarial Network (GAN) (Goodfellow et al., 2014, 2020), AE (Albahar & Binsawad, 2020) and their variations have been implemented to balance uneven real-network dataset for better performance of IDS.

Yang et al. (Y. Yang et al., 2019) have also explored the usage of CVAE for data synthesis for intrusion detection. An improved version of Conditional Variational AutoEncoder (ICVAE) is used to overcome the data imbalance problem, and a Deep Neural Network (DNN) is utilized for classifying intrusions in the system. The learned weights of ICAVE

are used to initialize the operation in DNN hidden layers. The ICVAE-DNN model improved overall detection accuracy. However, the method lacks detection accuracy of minority attack categories, such as U2R, R2L, Probes, and DoS attacks. Additionally, the overall detection accuracy and False Positive Rate (FPR) could be improved. This is due to the fact that they neglected the cost sensitivity of imbalanced intrusion data to generate high-quality synthetic data samples for minority intrusion classes. For instance, the minority samples are so small that the reconstruction loss is dominated by the majority class and the minority class samples are neglected. Hence, the reconstruction of minority class samples deviates significantly from the observed samples. The default CE loss in ICVAE may not be able to optimize the latent distribution in the decoder and may lead to degrading the quality of decoded samples. The CE loss function assigns equal weights to each data sample. When training ICVAE with the imbalanced dataset, the majority class overwhelms the overall loss. Hence, the minority classes cannot be learned efficiently, thus, generating data far from the original data. Therefore, the generated data samples deviate from observed data which leads the classifier to perform poorly. This results in degrading the intrusion detection rates for minority attack classes and overall intrusion detection accuracy.

To overcome these issues, this work proposes a novel intrusion detection framework called CFLVAE-LDNN. CFLVAE-LDNN inherits the strengths of Variational AutoEncoder (VAE) and utilizes improved Class-wise Focal Loss CFL) as an objective function instead of the traditional reconstruction loss (CE) to train the VAE model by replacing the traditional loss function with focal loss. To better apprehend the representation and the property in the observed intrusion data for its minority attack classes, this research designs a novel objective function called CFL and develops an appropriate data generative model using VAE. The model focuses on the minority attack classes and adjusts weights for each class

sample individually. CFLVAE-LDNN framework consists of two models: 1) CFLVAE model is trained to generate realistic synthetic data to balance, and 2) Lightweight Deep Neural Network (LDNN) classification model is used for classifying the attack categories. The proposed technique generates realistic synthetic data for the classifier to provide high detection accuracy. The problem statements (PS) of this research are as follows:

- **PS1:** IoT suffers from security threats, attacks, intrusions and vulnerabilities; and there are limitations associated with present security solutions utilized for intrusion detection in IoT.
- **PS2:** Real network traffic is imbalanced, which leads the learning-based classifier to perform poorly in intrusion detection for minority class attacks.
- **PS3:** Deep learning-based intrusion detection and classification model is not suitable for resource-constrained IoT devices; therefore, it is necessary to establish and evaluate a lightweight deep learning intrusion detection and classification model for IoT.

1.4 Research Objectives and Questions

1.4.1 Research Objectives

This research aims to establish and evaluate a lightweight deep neural network framework for Intrusion Detection Using Class-wise Focal Loss Variational Autoencoder for IoT. The objectives of this research are as follows:

- **RO1:** To identify existing security threats, attacks, intrusions, and vulnerabilities, and to recognize current solutions used for intrusion detection associated with the Internet of Things (IoT) and their limitations.
- **RO2:** To develop a data generation model to balance a intrusion detection dataset.
- **RO3:** To establish a lightweight deep learning model for intrusion detection in IoT.

- **RO4:** To evaluate the performance of the proposed lightweight intrusion detection model for IoT.

1.4.2 Research Questions

IoT security mechanisms could serve as a useful manual of existing security threats in the IoT and educate researchers on numerous solving techniques. It will also serve as a reference point for future research in improving and unifying the IoT security framework. The questions of this research are as follows:

- **RQ1:** What are existing security threats, attacks and vulnerabilities associated with IoT?
- **RQ2:** What current security solutions are used for intrusion detection in IoT?
- **RQ3:** What limitations are associated with present security solutions utilized for intrusion detection in IoT?
- **RQ4:** How to utilize the deep data generation model for balancing IoT intrusion detection dataset?
- **RQ5.** How to overcome the limitation of the current data generation model to solve the data imbalance issue by generating diverse and realistic samples for intrusion detection?
- **RQ6.** How to establish a lightweight intrusion detection and classification model for IoT?
- **RQ7:** How can the generated data improve intrusion detection accuracy?
- **RQ8:** What improvements can be achieved when using the lightweight intrusion detection classification model with a balanced intrusion dataset for intrusion detection in IoT?

Table 1.1 presents the map between the problem statements, research objectives, and

research questions.

Table 1.1: Research Components Mapping

Problem Statement	Research Objectives	Research Questions
PS1. IoT suffers from security threats, attacks, intrusions and vulnerabilities; and there are limitations associated with present security solutions utilized for intrusion detection in IoT.	**RO1.** To identify existing security threats, attacks, intrusions, and vulnerabilities, and to recognize current solutions used for intrusion detection associated with the Internet of Things (IoT) and their limitations.	**RQ1.** What are existing security threats, attacks, and vulnerabilities associated with IoT. **RQ2.** What current security solutions are used for intrusion detection in IoT? **RQ3.** What limitations are associated with present security solutions utilized for intrusion detection in IoT?
PS2. Real network traffic is imbalanced, which leads the learning-based classifier to perform poorly in intrusion detection for minority class attacks..	**RO2.** To develop a data generation model to balance a intrusion detection dataset.	**RQ4.** How to utilize the deep data generation model using VAE for balancing IoT intrusion detection dataset? **RQ5.** How to overcome the limitation of current data generation model to solve data imbalance issue by generating diverse and realistic samples for intrusion detection?
PS3. Deep learning-based intrusion detection and classification model is not suitable for resource-constrained IoT devices; therefore, it is necessary to establish and evaluate a lightweight deep learning intrusion detection and classification model for IoT.	**RO3.** To establish a lightweight deep learning model for intrusion detection in IoT. **RO4.** To evaluate the performance of the proposed lightweight intrusion detection model for IoT.	**RQ6.** How to establish a lightweight intrusion detection and classification model for IoT? **RQ7.** How can the generated data improve intrusion detection accuracy? **RQ8.** What improvements can be achieved when using the lightweight intrusion detection classification model with a balanced intrusion dataset for intrusion detection in IoT?

1.5 Thesis Organization

This research work is categorized into six chapters, as briefed below.

Chapter 2: Literature Review

This chapter presents an overview, architecture and enabling technologies of IoT. This chapter also discusses security issues in different IoT domains and provides a classification of possible security attacks on IoT. Moreover, this chapter explores existing security solutions and their limitations for IoT.

Chapter 3: Methodology

This chapter illustrates the methodology, design, and development of a novel intrusion detection framework. The experimental test-bed is described along with its mathematical derivations.

Chapter 4: Experimentation

This chapter exposes the research problem in more detail through implemented experiments. The experimental test-bed is described along with its experiments carried out are listed.

Chapter 5: Results and Discussion

This chapter presents and discusses the detection performance of the proposed intrusion detection framework and elaborates on comparative studies.

Chapter 6: Conclusion and Future work

Finally, this chapter concludes the study and suggests recommendations and future work.

1.6 Chapter Summary

This chapter has given a brief background to the research, describing the motivation, problem statements, and research questions addressed. It has also set out the aims, and objectives. The original contribution has been highlighted and the thesis structure is outlined.

CHAPTER 2: LITERATURE REVIEW

This chapter presents an overview of IoT, its architecture, enabling technologies and protocols. A number of literature on different IoT application domains and their security issues, threats, attacks, and vulnerabilities are studied. Additionally, a comparative study of conventional Internet security with IoT security is presented, and the limitations and necessities of the security for IoT are discussed.

Furthermore, this chapter explores, addresses, and brings together the extensive and up-to-date security attacks taxonomy. It provides a state-of-the-art taxonomy and analytical comparison of security attacks based on three-layer IoT architecture. Moreover, this chapter explores and compares different intrusion detection techniques, including encryption algorithms and autonomic and learning-based techniques and finally, justifies the suitability of implementing them in IoT. This chapter aims to provide a useful user manual of those security aspects for a heterogeneous IoT environment by discussing a range of possible solutions to guide researchers to improve the security issues in the context of IoT.

2.1 Definition of IoT

IoT is defined as a world of networked smart objects, where every physical "thing" with a digital element is interconnected. An official definition of IoT provided by the international telecommunication union (ITU) is as follows(Sundmaeker et al., 2010):

“A global infrastructure for the information society enabling advanced services by interconnecting (physical and virtual) things based on, existing and evolving, interoperable information and communication technologies”.

A similar definition is also provided by (Amaral et al., 2011). IoT enables the interconnectivity of billions of devices to aid computing and communications. Digital entities

such as sensors, Radio-Frequency Identification (RFID), the Internet and localization technology make it possible to transform everyday objects into smart objects capable of interpreting and interacting with each other (Amaral et al., 2011). The embedded sensors in smart objects monitor, sense, and collect various data about equipment, environment, and human social life (Yan et al., 2014). IoT is not only about connecting or communicating but also exchanging information, which has and will continue to affect every aspect of our daily lives.

The information represents the link between the cyber and physical world with the primary concern for security susceptibility. The connections among humans, devices, sensors, and services are universal and continuous. No matter how meticulously designed, intelligently configured, implemented and properly maintained a security system is, it will have to rely on human intervention. The Human element is the most appealing and overarching security concern as dealing with it is challenging. Therefore, designing cybersecurity solutions without considering the human element is just an illusion (Svensson, 2013). Although innovations and technological developments have made security solutions more impressive, the large scale of security relies upon our hands (Frangopoulos et al., 2013).

Interestingly, IoT enables the interconnectivity of several heterogeneous devices and networks using different communication technologies. According to (Al-Fuqaha et al., 2015; Horrow & Sardana, 2012), communication may occur between machine-to-machine (M2M) or thing-to-thing (T2T), human-to-thing (H2T) or human-to-human (H2H) through different means of connectivity. IoT aims to provide smart and advanced services to its users through information networks formed by consistent integration of physical objects (e.g., personal computers, smartphones, wearable devices, washing machines, fridges, lights, microwave ovens, and medicines). The things are interconnected or connected to

the Internet or humans and are capable of transmitting real-time information about patients, property, traffic, and electricity (Botta et al., 2016). These smart objects are also capable of delivering the collected lightweight data around the globe. Furthermore, devices equipped with actuators can extract data, process them and boost communication efficiency among smart objects.

IoT is distributed and heterogeneous, and therefore, the issues related to security need to be given considerable attention. However, IoT differs from conventional Internet in several contexts, including security. IoT also differs in terms of technology and deployment. IoT devices are connected under the constraints of Low Power and Lossy Networks (LLNs), which are weak in energy, memory and processing capabilities (Botta et al., 2016; Yan et al., 2014). Furthermore, unlike typical IT infrastructure, IoT is globally connected through compressed Internet Protocol Version 6 (IPv6).

The main objective of IoT is to offer integration among software, sensors, inter-operable communication protocols, network infrastructures, and physical objects anywhere and anytime (Aazam et al., 2016; Yan et al., 2014). With the advancement of smartphone technology, an enormous number of objects are capable of being part of IoT. However, the necessity of rapid and large-scale deployment of IoT devices can lead to a significant security concern. The authentication, authorization, system configuration, verification, access control, information storage, and management verification, to name a few, are the main security challenges and issues in the IoT realm (Jing et al., 2014). Embedded devices and smartphones, for instance, offer many digital services, making our lives easy and worry-free. These devices can easily control, operate other devices, and share data from long distances. However, the security of these devices, information and users' privacy is not guaranteed. Users' vital information may leak or tamper with anytime and anywhere.

2.2 Imbalanced IoT Data

The advancement of IoT and big data significantly affects businesses, organizations, and individuals. A large number of communication devices in the IoT produce and transmit from one device to another as network traffic. The network traffic may contain intrusion/attack data in some cases. Data collecting IoT devices sense and transmit data using embedded sensing and communication modules. The volume of data generated by sensors, devices, and different IoT applications is continuously increasing (Khushi et al., 2021; Y. Ma et al., 2012; Taherkordi et al., 2017; **?**). The generated large amounts could be structured, unstructured, or semi-structured. These data are generally voluminous, heterogeneous and generated from different IoT applications with distributed and decentralized settings.

The enormous volume of these data produced by IoT applications could be utilized to develop business analysis systems, business recommendation systems, intrusion detection/prediction systems (J. Gao et al., 2021; Khushi et al., 2021; Y. Ma et al., 2012; Marjani et al., 2017), and so on. However, the collected IoT network data traffic is of different types and comes with higher real-time requirements. These various types of data can be utilized to create manufacturing datasets that often have a dramatically skewed distribution. For example, in an IoT application, the network traffic may contain a huge number of normal/benign data samples and fewer attack samples. As a result, this network traffic creates an imbalanced dataset. An imbalanced dataset refers to an unequal distribution of class samples in a particular dataset (Japkowicz & Stephen, 2002; Khushi et al., 2021; X.-Y. Liu et al., 2008). The class with a more significant number of samples is called the majority class, with fewer samples known as the minority class. Minority class is also known as low-frequency or unknown class. So, the minority class is simply that has the lower frequency in the class distribution of the training dataset (Japkowicz & Stephen, 2002; X.-Y. Liu et al., 2008).

The imbalanced datasets are normally utilized to build predictive models. However, building a predictive or intrusion detection model on an imbalanced dataset would cause a model to perform poorly when predicting classes that appear fewer times in a dataset (Ding et al., 2022; X.-Y. Liu et al., 2008). Because of this, the model would not be able to generalize well to the new data in the low-frequency class (Johnson & Khoshgoftaar, 2019; Telikani & Gandomi, 2021). It is important to look into these imbalanced datasets to build models that aim for high prediction/detection accuracy for minority classes. Many researchers are looking into the imbalanced nature of datasets that are typically encountered in IoT applications.

2.3 IoT Security Concerns

Robust security is essential to provide users with the feeling of privacy over personal information to embrace the blessings of the broad deployment of IoT (F. Li et al., 2016; S. Li et al., 2016). For instance, on January 29, 2018, several cyber-attacks were launched against the top three banks in the Netherlands to make the internet banking service unavailable and block their websites (Hague, 2018). One of the leading aluminium companies, Norsk Hydro, was invaded on March 18, 2019, by LockerGoga (a variant of ransomware) (Briggs, 2019). The ransomware caused a shutdown of automated production lines of the aluminium company in Europe and the USA. National Cyber Security Centre (NCSC) reported that the Labor Party in the UK was hit by massive Distributed Denial of Service (DDoS) attacks on November 12, 2019 (Palmer, 2019). The DDoS attacks flooded millions of message requests, targeted the party's website, and destroyed the campaign tools and platforms of the party.

The surveys conducted (Button et al., 2016; Finnerty et al., 2019, 2018; Johns, 2020; Klahr et al., 2017; Miller et al., 2015; Vaidya, 2019) reveal that security breaches have increased drastically in the past years. Furthermore, they reported the security breaches

and attacks witnessed by large and small businesses in the United Kingdom. Figure 2.1 depicts the rate of security breaches experienced by large and small organizations between 2015 and 2020 in the UK.

In 2015, 90% of large organizations had suffered security breaches; also, 74% of entities of small business organizations reported that they suffered security breaches (Miller et al., 2015). Surveys conducted in 2016 and 2017 (Button et al., 2016; Klahr et al., 2017), listed that 65% of large firms and 33% of small firms identified breaches in the last 12 months, 52% of large, and 68% of small firms suffered security breaches in the past 12 months (Finnerty et al., 2019, 2018; Vaidya, 2019). According to reports conducted in 2018 and 2019, 72% and 61% of large firms and 47% and 52% of small firms experienced security attacks, respectively. 2021 report (Johns, 2020) showed, 64% of large organizations had suffered security breaches and 39% of entities of small business organizations reported that they suffered security breaches in 2020.

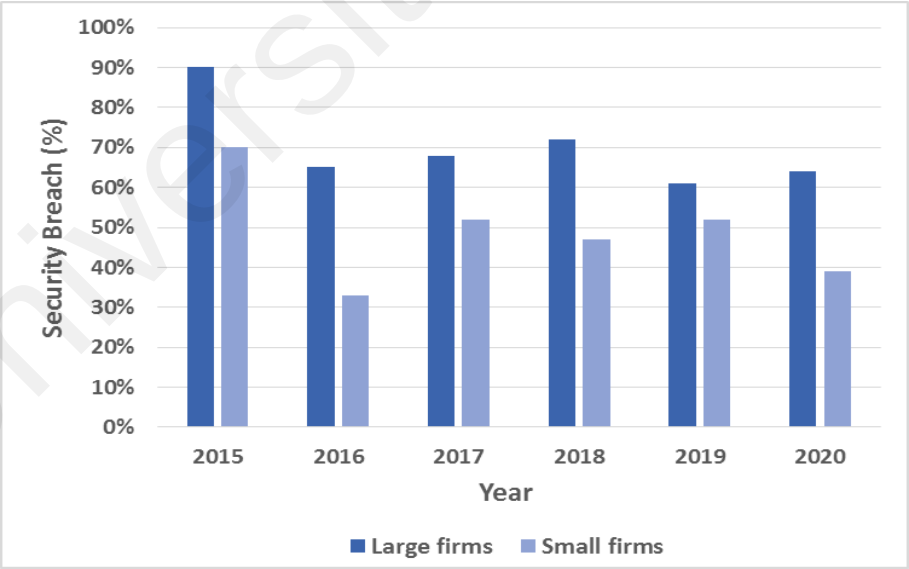


Figure 2.1: The organizations experienced security breaches or attacks in the UK (Button et al., 2016; Finnerty et al., 2019, 2018; Johns, 2020; Klahr et al., 2017; Miller et al., 2015; Vaidya, 2019).

IoT produces a massive amount of data for organizations and businesses, which makes it a target and an alluring venture for adversaries who seek to steal business information for

ransom or other intents resulting in financial losses on the part of the organization. Since IoT is becoming a mission-critical element of small, medium, and large organizations and their businesses, security has become an essential component and a requirement of IoT. It is also evident that security solutions of IoT have improved over time (Sudqi Khater et al., 2019), yet security threats are also evolving in more far-reaching and destructive ways.

Figure 2.2 presents security attack scenarios of some key IoT applications. IoT applications are deployed in almost every aspect of our daily lives, including homes, hospitals and industries. Multiple sensors in an application area (e.g., smart home, smart hospital, smart industry, and smart transportation) communicate with each other and transmit vital information. Considering a scenario where a driver uses a Global Positioning System (GPS) to navigate a destination in order to catch up with an urgent meeting, the car's GPS device will usually be connected to multiple devices and utilizes different networks, which are exposed to cyber-attacks. An attacker can potentially bypass the firewall and may launch a DoS attack, making the navigation service unavailable or sending a wrong signal that misleads the driver. In another scenario based on the same figure, remote operation of the smart home appliances exposes private data to an attacker, or the smart lock of the home could be broken to gain access to home appliances.

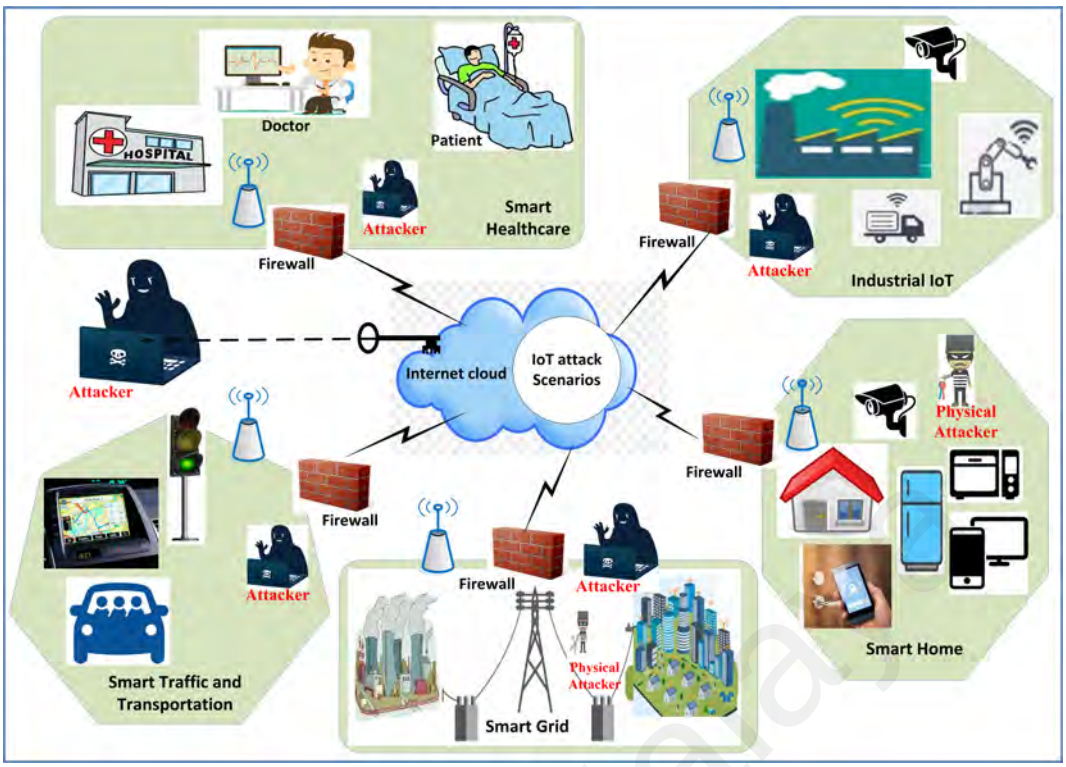


Figure 2.2: IoT Security attack scenarios in different application areas.

In another scenario shown in Figure 2.2, patients get treatment and medication at home or by a healthcare service provider from a remote hospital. However, the patient's sensitive information may risk being stolen or manipulated by the invader who bypasses the hospital firewall, sitting either at the local network or on the cloud Internet. The highlighted scenarios present issues that are related to hacking, terrorism, and sabotage, which could potentially affect large-scale intelligent IoT infrastructures such as electricity, hospitals, offices, industries and buildings.

2.4 IoT Architecture

Despite the wide-ranging opportunities and benefits to users and businesses, there are yet significant IoT security concerns that must be addressed. IoT applications generate a vast amount of data for individuals and organizations, which are prone to security attacks. Since low-power IoT devices are commonly deployed in hostile physical environments, more robust security approaches must be implemented in addition to

conventional Internet security approaches. This section provides an overview of IoT and introduces its architecture.

Given the continuous development and expansion, IoT requires a universal and adaptable architecture that suits its heterogeneity and the diverse scope of its application. Currently, there is no universally adopted architecture. Several researchers have proposed many different architectures for IoT (R. Khan et al., 2012; Sethi & Sarangi, 2017; Z. Yang et al., 2011). This research adopted the three-layered architecture (Alaba et al., 2017; Z. Yang et al., 2011) and outlined the critical concept of IoT. Figure 2.3 presents a typical architecture of IoT, which is divided into three basic layers together with their functionalities. The layers are presented and discussed next.

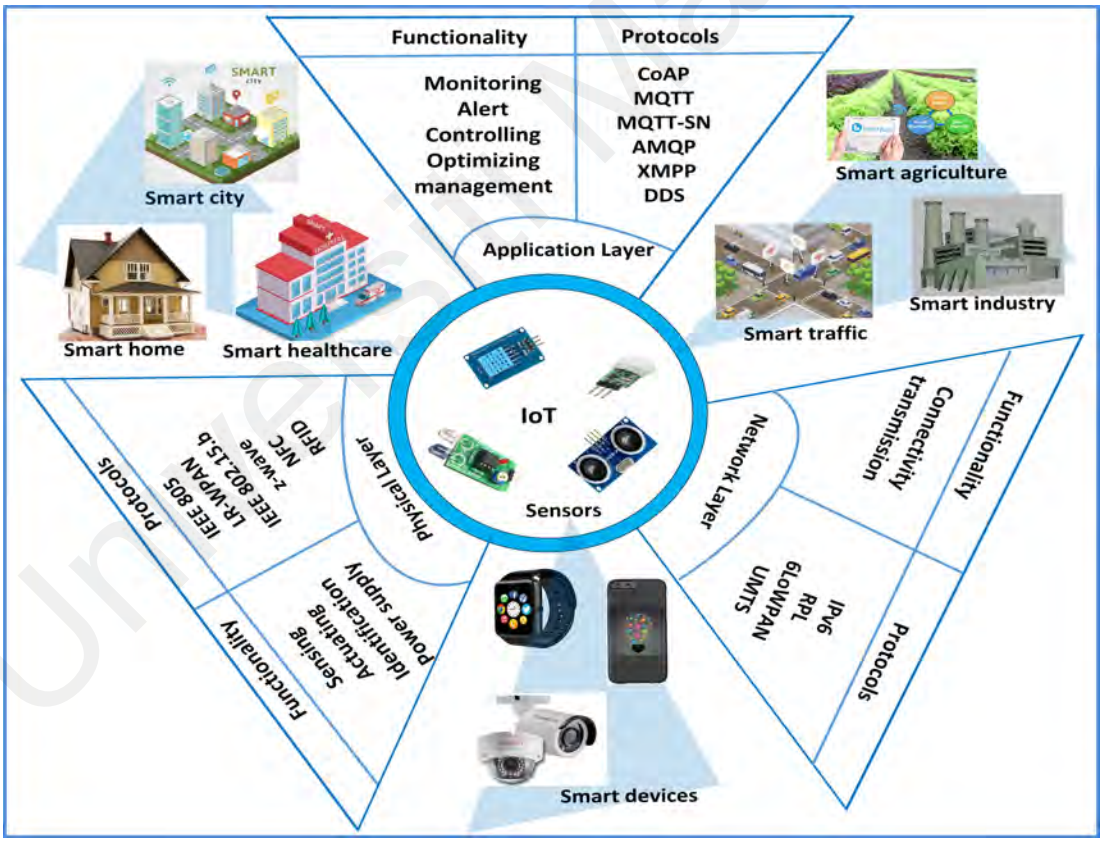


Figure 2.3: IoT functionalities, enabling technologies, and applications in different IoT layers.

2.4.1 Application Layer

This layer consists of an array of smart IoT application solutions (Jing et al., 2014; R. Khan et al., 2012; Z. Yang et al., 2011). The IoT market has enormous potential that attracts the development of smart applications in almost every aspect. Many IoT applications have already been deployed in specific domains such as smart buildings, including homes and offices, smart cities, wearable bands for health monitoring, smart traffic systems, environment monitoring, smart alarm system, and smart personal assistants (Alaba et al., 2017; R. Khan et al., 2012; Sethi & Sarangi, 2017; Z. Yang et al., 2011). The IoT application layer is the highest layer within the IoT architecture, which provides an interface between objects and networks. It offers a variety of functionalities such as data formation, presentation, monitoring of device conditions, notifications, alerts, controlling device functions, management and processing of data, device performance optimization and autonomous operations, providing quality-of-service to end-users (Čolaković & Hadžialić, 2018; M. Wu et al., 2010).

Moreover, a typical application layer includes a service support platform, middleware, computing and communication software (Yaqoob et al., 2017). A survey by (Donta et al., 2021) presented real-time applications, conventional and recent developments in IoT application layer protocols. The main goal of the IoT application layer is to provide different application services to the end-users. Data Confidentiality, Integrity and Availability (CIA) should be guaranteed at this layer by securing applications from unauthorized access, ensuring software/logs integrity and keeping the application service available at any time (Reshan & Saleh, 2021). During the processing of sensitive data, issues such as illegal access and malicious modification of data may arise (W. Zhang & Qu, 2013). This layer could also be susceptible to a number of security attacks such as Spoofing, Message Forging, Viruses and Worm, among others.

2.4.2 Network Layer

IoT Network Layer is comprised of software, protocols, and technologies that enable object-to-object and object-to-internet connectivity (Čolaković & Hadžialić, 2018). It is mainly formed using either local area networks such as the wireless or wired network. This layer also consists of personal area network (e.g., ZigBee), Near Field Communication (NFC), Bluetooth and wide area networks such as Global System for Mobiles (GSM), Long Term Evolution (LTE), 5th Generation (5G) mobile network, and cloud computing (Domingo, 2012; R. Khan et al., 2012; Pongle & Chavan, 2015). The variations of the IoT communication model have been outlined by (Tschofenig et al., 2015) as Machine-to-Machine (M2M) communications, machine-to-gateway model, machine-to-cloud communications, and back-end data-sharing model.

Furthermore, the main function of this layer is to transmit gathered data in the form of a digital signal, which is collected from the physical layer of corresponding platforms via a connected network. This layer is vulnerable to a number of security threats and attacks (Alaba et al., 2017). Common attacks in this layer include Denial of Service (DoS), Sinkhole, Hello Flood, and Blackhole, to name a few. Therefore, it is essential for the network layer to have communication security for secure data transmission over a public network (Airehrour et al., 2016; Naru et al., 2017).

2.4.3 Physical Layer

The bottom layer of IoT architecture is known as the physical layer. In IoT, this layer is also referred to as the perception layer (Alaba et al., 2017; R. Khan et al., 2012; Z. Yang et al., 2011). It includes physical world objects and virtual entities. The main task of this layer is to collect data from the environment through various sensors and transmit them to the network or other devices. Accordingly, IoT devices are embedded with electrical and mechanical hardware components such as sensors, antennas, actuators, and processors

(Čolaković & Hadžialić, 2018). Smartphones, RFID technology (Domingo, 2012; Jing et al., 2014), and wearable devices are capable of processing, identifying, connecting, communicating and storing data. In the perception layer, the sensors or RFID convert the collected raw data of the physical objects to readable digital signals. For instance, IoT objects sense and gathers data from the physical world, such as temperature, humidity, and proximity, to name but a few. However, this layer of IoT is prone to many security attacks such as Jamming, Tampering and Collusion (Alaba et al., 2017).

2.5 IoT Enabling Protocols and Technologies

Numerous protocols connect and work together; thus, an appropriate communication system architecture should be used to ensure inter-operation. Nevertheless, there are still issues with interoperability among diverse network technologies. For example, authors in (Palattella et al., 2016) state that the standardization of the latest progress is the only way to the future development of IoT. Evolving new IoT-based protocols and technologies will play significant roles in the future. The protocols used in conventional Internet for data sharing are not compliant options for low-power IoT constraints. Therefore, there have been standardized protocols for IoT to connect smart things and end-user applications. IoT protocol stack and enabling elements are presented in Figure 2.3. This figure also demonstrates the functionalities of the protocols for each layer of IoT. Some of these significant protocols are presented below.

The Constrained Application Protocol (CoAP) is a widely used application protocol for IoT. On the other hand, CoAPs are the secure version of CoAP which utilizes Datagram Transport Layer Security (DTLS) to protect data between two applications (Čolaković & Hadžialić, 2018; Elhadi et al., 2018; Raza et al., 2013; Sethi & Sarangi, 2017). Next, the Message Queuing Telemetry Transport (MQTT) for Sensor Networks (MQTT-SN), Data Distribution Services (DDS), Extensible Messaging and Presence Protocol (XMPP), and

Advanced Message Queuing Protocol (AMQP) are some other application layer protocols for IoT (Čolaković & Hadžialić, 2018; Elhadi et al., 2018). Moreover, a Quick Constrained Application Protocol Internet Connection (QUIC) is an IoT transport layer protocol (L. Lin Yang, 2018). QUIC was designed by Google to offer security protection, and flow control over User Datagram Protocol (UDP) and to avoid congestion as well as reduce transport latency by using congestion control mechanisms similar to Transmission Control Protocol (TCP). IPv6 is one of the key internet layer protocols appropriate for IoT (Jara et al., 2013; Raza et al., 2013).

Namely, IPv6 offers end-to-end IP datagram transmission for the packet-switched network through multiple IP networks. IPv6 over Low Power Wireless Personal Area Network (6LoWPAN) is a low-power and low-cost communication network that connects IoT devices to the Internet through IPv6. Routing Protocol for Low Power and Lossy Networks (RPL) is a standardized IPv6 protocol for constrained IoT networks. RPL is the IPv6 routing protocol standardized for IoT (Gulzar & Abbas, 2019; Tahir, 2018).

Consequently, IoT is envisioned to integrate different wireless technologies. Bluetooth Low Energy (BLE), Z-Wave, and EPCglobal are some of the IoT physical layer protocols. RFID and NFC (Catarinucci et al., 2015; Y. Choi et al., 2017; Dragomir et al., 2016; Ray, 2018) are technologies for short-range communication for IoT. IEEE 802.15.4 is the Low-Rate Wireless Personal Area Networks (LR-WPANs) (Ray, 2018) utilized for IoT due to security, authentication, encryption, reliable communication, high message throughput, and to accommodate a huge number of nodes (Andrews, 2013). Bluetooth operates in the 2.4 GHz frequency, and it is one of the key technologies for short-range communication.

Likewise, IEEE 802.11 is another physical layer specification for Wireless Fidelity (WiFi) or Wireless Local Area Network (WLAN). The energy consumption is higher in WiFi than that of Bluetooth and ZigBee (Palattella et al., 2016). Cellular technologies such

as 2G (GSM), 2.5G (GPRS), 3G (UMTS/WCDMA, HSPA), 4G LTE, 5G can also be used for IoT communication. As all the protocols for IoT are designed for resource-constrained devices and networks, these protocols could be susceptible to security attacks to a large degree. The constrained devices are vulnerable to attacks from inside the 6LoWPAN network and the global Internet. Therefore, lightweight security solutions are to be developed for these constrained devices and networks (Suo et al., 2012).

2.6 IoT Applications

IoT applications have become an integral part of our everyday lives. These applications are growing rapidly. However, IoT applications suffer from a number of security threats, privacy and trust issues. The type of security threats and attacks vary from application to application, and industry to industry. Nevertheless, effective security solutions should be enforced on different IoT domains based on the nature of applications and functionalities to ensure secure communication and let people enjoy the complete benefit of IoT without compromising their privacy. This section presents some recent application domains and discusses their respective security issues, trust and privacy.

- **Smart Home:** Some modern homes are equipped with smart and automated appliances, such as smart lighting, refrigerator, washer, air-conditioner, electric meters, alarm systems and CCTV. Additionally, these homes are powered by smart cameras, sensors, smart locks, and alarm systems to ensure safety. These smart appliances can be operated through the Internet from long distances. The usage of such smart technologies provides a high level of comfort in smart homes and enhances overall security, trust and privacy while reducing overall expenditure (Komninos et al., 2014; Samuel, 2016). Moreover, to ensure security and to protect smart houses from theft or intrusion, criteria such as confidentiality, auto-immunity, and reliability must be met (Komninos et al., 2014). IoT

devices installed in smart homes should be password-protected, and user login must be confidential.

- **Industrial IoT:** The cyber-physical system is the basis of industrial IoT, which is capable of real-time monitoring, diagnosing and controlling physical processes and production remotely. Meanwhile, IoT-equipped smart industries and factories optimize production processes, enable the manufacture of smart products and provide knowledge-based smart services by utilizing resources with the help of the data gathered by the IoT system. Smart products are usually powered by RFID for digital identity that also can collect and store data (Virat et al., 2018). However, the industrial IoT and its products attached to the digital entity are vulnerable to many security issues such as trust, privacy and confidentiality. They also introduce challenges such as standardization of the production system and social and legal aspects. Consequently, the diverse industrial IoT devices demand highly scalable addressing systems, security solutions and data privacy. Due to resource limitations, the industrial IoT architecture requires low-cost, low-powered infrastructures yet fully integrated with robust security solutions.
- **Smart City:** The concept of a smart city comes from the integration of different IoT applications in various sectors. In a smart city, the integration of multiple services supports its stakeholders in a distributed and dependable manner (Kotsev et al., 2016). However, providing privacy and trust among the stakeholders in the smart city applications remains an important issue. Undoubtedly, there are security issues related to hacking, terrorism,and physical damage which could destroy the infrastructure of smart city applications in areas such as electricity supply, healthcare, corporate offices, factories and traffic systems (Colding & Barthel, 2017).
- **IoT Healthcare System:** It is one of the most prominent and fascinating application areas of IoT (Pang, 2013). The smart hospital-based treatment or remote healthcare

services have gained popularity in recent years. IoT medical services such as distant health monitoring, elderly care, chronic healthcare and fitness programs are some of the potentially rising applications (Baker et al., 2017).

However, a patient's private and sensitive information may be at risk of being stolen or manipulated. Patients' personal information is confidential, and thus, it is important to secure them from exposure to any unauthorized access. Likewise, if the medical report of a patient is leaked and altered, the doctor may end up treating the patient erroneously, which can be lethal and life-threatening for the patient. Therefore, patient data privacy and authentication are of immense importance; therefore, medical applications of IoT should be highly secured (Mathur et al., 2016).

- **Smart Traffic System:** RFIDs and various sensors make urban driving pleasant and traffic management more efficient. Smart traffic applications of IoT give people a sense of 'living in the future. For example, an IoT-enabled traffic system provides route information such as the number of cars in a particular route or lane; parking information such as availability and directions to the parking space; public transport information such as the number of occupants and availability of seats on a bus or train. Similarly, sensors are already used in urban vehicles for safe driving (Alonso et al., 2011). However, the automation of the system may bring security and trust issues for passengers (Stefansson & Lumsden, 2009). As smart cars, buses, and trains, among others, are connected to the Internet, the passengers' data become exposed to the risk of being compromised.
- **Smart Grid:** The smart electrical supply system is known as a smart grid, which is mainly a network of electric transmission lines, transmitters, and substations to distribute electricity across homes and businesses from the power plant in the most efficient way. Smart meters, sustainable energy resources, smart machines and efficient energy properties are some of the power functions of a smart grid (Komninos et al., 2014;

Mahmoud et al., 2015). The use of IoT in electric grids makes the energy distribution and management much more efficient through two-way communication between them and reduces the impacts on climate (Borgia, 2014). Technological improvements in smart electric grid systems increase their security vulnerabilities and threats. Undoubtedly, authentication, confidentiality, trust, integrity, and availability represent the key areas of concern that should be addressed when dealing with a smart power grid.

- **Smart Farming:** Integration of different sensors and RFID technologies make conventional agriculture, animal and fish farming smarter. Various sensors are capable of monitoring temperature, humidity, soil moisture, and microbial contaminants in smart farming (Hassija et al., 2019). Sensors and RFID attached to the animal's body or fish farm are able to monitor health conditions, keep track of their activities and notify the stakeholder remotely. However, smart farming industries are prone to several security issues. The agricultural products can be damaged, or fish and animals can be carted away if the security of such applications is not ensured.

2.7 IoT Security Goals

The security goal or necessity of IoT is discussed in this section. The traditional and common security goals include Confidentiality, Integrity and Availability (CIA). However, apart from this CIA triad, other requirements such as privacy, lightweight solutions, authenticity, and standardized policies have become very important. Figure 2.4 shows the security goals for IoT including lightweight security solutions, privacy and CIA triad. The following security principles should be considered to achieve secure communication for IoT.

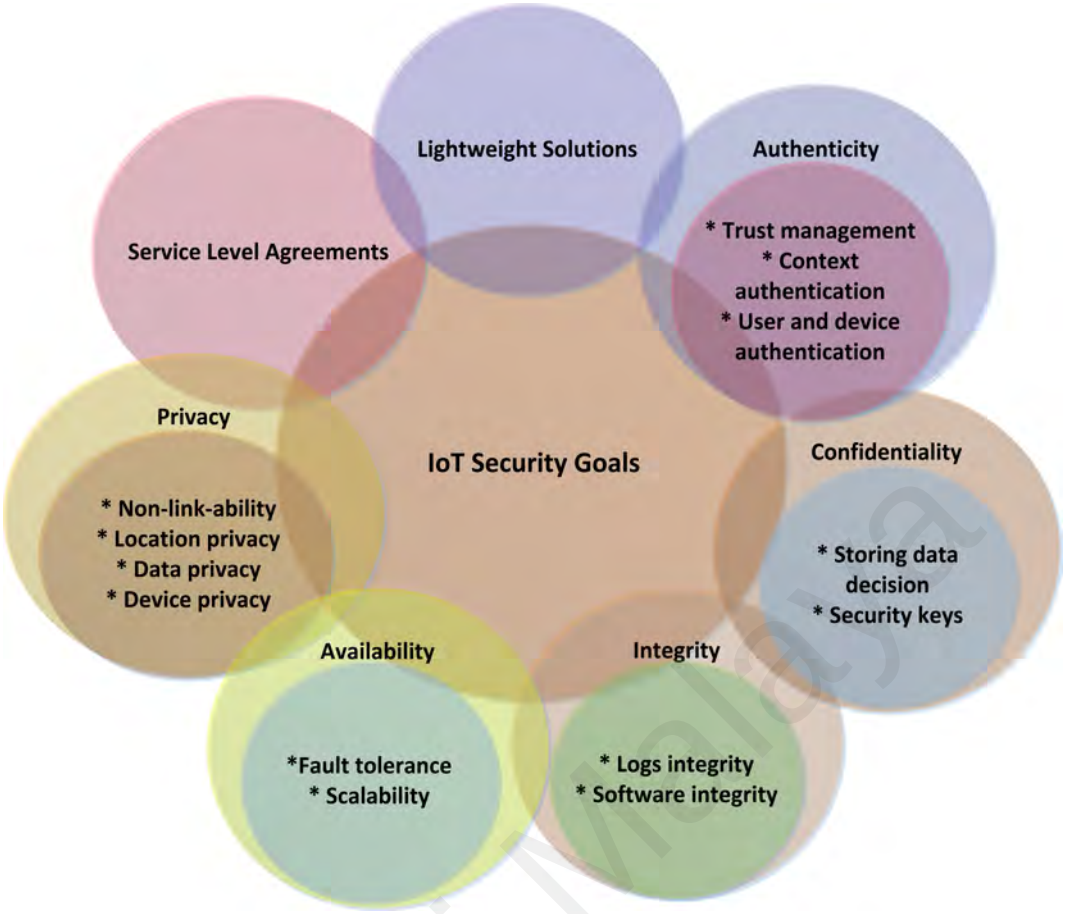


Figure 2.4: IoT security goals.

2.7.1 Lightweight Solution

IoT devices are resource constrained. Hence, conventional heavy security solutions are not suitable for IoT. Therefore, lightweight security solutions can be introduced as a unique feature since IoT devices are considered computationally less powerful and embedded with limited memory (Thakor et al., 2021). The lightweight approach must be considered a security requirement while designing, developing and implementing encryption or authentication protocols for IoT (Mahmoud et al., 2015).For example, RFID tags in e-passports can suffer from un-traceability attacks; hence, lightweight yet robust security solutions must be designed for such ultralight protocols. Furthermore, as the security algorithms or protocols are meant to be run on IoT devices, these must be compatible with the device's limited capabilities (Thakor et al., 2021).

2.7.2 Authenticity

Authenticity validates the legitimate parties involved in communication. It guarantees that the data originates from the actual user it claimed to be from (K. Grover & Lim, 2015). Therefore, a thin and robust authentication protocol is essential. To address the constraints of IoT, it is essential to verify and validate the users involved in communication. A comprehensive review of authentication mechanisms has been presented in (Nandy et al., 2019). Recently, a lightweight authentication mechanism has been proposed for resource-constrained devices (Chuang et al., 2018). RFID tags and NFC are few examples of such advanced innovations, which IoT devices may benefit from as an authentication scheme. An NFC-based authentication mechanism has been proposed by (Petrov et al., 2014) to ensure that energy and processors are not in use at end nodes. However, conventional authentication methods may not be suitable for IoT. For instance, literature (Mahalle et al., 2014) suggests a group-based authentication method for IoT. The method validates the authenticity of all the devices which take part in the communication. Biometric technologies such as fingerprint and face tracking authentications are also not considered secure for IoT devices (Ren et al., 2013). Other than these, trust management, data, device, and user authentication are also important. The following subsections briefly elaborate on some authentication requirements for IoT.

- **Context Authentication:** Obtained sensed data and control information, functional properties, and states of the devices are to be authenticated as a prerequisite. Monitoring also may include device faults detection, configuration changes, and collecting and monitoring performance data. In this case, identify and authenticate the devices that participate in monitoring such services (Battat et al., 2014).
- **Trust Management:** Trust management is essential as it reduces risk factors and allows customer acceptance. Adaptive routing in the smart grid is an entirely trust-

based scheme for IoT components which is reported in (Xiang et al., 2014). Trust management does not only contribute to IoT security, but it also improves the overall network performance (D. He et al., 2012). There are different data aggregation algorithms or machine learning approaches available to obtain trustworthy data in IoT (Yan et al., 2014).

- **User and Device Authentication:**IoT devices and the central unit should autonomously be able to authenticate a user identity that is demanding a certain action. In this process, a single-sign-on mechanism can be applied as once authenticated; the users may interact with several devices. Likewise, ensuring data flow is generated by a certain entity and the device authentication is required. Traditionally, public key cryptography is widely used for such purposes. However, it is a complex framework and too heavy for light-duty nodes (Huang et al., 2016).

2.7.3 Confidentiality

Confidentiality is one of the key features for securing IoT. All information must be protected from unauthorized nodes during any transmission (Nasiri et al., 2019). This assures that only those who have authority to access certain information can view them and it also guarantees that transmitted information is not revealed, viewed or understood by any third-party users as data passes through intermediate nodes (Mahmoud et al., 2015; F. Muhammad et al., 2015). This can be done by using a shared key, where both sender and receiver use this key to encrypt and decrypt data. The following subsections briefly elaborate on some confidentiality requirements for IoT.

- **Storing data:** The autonomic decision should be made to protect confidentiality during storage of vital data locally and in the cloud.
- **Security Keys:** The underlying system should constantly be able to use a self-

protection mechanism to monitor and manage security keys where message or communication confidentiality is vital (Nasiri et al., 2019). In addition, the system should be able to use a self-healing mechanism if any security breach occurs and switch to a failsafe mode or generate new keys. Developing an autonomic system when it comes to confidentiality is challenging due to the constrained nature of IoT (Ashraf & Habaebi, 2015; Nasiri et al., 2019; Thakor et al., 2021). The organization and communication aspects should also be taken into consideration. While comparing symmetric cryptography, the asymmetric cryptographic schemes are more resource-consuming for IoT. A few lightweight encryption approaches are still under development. An adequate research effort is required to overcome the challenges to support the autonomic version of such key management schemes. Symmetric key schemes may support IoT with acceptable overhead (Ashraf & Habaebi, 2015; Thakor et al., 2021).

2.7.4 Integrity

Data integrity ensures that the information remains unchanged during transmission (Mahmoud et al., 2015; F. Muhammad et al., 2015). A symmetric cryptographic algorithm is typically used to help data under transmission by creating signatures for them. Another approach, namely, Message Integrity Check (MIC), is used to verify the integrity of received data (Wara & Yu, 2020). However, such typical cryptographic solutions require a huge amount of energy and bandwidth to operate. An autonomic security solution may provide an acceptable level of data integrity for IoT regardless of inadequate resources (Ashraf & Habaebi, 2015; Nasiri et al., 2019). The autonomic decision-making integrity components are as follows.

- **Logs integrity:** The intrusion detection system should have the ability to expose the

path by generating activity logs in case any changes were made by anyone. These logs are also to be reserved either locally or centrally for a short-term or long-term basis (Ashraf & Habaebi, 2015).

- **Software Integrity:** Tiny software runs on IoT devices. That software integrity also to be ensured by the intrusion detection mechanism. The mechanism must be able to monitor if the device is seized or network is flooded with messages by an adversary (Ashraf & Habaebi, 2015).

2.7.5 Availability

Data availability refers to continuous access to the system data. It guarantees that the entire system, its components, functional properties, and required services are available at any time. The availability of these services and components may hamper due to security attacks (Mahmoud et al., 2015; F. Muhammad et al., 2015). For example, DoS, malware, and Jamming are the common types of availability attacks in this regard. Such attacks may physically harm IoT nodes and networks. The malicious node makes constant queries to an IoT device to launch this kind of attack, which make the device more functional and due to that, the battery-driven IoT devices may run out of power. These issues require proper attention and a robust security model for IoT system is needed to prevent and recover from such attacks on the availability. The connected things should be available and functional whenever they are required (Nasiri et al., 2019). The following security goals on availability must be considered for constant data and system availability.

- **Fault Tolerance:** The system must be able to use the self-protection approach along with self-healing in case of a failure or an attack (Nasiri et al., 2019).
- **Scalability:** The system must be scalable in terms of adding new resources. IoT nodes can be organized hierarchically to support scalability; however, RPL is not

a hierarchical protocol. The packet flow can be centralized to achieve this feature (A. Ahmed et al., 2017). The system must make an autonomic decision on duty cycling on the network on when to switch off or on without losing functionality (Ashraf et al., 2014).

2.7.6 Privacy

Privacy refers to the state or condition in which data or service is meant to be accessed by an individual. This is a hot research topic and a lot of research has been conducted on conceptual security frameworks for privacy issues (Kalloniatis et al., 2008; Y. Yang, Wu, et al., 2017). However, as different IoT layers and systems require different privacy requirements, these solutions are appropriate for issues only at the application layer. To keep the nodes scalable and to consider various IoT applications, a robust privacy policy is required to be developed, which should match with individual node identification and consider providing some level of control to the user. IoT devices are equipped with RFID tags, which can be tracked easily. The privacy of those devices should be protected. Some aspects of privacy include location, context privacy, non-link-ability, anonymity trust management, and identity management (Vidalis & Angelopoulou, 2014). Several research works have been investigated to provide privacy for IoT (Ali et al., 2020; Garms & Lehmann, 2019; Gu et al., 2020; Kortesniemi et al., 2019; Vijayakumar et al., 2019). The privacy goals are categorized as follows (Ashraf & Habaebi, 2015).

- **Non-Link-Ability:** It refers to specified private data that is not linkable to any user. Unauthorized users should not be able to create a profile from the personal data of other users. An attacker may not be able to search for patterns to reverse engineer any sniffed data. The group-signature-based approach was reported to tackle non-link-ability problems in (Garms & Lehmann, 2019).

- **Location Privacy:** The intrusion detection system should guarantee that the location of a device is not exposed to an unauthorized person. The authors (Vijayakumar et al., 2019) proposed an effective technique for preserving location privacy. The technique is anonymous authentication for wireless body area networks which ensures lower computational cost.
- **Data Privacy:** Wearable devices connect the human body to the Internet, thus personal information (e.g., healthcare) should be kept secured.
- **Device Privacy:** RFID tags make the sensor nodes to be traceable and identifiable. Anonymous communication is required to hide the identity of devices for resource-constrained communication protocols. The authors (Kortesniemi et al., 2019) proposed a decentralized identifier-based method to provide privacy for IoT devices. The authors claimed that the model could be deployed in small IoT devices.

2.7.7 Service Level Agreements

In order to protect and transmit data in an efficient way, there must be standardized policies and mechanisms to enforce the policies. It is also important to ensure that the standards and policies are applied to every entity in the network (Girs et al., 2020; S. Li et al., 2020). All services should clearly identify a Service Level Agreement (SLA), which is one way of maintaining the policies and standards. Considering the nature of IoT, the classical SLAs may not be applicable; thus, there should be an autonomous decision on policies to meet SLAs according to diverse services (Girs et al., 2020). These policies are to be enforced in order to foster trust in the IoT paradigm.

2.8 Security Threats, Attacks and Vulnerabilities in IoT

Security attacks may lead to millions of dollars in losses to large businesses and intellectual property theft. The major security threats, challenges, methods of security

attacks and actual security attacks are presented in the following sections.

2.8.1 Security Threats or Challenges

This section provides the security challenges while implementing security in IoT for application, network and physical layers. The representation of different types of attacks based on the properties of IoT assets and their available solutions are provided in this section. The adversary may be an insider or outsider of a network and can be a threat to these assets, such as communication channels, a protocol stack, devices, and personal information (HaddadPajouh et al., 2021). Based on device, network, location or other properties, the adversary performs malicious activities to interrupt IoT services, obtain unauthorized access or physically damage the device. The following sections provide the taxonomy of types of security attacks based on IoT assets and their properties according to the literature (Hossain et al., 2015).

2.8.1.1 Threats based on Device Property

IoT devices are heterogeneous. Therefore, an invader may attack IoT devices based on device properties. Two such methods are given below.

- **Threats on Low-End Devices:** Devices with low memory, power and computational capabilities are considered low-end devices. The attacker uses such devices to launch attacks on other IoT devices (Ojo et al., 2018). For example, an adversary may utilize low-end IoT devices such as a smartwatch to launch attacks on a smart TV or smart refrigerator to threaten privacy, integrity or confidentiality (Hahm et al., 2015).
- **Threats on High-End Device:** A high-end device refers to a powerful and fully-functional device (Ojo et al., 2018). An adversary may launch attacks using high-end devices (i.e., PC, laptop) in order to gain access and cause damage to IoT devices and networks from anywhere.

2.8.1.2 Threats based on Location Property

IoT devices are connected globally and are prone to attacks from the Internet or within 6LoWPAN networks. The methods of such attacks are as follows (Bairagi et al., 2016).

- **Internal Threats:** An adversary's attack from a local network either using his/her own device or a compromised legitimate device. Such attacks may include routing attacks, namely Flooding, Blackhole, and Sinkhole attacks (HaddadPajouh et al., 2021).
- **External Threats:** Initiating an attack on IoT devices or networks, the attacker might be deployed outside and far from a native network. Examples of such security challenges are Brute-force, malware, Secure Sockets Layer (SSL), and Domain Name System (DNS) attacks (HaddadPajouh et al., 2021).

2.8.1.3 Threat Level

An adversary may attack IoT devices or networks at different levels, such as active attacks to disrupt usual functionality or passive attacks in order just to acquire vital information. The security challenges based on threat level are described below.

- **Active Threats:** An attacker initiates direct attacks to interrupt the regular service-ability of IoT networks or devices are known as active security challenge (Uthumansa & Shantha, 2020). DoS, and Blackhole attacks are two examples of such attacks (Alaba et al., 2017).
- **Passive Threats:** These types of attacks are launched to gather important information from IoT networks and devices, but the normal functionality of a device or network is not disrupted (Uthumansa & Shantha, 2020). These attacks are initiated to disrupt the privacy of IoT, such as eavesdropping, and monitoring of data transmission.

2.8.1.4 Threat Strategy

An attacker may belong to different interest groups. Therefore they may attack the IoT device or network using different strategies based on their interest levels as follows.

- **Physical Threats:** The attacks are launched in order to cause physical damage to IoT devices or change device configurations. Malicious Node Injection and Tampering are examples of physical attacks (Andrea et al., 2015; Deogirikar & Vidhate, 2017).
- **Logical Threats:** The attacks are initiated in order to make IoT devices or networks dysfunctional without doing any physical damage to them. Traffic analysis of the communication channel is an example of a logical security challenge (Khanam et al., 2020).

2.8.1.5 Damage Level

IoT devices, networks, and applications are prone to a multitude of security attacks, which may cause different levels of damage. They may range from information leaks and service disruptions to physical damage to the IoT device. Two such threats are provided as follows.

- **Service Unavailability:** Service shut down, or power outage may occur naturally. However, resource exhaustion may occur from DoS attacks, which in turn makes service unavailable (HaddadPajouh et al., 2021; Khanam et al., 2020). Service may be interrupted by such attacks. Thus, recovery mechanisms for such interruptions should be available (Suo et al., 2012). Such attacks can be detected using an effective intrusion detection system.
- **Interruption:** In this type of threat, an invader sits between two IoT nodes, intercepts the communication and tricks them by communicating with both (Tertytchny et al., 2020). In other words, the attacker listens to the private messages which

are transmitted through private communication links. Eavesdropping, Alteration, Fabrication, and Man-in-the-Middle (MitM) attacks are examples of such kinds. These attacks may mislead or create confusion among IoT users (Alaba et al., 2017; Khanam et al., 2020). The intruder may alter or fabricate additional data. Such security challenges can be made either externally or internally, and messages lose their integrity. Such attacks threaten message confidentiality. RFID devices are vulnerable to such security challenges (Laurie, 2007).

2.8.1.6 Host-Based Threats

The devices used in IoT are embedded with software that may contain private information, cryptographic keys and other sensitive information. The data can be targets of the attackers. Some of these attack methods are as follows.

- **User Credential:** An adversary may trick a user into discovering their personal credentials, such as usernames and passwords. User credentials should be protected or shared in a secured manner (Khanam et al., 2020).
- **Software Compromise:** IoT devices and their embedded software are not much powerful. Therefore, the operating system and other software might be vulnerable to security threats (HaddadPajouh et al., 2021; Khanam et al., 2020). An adversary may take advantage of that and compromise the embedded software.
- **Hardware Compromise:** An adversary can damage IoT devices by extracting hardware credentials such as keys, data, or program code that are embedded in the devices (HaddadPajouh et al., 2021). Physical access is usually required to initiate such attacks. IoT devices should be tamper-resistant in order to remain protected from such attacks.

2.8.1.7 Protocol Level Threats

Malicious attackers compromise standard protocols of IoT devices and networks in order to disrupt communication among the devices (HaddadPajouh et al., 2021). Examples of such attacks include the following.

- **Protocol Deviation:** An adversary breaches and diverges from standard communication or application protocols and becomes an insider in order to launch attacks (HaddadPajouh et al., 2021).
- **Protocol Disruption:** An intruder may disrupt standard protocols such as synchronization, data aggregation or key management protocols from inside or outside a network (HaddadPajouh et al., 2021).

2.8.2 Layer-based IoT Security Attack Taxonomy

IoT architecture comprises different technologies which work independently to make a complete system. In section 2.3, we explored the three-layered IoT architecture. In this section, we classify IoT attacks based on the three-layered architecture that consists of application, network and physical layers. The following sub-sections present the proposed attack taxonomy, which is summarized in Figure 2.5.

Some attacks are categorized as multi-layer/dimensional attacks as they exploit more than one layer of the IoT architecture; for instance, DoS or cryptanalysis attacks may take place in the application, network and physical layers of IoT. Tables 2.1 and 2.2 provide an analytical comparison of different attacks in different IoT layers, the method of launching them and the impact of those attacks on IoT.

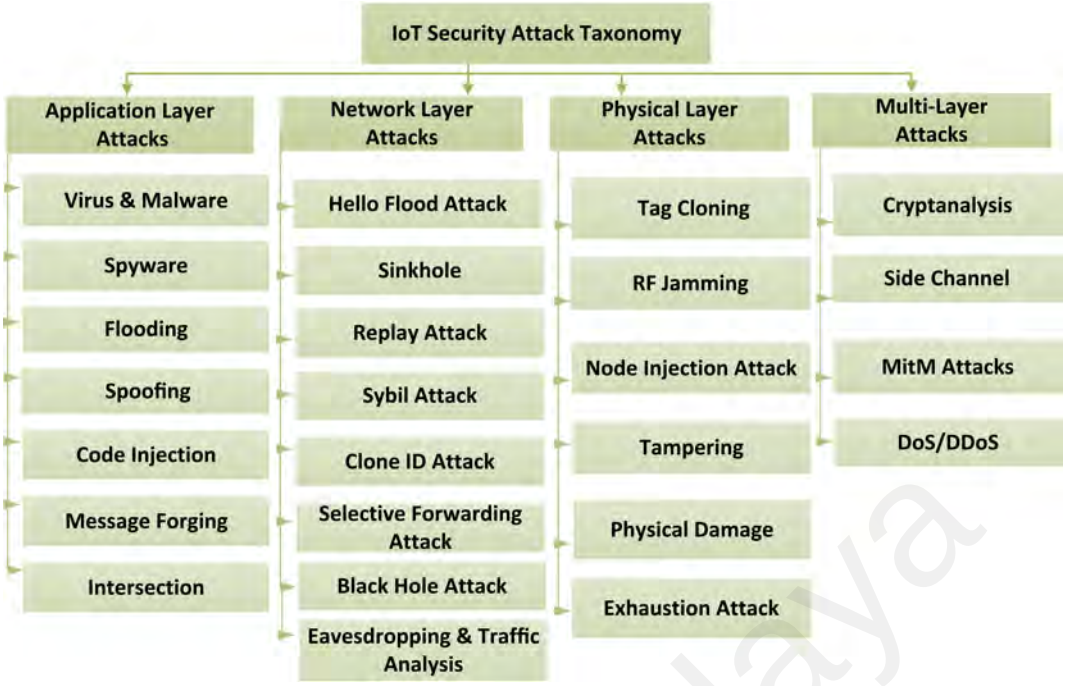


Figure 2.5: Layer-based IoT security attack taxonomy.

Table 2.1: Summary of Different Attacks

Layer	Attack names	Objective	Methods of attack	Impact
Application Layer	Virus & Malware (Tahir, 2018)	To hack and attack confidentiality of application, steal user credential and system shutdown	In the form of trojans, Spams, Worms	Cause damage or harm to IoT high-end device, applications and Bluetooth technologies
	Spyware (Tahir, 2018)	To spy or monitor users' activities and gain users' credential	In the form of the application installed in the user device	Indirect harm to users or device
	Flooding (Aluvala et al., 2016)	To exhaust node resources	By broadcasting a multitude of messages	Reduces device lifetime
	Spoofing (Schaffer et al., 2012)	To hamper authentication and user privacy	By impersonating a node	May cause losing trust and confidentiality
	Message Forging (Shim, 2019)	To send wrong information to the user	By modifying or creating a message	Mislead user by different message other than the original may cause great harm
	Code injection (F. Muhammad et al., 2015)	To steal user ID and password	By injecting malicious code into an application	Hack into users vital account
	Intersection (Lu et al., 2015)	To hamper system privacy	By gaining the system's secondary information	May lead to other attacks
Network Layer	Hello flood (Aluvala et al., 2016)	To mislead routing path	By broadcasting many invalid routing paths	Hello messages from intruder may exhaust system
	Sinkhole (Y. Liu et al., 2018)	To launch several other attacks	By making the central node of the network unavailable	Network failure
	Replay (Elsaeidy et al., 2020)	To exhaust network/ system/database resources	By re-transmitting packets	Network failure or system unavailable
	Sybil (Tandon & Srivastava, 2019)	To eliminate original and valid node from the network	By creating its own numerous identities	Lead to dropping packets
	Clone ID (Morales-Molina et al., 2021)	To gain and access user traffic	By cloning identity of a legitimate node	Missing of user data

Table 2.2: Summary of Different Attacks (cont... Table 2.1)

Layer	Attack names	Objective	Methods of attack	Impact
Network Layer	SF (Patel & Jinwala, 2022)	To deprive bandwidth and delay network transmission	By dropping certain incoming packets	compromising of availability and confidentiality
	Blackhole (Sivaganesan, 2021)	To affect network operation	By dropping all incoming packets	The entire network may fail
	Eavesdropping & Traffic analysis (Zou & Wang, 2015)	To gain information to launch other types of attacks	Gain information by sending control message and make an analysis of the gained messages	Affect user privacy and confidentiality
Physical Layer	Tag cloning (F. Muhammad et al., 2015)	To make the victim confuse about genuine tags	By replicating data from direct access to RFID device or by reverse engineering	To hamper the authenticity of an object, cause financial loss, jeopardize personal safely
	RF Jamming (Salameh et al., 2018)	To make the sharing bandwidth ineffective	By interrupting radio frequency and by making interference	May cause interference and noise in the signal. May lead to DoS attack
	Node injection (Deogirikar & Vidhate, 2017)	To take part in communication among the legitimate node and may	By deploying additional node in a network topology	Take control of the network traffic
	Tampering (Pathak et al., 2021)	To modify, add or delete data from end device	By physical capture and compromise of an end node	To hamper confidentiality and accessibility
	Physical damage (Deogirikar & Vidhate, 2017)	To deactivate the network or to make service unavailable to the user	By removing node physically or deactivating a node by sending kill command	Shutting down a network node makes service unavailable to the user
	Exhaustion (Ashraf & Habaebi, 2015)	To exhaust network resources	By launching other attacks such as retransmission, flooding, repay attack.	Reduces node and network lifetime.
Multi-layer Attacks	Cryptanalysis (Andrea et al., 2015)	To find encryption key	In the form of trial and error by guessing every possible key	Break encryption system and gain access to ciphertext
	Side Channel Information Attacks (Gnad et al., 2019).	To recover key information	By time, power, fault analysis of a system	Lead to other attacks
	DoS (Y. Lee et al., 2017)	To make service unavailable	One way is to attack by exhausting network	Service unavailability may cause serious damage to large organization

2.8.2.1 Application Layer Attacks

Since global standards and policies are yet to be established for IoT to govern the development and interactions for IoT applications, the IoT application layer is still susceptible to many security attacks. Diverse applications of IoT use different authentication techniques, which makes it difficult to integrate them in order to ensure authentication and data privacy. The number of applications is growing, and a huge number of devices are being connected that will share a tremendous volume of data. Applications, which analyze those data or information, may have a large overhead, and service may become unavailable due to security attacks. The major attacks on the IoT application layer and their impacts are described below.

- **Virus and Malware:** These attacks are targeted at the system with the goal of breaching confidentiality. They usually occur in the form of applications such as Trojans, spams, worms or other viruses (N. K. Gyamfi & Owusu, 2018; Tahir, 2018). In IoT networks, smartphones, sinks or gateways, and other high-end IoT devices are significantly at higher risk of these kinds of attacks than sensor-based motes. Furthermore, Bluetooth technologies such as 802.15.4 enabled devices are at high risk (Ashraf & Habaebi, 2015). Therefore, mitigation of such viruses and malware in IoT applications must be taken into serious consideration.
- **Spyware:** Spyware is a program that is installed on users' IoT devices without the users' consent. The main goal of this attack is to spy on or monitor users' behavior and gather sensitive information such as user IDs, passwords, keystrokes, and credit card information. Spyware generally does not cause any damage to the IoT devices or users directly; it mainly steals private information and sends it back to the distributor (Tahir, 2018). The information is then used as the basis for marketing analysis or pop-up ads. Traditional Spyware detection approaches are signature, behavior,

and specification-based techniques. Signature-based techniques detect only known Spyware; therefore, unknown Spyware instances remain unattended (N. K. Gyamfi & Owusu, 2018).

- **Spoofing:** An attacker may impersonate a node to launch a spoofing attack. A spoofing attack is one of the high-risk attacks due to its attacking method. With a suitable portable reader, a transmission might be recorded. As the attacker impersonates the node, the re-transmission might appear from a valid node. This attack may exist in all three IoT layers. Spoofing attacks by impersonating nodes are categorized as attacks of authentication, and it also violates the privacy principle (Schaffer et al., 2012).
- **Code Injection:** An attacker inserts malicious code into a smart application/system by misusing faulty programs (F. Muhammad et al., 2015). The attacker launches such attacks in order to gain access, steal users' sensitive data, take over the system control or transmit worms (W. Zhang & Qu, 2013). Code injection attacks may take place in a variety of forms, such as HTML script injection and shell injection. This attack may compromise users' privacy, or a system may lose control, resulting in a total system shutdown.
- **Message Forging:** This attack occurs when a malicious node modifies or creates a message to deliver contents other than the original (Shim, 2019). It can be classified as a type of Replay attack in the case of modifying information synchronization.
- **Intersection:** This attack is also known as a composition attack. It targets the system's privacy by gaining secondary information from the system (Erdin et al., 2015; Lu et al., 2015). The attackers gather such information from third-party sources or public records (Ganta et al., 2008). The adversary targets and makes use of the non-linkable element. The anonymized data of the privacy information from

different sources are then being used to link them.

2.8.2.2 Network Layer Attacks

IoT network layer communication differs from conventional Internet due to M2M communication between heterogeneous devices. As a result, this layer may suffer from security compatibility issues and is prone to different security attacks such as Hello Flood, Sybil and Blackhole attacks. Examples of such attacks are as follows.

- **Hello Flood:** Message flooding is one of the major attacks in the network layer where an attacker aims to exhaust network or node resources such as battery or bandwidth by sending multiple route establishment requests (Aluvala et al., 2016; Pongle & Chavan, 2015). Destination Oriented Directed Acyclic Graph (DODAG) Information Object Message, namely DIO, is used for advertising information about destination/root that is used to build the topology of RPL.

  Any node that receives a hello message considers that it originates from within the network and marks it as a communication route. In this case, an attacker or intruder whose intention is to place his/herself as a neighbor of other nodes in the network may convince other nodes that it is a normal node. It means the attacker node will broadcast a hello message to all nodes on the network to let them know that the attacker is a neighboring node (Pongle & Chavan, 2015). This may lead to bandwidth and network throughput inefficiency as the attacker drops the incoming packets; therefore, that packet(s) will be lost. Hello Flood attack may also result from unequal transmission areas. It is considered as a low-impact attack.
- **Replay Attack:** This attack commonly occurs during synchronization to mislead the destination node such that a malicious node stores transmitted information and only to re-transmit it at a later time. Missed frame re-transmission request is usually made

by transmitting packets repeatedly across a network with the sequence numbers to senders and receiver nodes. For example, it may occur during communication between an RFID reader and a tag (Elsaeidy et al., 2020; Mitrokotsa et al., 2010b). This attack exhausts network/system resources such as RFID and back-end database resources (memory, battery and processor).

Additionally, the adversary may broadcast the radio signal in order to gain reader grant access (Mitrokotsa et al., 2010a, 2010b). Replay attacks are classified as high-risk attacks, but they can be mitigated and prevented relatively easily. However, network efficiency will drop if the mitigation of this attack fails.

- **Sinkhole:** In this kind of attack, an attacker trespasses and compromises a central node of a network in order to make it unavailable which leads to packet dropping as well as DoS attacks. The risk level of sinkhole attacks is very high, where a large number of nodes can be compromised (Y. Liu et al., 2018). Regarding the infrastructure-based system, the sinkhole attacks could control the whole network.
- **Sybil Attack:** Sybil Attack is launched by creating a node and presenting its own numerous identities in the network in order to gain huge influence, which in turn leads to the elimination of original active nodes from the routing table. Here, the system's weakness depends on a few factors such as the ease with which those multiple identities are created, and the level of influence to which the system agrees to take inputs from a trusted entity, which is not linked to a chain of trust. A survey on Sybil attacks and its available defense mechanisms for IoT is presented (Tandon & Srivastava, 2019; Y. Zhang & Pengfei, 2014). Based on the attacker's skills, the authors categorized Sybil attacks into three different types, namely, SA-1, SA-2, and SA-3.
- **Clone ID**: The name implies that the adversary clones the identity of legitimate

IoT node in order to gain access to user data traffic(Morales-Molina et al., 2021; Pongle & Chavan, 2015). The malicious clone node can be identified by storing the geographical location and identity of each node at 6BR (6LoWPAN border router). It can also be traced using a distributed hash table.

- **Selective Forwarding (SF) attack:** In SF attacks, a malicious attacker enters into a network and drops selective packets. The adversary casually drops some packets and selectively forwards some to the next node. IoT networks are lossy by nature; therefore, it is difficult to identify the real reason for packet dropping (Mathur et al., 2016; Patel & Jinwala, 2022). This may lead to bandwidth deprivation and delay in the entire network (Bysani & Turuk, 2011). This can result in compromising availability and confidentiality. Possible solutions to this attack may include redundancy checks and probing. Some solutions focus on providing network complete recovery, whereas others try to lessen the damage being caused (Bysani & Turuk, 2011).
- **Blackhole Attack:** During a Blackhole attack, the malicious node drops all the packets that it encounters and the entire network operations get affected. This attack is classified as a high-impact attack as it absorbs all routing information. An intruder floods out malicious routing information to claim the best route to the destination (Pongle & Chavan, 2015; Raza et al., 2013; Sivaganesan, 2021). The sender then chooses the malicious route to transmit the packets. The attacker frequently sends fake route-reply (RREP) to the sender. The source node keeps transmitting its packets through the malicious route, the attacker drops all the packets, and he/she does not forward any traffic to the destination.
- **Eavesdropping/Traffic Analysis:** These attacks can be active or passive. They act as prerequisites for other types of security attacks. A network is usually unaware of the existence of such attacks (Dai et al., 2013). In active eavesdropping, an attacker

transmits a control message to initiate the attacks and the replies from the destination device are analyzed further to pave the way for other attacks. Passive eavesdropping, on the other hand, overhears the communication traffic to extract vital information from the transmission medium to launch other attacks. These attacks may affect users' privacy and data confidentiality. Information can eavesdrop at either M2M, network or cloud layers (Zou & Wang, 2015). Eavesdropping attacks are relatively easier on the M2M layer; however, the attacker can overhear only a selected part(s) of the system, and in most cases, raw data is not as useful (Rabbachin et al., 2011). IoT devices on a wireless medium are greatly vulnerable to such attacks. MitM attack (Conti et al., 2016) is one of the examples of an active attack, where the attacker acts as a router and connects with both sender and destination nodes independently and transfers information between them. The vital information is captured to analyze further and modify.

2.8.2.3 Physical Layer Attacks

The main components of the physical layer are sensors, RFID tags, Wireless Sensor Networks (WSNs), cameras, and so on. This layer of IoT suffers from a number of security attacks and threats. There are some solutions available to those attacks. However, implementing autonomic security solutions in the hardware at the physical layer is more robust and faster. Complex schemes are usually more costly and should be avoided. Lightweight approaches should be implemented in order to increase device lifetime and reduce complexity. Attacks in the physical layer are described as follows.

- **Tag Cloning:** RFID tags can easily be cloned by an adversary. This attack can be launched by attaining the required information by direct access to a device or using reverse engineering (W. Zhang & Qu, 2013). The literature (F. Muhammad et al.,

2015) presented a tag cloning attack where an RFID reader is unable to distinguish between genuine and compromised tags.

- **RF Jamming:** Radio Frequency (RF) Jamming causes the sharing of wireless bandwidth to be ineffectual for the underlying devices. There is a significant threat level from Jamming based attacks in IoT because of the feature of remote, unmonitored deployment of smart devices. It is a physical layer attack in which RFs are interrupted for interference and saturated noise signals. A DoS attack can result from RF signal Jamming of underlying channels. Proper monitoring of the cognitive spectrum may prevent it (W. Liu et al., 2013; Salameh et al., 2018).
- **Node Injection Attack:** This attack is a variation of the MitM attack. It is one of the most powerful attacks on the physical layer of IoT. The attacker injects or deploys an additional node between two or more IoT nodes in the network topology. The injected node takes part in communication and takes control of the traffic in the network (Deogirikar & Vidhate, 2017).
- **Tampering:** This attack violates confidentiality and accessibility. In this type of attack, the information of the end device is modified, added, or deleted by an attacker. The attacker physically captures and compromises an end node from the network. Thus, all information can be collected by the attacker. In addition, reprogramming, redeployment and recovery of data from the field can be carried out by such an attack. An attacker recovers the format and type of transmitted information, and then tampers and regenerates the same type of data (Andrea et al., 2015; Pathak et al., 2021). Therefore, the precision of data generated by the network becomes remarkably doubtful.
- **Physical Damage:** An attacker physically damages IoT nodes by removing or deactivating them. Hence, the service becomes unavailable (Deogirikar & Vidhate,

2017). As a result, the necessity of mitigation methods for such an attack is significant for IoT. Today, smart cities are packed with IoT elements such as sensors, cameras and smart lights that can easily be damaged or stolen by adversaries. The adversary tries to attack the interface of IoT nodes to shut down or physically damage them. A multitude of these attacks will cause the network to fail (Andrea et al., 2015).

- **Exhaustion Attack:** Jamming or previously mentioned DoS attacks may result in exhaustion attacks. Particularly, the battery-operated devices may suffer from energy exhaustion if an attacker continuously attacks the network (Ashraf & Habaebi, 2015). Repeated attempts of re-transmission may cause collisions in IoT MAC protocols, which leads to high-energy exhaustion. Exhaustion is considered a high-impact DoS attack and is linked to deactivating IoT devices in order to reduce the network size and permanently remove the nodes from the network.

2.8.2.4 Multi-layer/dimensional Attacks

The following attacks may take place in different layers based on their architectures and policies. These attacks are discussed below.

- **Cryptanalysis Attack:** The cryptanalyst or attacker, in this kind of attack, tries to access an encrypted message without owning the encryption key (Andrea et al., 2015). A Brute-force attack is one of the cryptanalysis attacks in which the attacker systematically tries and guesses every possible passphrase or password combination. The cryptanalyst eventually finds the correct one to gain access to the system. The Known-plaintext attack, Ciphertext-only attack and Chosen-plaintext attack are some of the other examples of cryptanalysis attacks (Andrea et al., 2015).
- **Side-Channel Information Attacks:** During the process of the encryption operation, the attacker obtains information and performs a reverse-engineering process to gather

the cryptographic credentials of an IoT device (Gnad et al., 2019; Sayakkara et al., 2019). This information can be gained from the encryption devices, not from plaintext or ciphertext during the encryption process. Timing attacks, power or fault analysis and electromagnetic attacks are some of the instances of such attacks. The adversary makes use of information leakages and recovers block cipher keys. The Side-Channel attacks can be succeeded by directly defeating the intrusion prevention system using Boolean Masking (Dubey et al., 2020).

- **MitM attacks:** The adversary sits between two IoT devices to monitor, control, and get access to private information and interfere in communication between the two IoT nodes (Agyemang et al., 2019). MitM attacks are the kind of attacks which can be devastating to all the IoT layers. In this case, the cryptanalyst tries to sit between two nodes to gain access to the ciphertext and break the encryption system to find the encryption key. The cryptanalyst then obtains access to the plaintext and possibly alters the message of those two parties without their consent.
- **DoS/DDoS:** Denial of Service (DoS) and Distributed DoS attacks may shut down any IoT device, network or application and make service inaccessible to its users. These attacks may occur in many forms. One way to attack is by generating huge network traffic and broadcasting a tremendous request to the victim. The main purpose of this attack is to make devices, software, network services, and resources unavailable to the target consumers (F. Muhammad et al., 2015; W. Zhang & Qu, 2013). Additionally, the adversary may leak users' sensitive information. The DDoS attack is more dangerous than that of DoS attack, which combines several attacking platforms to invade one or more systems. The impact of DoS attacks on IoT gateway has been assessed in (Y. Lee et al., 2017). The authors developed a prototype using wired and wireless interfaces to analyze the DoS attacks.

2.9 Current Security Solutions (Countermeasures) for Security Attacks of IoT

Using the conventional and existing security approaches directly in the resource-constrained IoT devices is not straightforward. In short, the security approaches, models and architectures of the conventional network are designed based on the users' perspective, which may not always be suitable for M2M communication. Each IoT layer comprises a set of security protocols, techniques, algorithms, and security kits employed to make it harder for an adversary to attack or hack into the system. A better understanding of these notions will enable the researchers to analyze the security breaches and the level of defense needed. In addition, IDS, Intrusion Prevention Systems (IPS), and other complete security solutions can be applied to protect IoT from security threats.

The security threats, attacks and vulnerabilities may be similar for both conventional Internet and IoT, but the solution techniques and approaches are different for each network (Kortesniemi et al., 2019). This section brings the existing countermeasures, including learning-based, encryption-based, autonomic, and other methods, together to secure IoT systems from the application, network and physical layers. We present learning-based, encryption-based and autonomic approaches and discuss their relevance for constrained IoT.

2.9.1 Autonomic Approaches

Security approaches should be dynamic and with minimal human intervention. Although different security attacks/issues may require different security solutions, however, some researchers proposed self-secure/autonomic approaches. The term 'autonomic' refers to 'self-sufficient' or 'self-healing', and 'self-protection' mechanism, which manages the resources of the security system without user intervention (Ashraf & Habaebi, 2015; Nasiri et al., 2019). Self-healing solutions use specific countermeasures after an attack has been detected, and self-protection is used to prevent the attacks before they happen

(J. Zhang et al., 2018). Self-protection refers to a system which is capable of identifying and protecting itself from random attacks. The combination of a self-healing and self-protecting mechanism is called a hybrid approach. This section presents and analyzes the possible solution approaches that are classified based on different IoT architectures. Different intrusion mitigation and detection approaches follow autonomous methods for securing IoT.

An autonomic manager module is used in self-sufficient mechanisms, which manages resource elements using a structural arrangement called MAPE (monitoring, analysis, planning, and execution) control loop (Ashraf & Habaebi, 2015; J. Zhang et al., 2018). Autonomic approaches are the most popular techniques for mitigating IoT attacks. The basic working principle of MAPE architecture in an autonomic approach is as follows. Sensors collect information from the external environment. This symbolizes the monitoring segment of the MAPE architecture. This information is matched with recognized patterns and acute values for certain parameters in the analyzing module. This helps the self-regulatory body to analyze the operational state of the system in order to predict future behavior (J. Zhang et al., 2018). The planning module is responsible for further planning system goals and objectives on the basis of system constraints. Finally, the executing module implements the plan (Jahan et al., 2020; J. Zhang et al., 2018). In the autonomic approach, authentication and device identities are properly checked for self-protection. Table 2.3 exhibits the up-to-date autonomic countermeasure approaches and summarises each approach's objectives, advantages and limitations.

Table 2.3: Countermeasures on autonomic approaches

Ref.&Year	Objective	Description	Advantage	Limitation
(P. Kaur & Sharma, 2015)	Detects Spyware	Uses three parameters namely, description mapping, interface analysis and source code analysis	Determines the malicious behavior	Code obfuscation may affect the detection accuracy
(Alnabulsi et al., 2018)	Defends against Code Injection attack	Used gathering multiple signatures approach (GMSA)	Showed 99.45% accuracy	High computational cost
(Wolinsky et al., 2013)	Resists Intersection attack	A systematic design called Buddies in practical anonymity systems	Can choose appropriate mitigation policy for each pseudonym	If buddy is offline, user is unable to transmit data
(Koh et al., 2013)	Mitigates Hello Flood attack.	A puzzle scheme used for authentication which can be included in autonomic solution	Autonomic solution	Exhaust resource due to recursive operation to solve the puzzle
(Le et al., 2016)	Mitigates Sinkhole attack.	A semi-auto profiling RPL specification-based IDS	Successfully mitigate attack when IDS agent is functional	May not work if IDS agent is shutdown
(Abbas et al., 2012)	Detects Sybil attacks	A lightweight detection scheme resides in the lower layer and supports varied transmit powers	Lightweight	A powerful attacker may bypass the received signal strength indicator values
(Bu et al., 2015)	Detects Tag Cloning attack	Deterministic detection used three protocols namely BASE, declone, and declone+.	Can detect Tag Cloning attack for large anonymous RFID systems	Demands both genuine and clone tags to be presented in a specific location at same time
(Elngar & Bhatt, 2018)	Tamper detection	A tamper detection (TD) mechanism for IoT real data for healthcare applications	Great deal with security violations	May not be efficient for other IoT application

2.9.1.1 Countermeasures to Application Layer Attacks

In the MAPE architecture, viruses or malware patterns can be classified by analyzing them, and then they can be mitigated by executing the mitigation service(s)(Canzanese et al., 2013; J. Zhang et al., 2018). A constant vulnerability scan is one of the mitigation solutions which applies risk mitigation services and malware pattern classification. The authors in (Sharmeen et al., 2018) studied industrial mobile-IoT malware detection techniques and analyzed them in terms of static, dynamic, and hybrid approaches. A hybrid approach is proposed in (G. Kaur et al., 2015) for detecting Spyware using and comparing various antivirus software. This technique is based on three parameters: description mapping, interface analysis and source code analysis. These parameters determine the malicious behavior of an application.

An autonomic solution is necessary to mitigate Spoofing attacks. A detection algorithm called Enhanced Location Spoofing detection using Audibility (ELSA) was developed for IoT (Koh et al., 2016). The implementation of the proposed algorithm can be at the existing IoT backend server. The authors (Alnabulsi et al., 2018) proposed a tool called the Gathering Multiple Signatures Approach (GMSA) to defend against code injection attacks and showed an accuracy of 99.45% for the proposed algorithm. Interested readers also can refer to the framework proposed by (T. K. George et al., 2018)and (T. K. George et al., 2019) to detect such attacks. A model called differential-linear cryptanalysis has been presented in (Rao & Premchand, 2018) to evaluate a combined Cryptanalysis attack. The evaluation was performed on a complex cryptographic security system.

The authors (Stiawan et al., 2019) investigated the patterns of various Brute Force attacks to help IoT researchers and administrators to further analyze the attack type. They utilized a time-sensitive statistical relationship approach to identify the pattern and its configuration. The various forms of Forging attacks and their design and implementation

were presented in (J. Grover et al., 2013). They proposed an infrastructure supported detection approach for detecting Forging attacks in vehicular networks. Intersection attacks can be mitigated by using a self-protecting approach. K-anonymity technique was proposed by (Sweeney, 2002) to mitigate intersection attacks. The authors (Wolinsky et al., 2013) proposed a systematic design for resisting an intersection attack called Buddies in practical anonymity systems. In this design, users are able to choose appropriate mitigation policies for each pseudonym.

2.9.1.2 Countermeasures to Network Layer Attacks

The existing security solutions may not be suitable for IoT. The integration of autonomic approaches may protect the IoT network efficiently. The authors (Mehmood et al., 2018) proposed Naïve Bayes classification-based IDS by using multi-agents to detect misbehaving traffic of the network nodes to detect DoS attacks. Flooding attacks can be mitigated using an automatic self-protection mechanism by establishing connection barriers (Ashraf & Habaebi, 2015). One way to mitigate the Hello Flood attack is by means of a parameter, namely the link-layer metric, while selecting a default route (Wallgren et al., 2013). The authors in (Yi et al., 2006) proposed a solution to recover exhausted bandwidth automatically to save resources and defend against Flooding attacks. However, due to continuous broadcasts of route requests by an intruder, the interference may not be prevented by this solution. The authors in (V. P. Singh et al., 2010) presented a fundamental solution for countermeasure, which is an acknowledgement-based system. However, acknowledgement-based solutions require huge energy resources, which IoT devices are not capable of supplying. A puzzle scheme in (Koh et al., 2013) was proposed to mitigate this attack. This scheme and the use of the authentication mechanism can be included in the autonomic solution to mitigate Hello Flood attacks.

An IDS scheme is known as a compression header analyzer intrusion detection system

(CHA-IDS) is proposed in (Napiah et al., 2018), which analyzes compression header information. This scheme is capable of eradicating both individual and combined routing attacks in 6LoWPAN. Several countermeasures exist to mitigate Replay attacks, such as TDMA-based approach (Campagnaro et al., 2020; Ghosal et al., 2012; Marigowda et al., 2018). However, TDMA-based countermeasures are vulnerable due to several attempts of re-transmission where the authorized node's time slot is consumed, and the packet gets lost. Other countermeasures are presented in (Manzo et al., 2005), where two separate methods are explained for both single and multi-hop routing. Data encryption is also an effective method against Replay attacks. The authors in (Mahalle et al., 2014) proposed a group authentication called TCGA approach for IoT, which changes the session key dynamically to confront the Replay attack.

Many self-healing approaches have been proposed for sinkhole attacks. A semi-auto profiling RPL specification-based IDS was proposed by (Le et al., 2016) to protect from sinkhole attacks. However, this system may fail to detect Sinkhole attacks due to the centralization approach. This detection mechanism becomes non-functional if the IDS agent shuts down because of such attacks or low power. The authors in (B. G. Choi et al., 2009) proposed an IDS-based routing protocol using Link Quality Indication (LQI) and managed to detect the Sinkhole attacks for the network layer. However, once the detection occurs, such an autonomic system urges to take reactive action. Another IDS-based detection of Sinkhole attacks on 6LoWPAN for IoT called INTI has been presented in (Cervantes et al., 2015). This scheme analyzes the behavior of IoT devices by associating reputation, watchdog and trust policies for detecting the adversary. A model (Al Hayajneh et al., 2020) is proposed to mitigate MitM attacks for software-defined IoT networks. The authors made use of a traffic separation mechanism using deep packet inspection. They implemented the proposed model in Raspberry Pi. Combined with an intrusion detection

technique, a hybrid routing protocol is designed and proposed in order to prevent MitM attacks (J. J. Kang et al., 2019). The authors utilized a trusted third party to best deal with the performance difference of the protocol across various networks.

Sybil attacks are better mitigated using hybrid approaches. A Local Sybil Resistance (LSR) scheme has been presented in (X. Lin, 2013). It studied the accessibility of a Roadside Unit (RSU) to detect and stand against Sybil attacks in vehicular networks. The authors in (Zhou et al., 2011) aim to detect Sybil attacks on vehicular networks through workload and passive overhearing by preserving privacy and minimal network delay and overhead. A lightweight detection scheme mentioned in (Abbas et al., 2012), is able to identify new Sybil attacks without any centralized third party or any additional hardware. However, this scheme might not work well in all circumstances as the measures depend on the Received Signal Strength Indicator (RSSI) values; however, a powerful attacker may bypass the scheme. A comprehensive study of the behavior of a Sybil attack has been presented in (Mishra et al., 2018), which may help to formulate an effective countermeasure to defend IoT from such attacks. Authors defined the defense mechanisms as Behavior Classification-Based Sybil Detection (BCSD), Mobile Sybil Detection (MSD), and Social Graph-Based Sybil Detection (SGSD) (Y. Zhang & Pengfei, 2014) to defend against such attacks in IoT.

Another network layer attack is called the Clone ID attack, that can be prevented by using the instances' tracking number of each node. A lightweight and efficient mobile agent-based detection algorithm against the Clone ID attack is presented by (Sathish & Kumar, 2013). A scheme presented in (Shila & Anjali, 2008) consists of detection and localization phases to detect Selective Forwarding (SF) attacks. In this scheme, a packet counter is used to monitor a sequence of control messages from the wireless link. A method called SVELTE was proposed for mitigating SF attacks in 6LoWPAN-based IoT

(Raza et al., 2013). The authors designed and implemented their IDS in the Contiki operating system and evaluated using Cooja simulator. Game theory-based detection model presented in (Khanam et al., 2012; Pathan et al., 2013) to model and detect SF attacks for Wireless Mesh Networks (WMNs) efficiently. The authors in (Pandarinath, 2011) proposed a solution which allows breaking the data packets into a number of smaller pieces. Those smaller packets transmitted through specific routes detect the presence of an attacker. Some autonomic solutions such as message-based detection, redundancy and probing can be used to protect IoT from SF attacks.

In (Chugh et al., 2012), the Blackhole attack was studied and tested on the 6LoWPAN network. The simulation was done in ContikiRPL, using the Cooja simulator (Kugler et al., 2013). IDS and autonomic solutions for detecting, preventing, and confronting such attacks still require further research. An efficient sensor scheduling technique for protecting wireless transmission against eavesdropping attacks for the smart industry has been reported in (Zou & Wang, 2015). In this scheme, a node with the capacity of the highest secrecy is scheduled in order to transmit data to its sink node.

2.9.1.3 Countermeasures to Physical Layer Attacks

The mitigation approaches to Jamming attacks usually fall under the self-healing paradigm. The system executes a suitable mitigation method when a possible Jamming attack is assumed. Inside the Jammer area, the hearing range of the wireless devices is analyzed using the technique proposed by (Z. Liu et al., 2010). The cancellation and the usage of different parts of the spectrum are introduced for neutralizing the Jammer signals, whereas some attempt to estimate the position of the Jammer for further action (T. Kang et al., 2013; Shoreh et al., 2014). Some of them have utilized autonomic computing (Ashraf et al., 2016; Cai et al., 2013; Hussein et al., 2017) to detect Jammer's location. Node Injection is another vital attack in the physical layer. Monitoring and verification of device

identity may prevent Node Injection attacks. A unified security solution that integrates both self-protecting and self-healing methods are required to detect and mitigate this attack appropriately (Nasiri et al., 2019). The authors in (Bu et al., 2015) proposed a deterministic detection and presented three protocols, namely BASE, DeClone, and DeClone+, in order to detect Tag Cloning attacks in large anonymous RFID systems.

Tampering attacks can be mitigated by implementing the MAPE framework. For instance, nodes generate data packets which are monitored by the MAPE system periodically to see whether the node has been compromised or not. Suspicious data generation can be mitigated based on this data control method. For example, the system may remotely control the node for deleting data in it (i.e., security patch). A Tamper Detection (TD) mechanism has been proposed for IoT healthcare applications to deal with security violations (Elngar, 2018).

Physical damage to IoT nodes is considered a high-risk attack and cannot effectively be protected using software methods. The software method may disable the remote kill command, but physical damage of the device cannot be stopped (Ashraf & Habaebi, 2015). The only way to protect the smart devices is to ensure physical security by surrounding them with a protective case. The IoT devices should be monitored physically as these attacks are more physical. Exhaustion in end nodes can be prevented and mitigated through the use of timers, rate limitation and cross-layer designing cognitive adaptation (D. Feng et al., 2012). The autonomic system decides on duty cycling and cognitive adaption, which protects the availability and prolongs network lifetime (Ashraf & Habaebi, 2015).

2.9.2 Encryption-based Countermeasures

In this section, we discuss various existing symmetric and asymmetric cryptographic countermeasures for securing IoT. Cryptography is the representation of standard mathematical methods to defend against cyber-security attacks against confidentiality, entity

authentication, integrity and authentication (Jeba et al., 2011). The network of things is composed of several constrained nodes that communicate with each other using IPv6-6BR. Figure 2.6 depicts the basic symmetric and asymmetric encryption mechanisms.

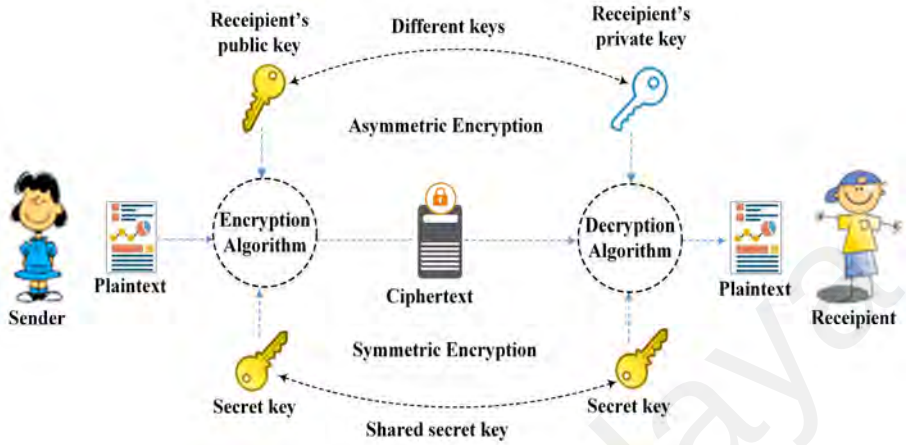


Figure 2.6: Symmetric and asymmetric encryption mechanisms.

Table 2.4 summarizes some up-to-date encryption-based countermeasures for IoT. The table analyzes and presents the techniques used in the schemes, their objectives, advantages, limitations, and applied area. The following variations of encryption-based countermeasures are applicable to different attacks of IoT architecture.

2.9.2.1 Countermeasures using Symmetric Key Cryptography

Symmetric Key Cryptography is also known as secret-key encryption, where the sender and receiver share a single key for both encryption and decryption. Some of them are Advanced Encryption Standard (AES), Data Encryption Standard (DES), 3DES, International Data Encryption Algorithm (IDEA), Tiny Encryption Algorithm (TEA), Twofish, RC6 and Blowfish (Chandra et al., 2014; Chaudhari & Patel, 2014; Eisenbarth et al., 2007).

Various symmetric encryption distributions are available for IoT, like Probabilistic Key Distribution where a shared symmetric key or bytes are selected randomly from a secured key pool and flashed at a constrained IoT device (Abualghanam et al., 2019). For instance,

the authors (K. Muhammad et al., 2018a) proposed a fast probabilistic, lightweight and robust, encryption algorithm suitable for IoT systems. The scheme encrypts the visual contents using image encryption prior to transmission. The algorithm is capable of producing a number of ciphered images with limited processing and memory requirements and ensures a high level of security. In Deterministic Key Distribution, a key pool is created, and the keys are distributed homogeneously in such a way that a common key is utilized for every two nodes to guarantee secure connectivity (Banupriya et al., 2021). An intelligent deterministic approach is proposed for secure device-to-device (D2D) communication for delay-sensitive IoT user equipment (Nauman et al., 2019).

Next, other types of key distributions are also available for IoT. For instance, in Offline Key Distribution, where either each node shares one key in the same network or two nodes share a network key pairwise depending on the utilized protocol. This scheme is also known as an offline key distribution (Meshram et al., 2019). There are several existing approaches based on the offline key distributed mechanism available that may be applicable in the context of IoT. Few such schemes, namely SPIN, BROSK and SNAKE (Lai et al., 2002) generate session keys without the necessity of a key server. A master secret key is shared among all nodes in the same network in these schemes. In the SNAKE scheme, two random nonces are hashed to obtain the secret key. The communicating nodes generate random nonce using a pre-shared key. In BROSK approach, the session key is constructed from a broadcasted nonce in the network.

Furthermore, a standard IPsec is implemented into IP-based WSNs using 6LoWPAN in (Raza et al., 2011). In this work, the authors proposed a header compression mechanism to support both the Authentication Header (AH) and Encapsulation Security Payload (ESP) header. However, one drawback of offline key distribution schemes is, that they do not support the re-keying services. The Protocol for Carrying for Network Access (PANA) has

been proposed as a key distribution solution for IoT based on an external assisted server (Forsberg et al., 2008). Pre-shared key distribution is one of the authentication methods supported by Extensible Authentication Protocol (EAP) and PANA, and it uses EAP and runs over UDP. An improved version of PANA is proposed by (Kanda & Chasko, 2012), which can be adopted by resource-constrained IoT. In this work, the authors have removed unnecessary PANA header fields and minimized the number of cryptographic primitives. However, it may reduce the code size for implementation, but it might not be suitable for IoT in terms of response time or energy consumption.

Another type of symmetric encryption is known as Server-Based Key Distribution (K. T. Nguyen et al., 2015). In such schemes, two or more nodes and one or more trusted and powerful servers engage in message exchanges. The server acts as a Key Distribution Centre (KDC). Many sessions can be created during the communication process and each session can be secured through the forward secrecy technique (Q. Feng et al., 2018). Forward secrecy is an encryption technique for safeguarding communications conducted over the Internet. This method prevents an adversary from accessing past data from a set of transmission sessions. In forward secrecy, the key use in one session has no relation to the key use for another session.

A lightweight encryption algorithm has been proposed for IoT by (Baskar et al., 2016), which uses a chaos map-based key applied in the Field-Programmable Gate Array (FPGA). The scheme uses 1550 logic gates and 128 bits of key size and achieves 200 kbps of maximum throughput. In (Du et al., 2006), a scheme which depends on the deployment knowledge is provided. This scheme gets rid of excessive key assignments. A mitigation technique (Gnad et al., 2019) was proposed for side-channel attacks called leaky noise. The authors carried out a leakage assessment and characterized noise using statistical methods for IoT devices. They provided key recovery using the AES method. However,

the method is not robust in terms of mitigating the attacks. Authors in (Ito et al., 2007) proposed a solution in which they mapped the keys on two-dimensional states. The authors added a probability density function to it in order to offer better key connectivity.

The authors in (Hussen et al., 2013) proposed a Secure Authentication and Key Establishment Scheme (SAKES) for IP-based M2M communication between an external internet host and a sensor node. In this scheme, an authentication module is deployed in unconstrained 6LBR to create trust relationships between the IoT nodes, 6LBR and remote Internet server. Diffie-Hellman (DH) (Rescorla, 1999) key agreement is then applied with the distant server, and the session key (SK) is calculated for the IoT device. Finally, using the SK, which the sensor node received from PBS, it can communicate securely with the server placed remotely.

Table 2.4: State-of-the-art lightweight encryption schemes

Ref. & Year	Base technique(s)	Objective(s)	Advantage(s)	Limitation(s)	Application
(Aziz & Singh, 2019)	RSA, ECC, CS	To secure the network from CPA attack	Good at signal compression and lowering computational cost	Weakly encrypted	Communication between BS and IoT nodes
(Habib et al., 2018)	NtruEncrypt, Rabin scheme, RSA, AES	To provide low-latency communication and achieve data security	Key generation time, data encryption–decryption time and latency are low	Did not consider energy and memory utilization	IoT
(K. Muhammad et al., 2018b)	Nonlinear chaotic map, cipher	To reduce the bandwidth consumption and transmission cost	Low processing time and high level of security	Applies only for video traffic	Industrial IoT
(Usman et al., 2017)	AES, PRESENT, DES	To provide security and resource utilization	Energy and memory efficient	It uses only an XORed key	Secure IoT
(Y. Yang, Zheng, & Tang, 2017)	Decisional Bilinear DH (DBD), HER	To reduce computation and communication overhead, and to provide security.	Better data encryption	Access right verification and session key management are not available in the scheme	Smart healthcare
(Al Salami et al., 2016)	PKI, IBE, DH	To reduce computation time	Reduced overhead cost	The focus is mainly on confidentiality	Smart home
(Baskar et al., 2016)	Blowfish XTEA	To maximize resource utilization and provide security	Achieved of 200 kbps of throughput	Occupancy of memory size is key issue	WSNs of IoT
(Yao et al., 2015)	ECC, DH, ECDDH, ABE	To address the security and privacy issues in IoT and to reduce overhead	Efficient for broadcasting encryption	Not scalable and flexible for IoT applications	Single authority applications
(Sahraoui & Bilami, 2015)	DTLS, HIP-DEX, DH, HIP-BEX, CD-HIP,	To achieve lightweight E2E security and to reduce energy consumption	Reduced communication overhead, energy and memory consumption	Incompatible	WSNs of IoT

2.9.2.2 Countermeasures using Asymmetric Key Cryptography

Asymmetric Key Cryptography (AKC) is a well-known approach to forming an efficient and secure communication among nodes and is also known as Public-Key Cryptography (PKC) (Du et al., 2005). In the AKC, the sender encrypts a message using the recipient's public key. The receiver decrypts the message by using his private key. Various asymmetric algorithms have been developed and implemented in IoT (Shah et al., 2020), such as Rivest–Shamir–Adleman (RSA), DH, Elliptic-Curve Cryptography (ECC), and Pretty Good Privacy (PGP) (N. Ahmed & Khan, 2021; AlMajed & AlMogren, 2020; R. Cheng et al., 2021; Kavin & Ganapathy, 2020; M. A. Khan et al., 2020; C.-K. Wu, 2021). AKC is also used to create Message Digest-5 (MD5) and Digital Signature Algorithms (DSA) (Chandra et al., 2014; Karim et al., 2021; S. Xiao et al., 2021). The major drawbacks of AKCs application for IoT are higher energy consumption and computation and operating costs. Regardless of those drawbacks, researchers still pursue applying AKCs in the IoT environment (K. T. Nguyen et al., 2015). It is because AKCs are a very powerful tool to secure communication over the Internet.

Furthermore, in AKC, if a public key or private key is used to encrypt a message, the same algorithm and the matching private key or public key can only be able to decrypt that message (Bala & Kumar, 2015). There are many variations of AKC algorithms. Key Transport Based Scheme is similar to the conventional key transport scheme that emphasizes the secure transmission of information using the public key. In order to establish safe and secure communication between two nodes in IoT, a Certificate-Based Encryption algorithm is the best choice. Each node in IoT maintains a certificate signed by a well-known and trusted third party (i.e., a CA). In fact, the CA guarantees the trustable relationship between the nodes (K. T. Nguyen et al., 2015).

Identity-Based Encryption (IBE) allows an arbitrary string to be the public key such as

a receiver's email address. In IBE, a Public Key Generator (PKG) generates the private key from its public key of each node. Attribute-Based Encryption (ABE) (Oualha & Nguyen, 2016) has changed the traditional concept of public-key cryptography relatively recently. It is the extension of the IBE scheme. Key Agreement Based Scheme is another technique based on asymmetric primitives and key agreement protocols by sharing the secret key among two or more parties in IoT.

Likewise, NtruEncrypt (Gaubatz et al., 2005) and Rabin's approach (Rabin, 1979) are examples of Raw Public Key (RPK) encryption methods that have been proposed for WSNs. Rabin's approach is similar to the conventional RSA algorithm. This scheme consumes the same energy for decrypting messages like that of the RSA algorithm with the same level of security. As one squaring is needed for encrypting a message, this encryption scheme is much faster. Lattice-based cryptosystems, namely NtruEncrypt algorithms are proposed for IoT (Hoffstein et al., 2009; Seyhan et al., 2021). The schemes is suitable and efficient for highly resource-limited things such as RFID tags and smartcards. With the inspiration from (Boneh & Franklin, 2001), the authors (L. Yang et al., 2013)proposed the IBAKA approach using pairing-based cryptography, which is mainly a combination of the IBE-ECDH scheme. However, in order to establish a session key, the IBE scheme is tailored into an Elliptic Curve Diffie-Hellman exchange (ECDH) (De Meulenaer et al., 2008) key exchange.

Lightweight encryption for smart home, namely LES (Al Salami et al., 2016), was proposed for home applications and the scheme consists of two sub-algorithms, called "KEYEncrypt" for session key encryption and "DATAEncrypt" for encrypting data. The scheme achieves confidentiality, adaptability and reduces overhead costs. Other type of AKC approach is known as Attribute-Based Encryption (ABE) (Ali et al., 2020; La Manna et al., 2021; J. Li et al., 2020; L. Li et al., 2020). The feasibility of implementing ABE in

IoT is still under investigation. A CP-ABE based lightweight ABE security approach is proposed in (Oualha & Nguyen, 2016). Again, a lightweight with a no-pairing method using the ECC scheme for IoT has been presented in (Yao et al., 2015). This is an efficient scheme for broadcasting encryption and access control based on the ciphertext.

Lately, another lightweight scheme was proposed in (Y. Yang, Zheng, & Tang, 2017), which aids distributed access control of Protected Health Information (PHI) among different healthcare applications by providing an efficient keyword search. Major heavy calculations are performed by a semi-trusted computation center in the data encryption phase. The security of this scheme is based on Elliptic Curve Decisional Diffie–Hellman (ECDDH) technique. An efficient Host Identity Protocol (HIP) based lightweight encryption has been proposed to ensure end-to-end security for IoT (Sahraoui & Bilami, 2015). It is a 6LoWPAN header compression of HIP packets. This scheme significantly reduces communication overhead, energy, and memory consumption.

2.9.2.3 Countermeasures using Hybrid Key Cryptography

Symmetric and asymmetric ciphers are combined to form a cryptographic technique called Hybrid Key Cryptography (HKC). Hybrid schemes utilize the benefits of the strengths of both approaches (Yehia et al., 2015). A great number of researches have shown that the combination of symmetric and asymmetric cryptography utilizes the strengths of both schemes and makes it suitable for IoT networks (Kavitha & Caroline, 2015; Mushtaq et al., 2017; Xin, 2015; Y. Zhang & Pengfei, 2014). However, more research works are still needed to improve hybrid security schemes to be a more lightweight and a stronger solution at the same time. Existing hybrid schemes are advantageous for large hierarchical networks, which can utilize the benefits of both public and secret key schemes.

There are numerous versions of hybrid cryptography available for resource-limited devices and networks. An Efficient and Hybrid Key Management (EHKM) (Y. Zhang &

Pengfei, 2014) is a hybrid scheme which is mainly designed for heterogeneous WSNs. The lightweight public key encryption method, ECC is placed at cluster heads and BSs, while adjacent nodes in the same cluster use a one-way hash function based symmetric encryption method. A hybrid lightweight encryption algorithm for IoT called LEA-IoT has been proposed in (Habib et al., 2018). This hybrid algorithm utilizes asymmetric encryption based on a linear block cipher and symmetric encryption based on a conventional private key and achieves data security. Key generation time and data encryption-decryption time were calculated as the lowest. This scheme achieved low-latency communication.

Secure IoT (SIT) utilizes symmetric key encryption of 64-bit block cipher with 64-bits key size and had five rounds. It is a lightweight hybrid solution based on Feistel and Substitution-Permutation (SP) networks (Usman et al., 2017). Some researchers proposed the Compressive Sensing (CS) technique to provide signal compression to make the scheme lightweight and encryption simultaneously. For instance, a Lightweight Secure Scheme (LSS) is proposed by (Aziz & Singh, 2019) to secure IoT networks from Chosen Plaintext Attack (CPA) and to prolong the network lifetime. LSS consists of three stages; key generation stage where BS and IoT nodes generate random numbers, key exchange stage where BS and nodes exchange the number in a secure way, and compression/encryption stage to generate secret compressed samples in order to mitigate CPA.

2.9.3 Learning-Based Countermeasures

Learning-based algorithms are based on Artificial Intelligence (AI) that allow systems to predict the future events without requiring explicit programming. The historical data of a particular application is used as input for predicting future behavior of that application. Learning-based algorithms are capable of solving real-time problems that lead to maximize the efficiency of a system (Adnan et al., 2021; Lansky et al., 2021). Learning-based approaches have been extensively used in almost all areas, including intrusion detection

because of their distinctive nature of resolving real-time problems. ML/DL methods mainly learn from existing data and predict the future behavior of a system. The DL approach is a specific type of ML algorithm. It can improve system performance by classifying normal or abnormal behavior of a system. ML algorithms require explicit feature engineering processes to prepare the dataset for training the model. In contrast, DL approaches are capable of extracting features during model training. The performance of such learning-based models could be evaluated in terms of classification accuracy. There are three categories of a learning algorithm in practice, such as supervised, semi-supervised, and unsupervised learning (E. Gyamfi & Jurcut, 2022). Supervised learning approaches use labelled input data to train the learning models. In contrast, unsupervised learning approaches learn from an unlabelled dataset and extracts useful information. The semi-supervised learning approach is a combination of supervised and unsupervised methods which make use of both labeled data and unlabeled data (E. Gyamfi & Jurcut, 2022; Lansky et al., 2021).

Since learning-based approaches learn from historical data, hence, datasets are essential part while designing an IDS model. Datasets contain network traffic of normal and abnormal observations of a particular application. The innovative learning-based detective and predictive algorithms make a well-designed dataset an inevitable part of learning models for IoT applications and networks. These datasets are commonly the IoT network packets retrieved from network flows, logs and sessions. Collecting and preparing well defined datasets for IoT intrusion detection can be intricate, time consuming and costly (E. Gyamfi & Jurcut, 2022; Lansky et al., 2021).

The intrusion detection of leaning-based models is usually in two phases, the training and detection phase. In the training phase, the model learns the distribution of features, while in the detection phase, the abnormalities are detected using the learned features

(Dua et al., 2019; Janarthanan & Zargari, 2017; Soe et al., 2019). Features are important information for classification and can be extracted from raw data. The features in intrusion dataset determine the effectiveness of a learning-based IDS, and they contain dependents and independent variables. The dependent variable is the class labels, which are the attack categories in the intrusion dataset. Normally, the class labels in network traffic are imbalanced, which also plays a vital role in the performance of learning-based algorithms.

Sometimes the data contain insignificant and redundant features. Hence, it is necessary to select important features and reduce the feature set. This could lead to improvement of intrusion detection performance of learning-based IDS and result in making the IDS to be of lightweight to fit for resource-constrained devices. The major benefits of feature reduction have been stated in (Manikandan & Abirami, 2018) as follows. A feature reduction method:

1. Meets the storage requirements of resource-constrained devices
2. Increases the speed of the learning algorithm.
3. Gets rid of noisy and redundant features.
4. Speeds up data analysis.
5. Improves data quality.
6. Increases model performance.
7. Optimizes resources such as memory and energy during detecting intrusion.

The three most used feature reduction/selection techniques are Genetic Algorithm (GA), Principle Component Analysis (PCA) and Information Gain (IG). Other methods include embedded, wrapper and filter approach. Among them, the filter method better optimizes the selection process and is more suitable for a high-dimensional dataset.

Meanwhile, various learning-based techniques are available for detecting intrusions in

IoT, such as Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), Artificial Neural Networks (ANN), Deep Neural Network (DNN), Recurrent Neural Network (RNN), Deep Eigenspace Learning (DEL), Deep Belief Network (DBN), Auto-Encoder (AE), Principle Component Analysis (PCA) and Convolutional Neural Networks (CNN) (Ahmad et al., 2021; Alsoufi et al., 2021; Thakkar & Lohiya, 2021) etc. A lot of studies are conducted on ML and DL for IoT security. Interested readers can refer to the literature (Aldweesh et al., 2020) for working principles and applicability of various ML/DL methods in IoT security. In this section, we are focusing on analyzing and reporting some advanced countermeasures approaches for IoT security based on learning algorithms. Table 2.5 presents some state-of-the-art learning-based security countermeasures for IoT in different layers. The table summarizes the objectives, advantages, performance accuracy, dataset used, and limitations of each learning-based security measure.

Table 2.5: State-of-the-art lightweight learning-based IDS

Ref, & Year	Learning method	Dataset	Objective	Advantage	Performance accuracy	Limitation
(Thamilarasu & Chawla, 2019)	DBN-based DNN	Simulated using Cooja	Detect Sinkhole, DDoS, Blackhole, Wormhole attacks	Can detect real-world intrusions effectively	Precision rate of 95% and recall rate of 97%	Require high processing power and large dataset
(L. Xiao et al., 2016)	Q-learning and Dyna-Q	–	Detect Physical layer spoofing	Robust against environmental Changes	Average detection error rate is less than 5%	Detection may be limited for high-speed mobility devices
(Vinayakumar et al., 2019)	Scale-hybrid-IDS-alertnet	NSL-KDD, KYOTO, UNSW-NB15, CICIDS-2017	Comparative study of ML methods	Able to perform better in both HIDS and NIDS	Varies with various datasets, proposed DNN and ML methods	Accuracy can be improved
(Yavuz et al., 2018)	DNN	IRAD	Detects Version Number, Blackhole and Hello Flood	Very high training accuracy	Accuracy 99.5% and F1-Scores 99%	High training time, computation cost
(Bostani & Sheikhan, 2017)	Optimum-Path Forest ML	–	Detects SF, sinkhole and wormhole suspicious nodes	Light IDS agent that will eliminate the local analysis	Detection accuracy up to 96.02%	Detection accuracy can be improved
(Hasan et al., 2019)	LR, SVM, DT, RF and ANN	DS20S	Dos, Probing, Malicious Control, Scan, Spying	Comparative study is given	Accuracy up to 99.4%	No new algorithm is devised
(Pajouh et al., 2016)	Naive Bayes, K-Nearest Neighbor	NSL-KDD	Detects U2R and R2L attacks	Lower resource requirements	Detection accuracy up to 84.86%	Can detect only low frequency attacks
(Shone et al., 2018)	NDAE and RF	KDD Cup '99 and NSL-KDD	Detects DoS, Probe, U2R and R2L attacks	Reduced training time	F-score of 87.37%, recall of 85.42% and precision of 100.00%	High computational cost
(Fang et al., 2020)	Elman neural network and SVM	DARPA	Ensures safety of information systems	Focused to improve training time	Detection rate is 100% and false alarm rate is 2.8%.	Difficult to select the number of hidden layers
(Aminanto et al., 2017)	Stacked sparse AE, SVM, DT, and ANN	AWID	Detect Impersonation attacks	SVM shows better accuracy	Detection accuracy 99.918% and false alarm rate 0.012%	Long training time

2.9.3.1 Countermeasures to Application Layer Attacks:

A linear SVM algorithm is proposed (Ham et al., 2014) to detect malware in Android. They analyzed the detection accuracy of SVM with other machine-learning algorithms in terms of malware detection accuracy and showed that the proposed approach outperforms other algorithms. A novel distributed deep learning method was proposed to detect attacks in fog-to-things computing (Abeshu & Chilamkurti, 2018). The results prove that deep-learning models are better than shallow models in terms of detection accuracy, false alarm rate, and scalability. The authors (Fang et al., 2020) proposed a method with a combination of the Elman Neural Network and the SVM algorithm. They introduced Back Propagation Through Time (BPTT) algorithm to transform the processing of the network at various times into a forward network.

Besides, the authors (Aminanto et al., 2017) presented a three-layer architecture to detect impersonation attacks using the AWID dataset. First, the feature extraction is performed using stacked sparse AE, and feature selection is made using SVM, DT, and ANN algorithms. Finally, normal or abnormal traffic is classified using the ANN algorithm. The experimental results showed that the SVM had better detection accuracy; however, it took the longest training time.

2.9.3.2 Countermeasure to Network Layer Attacks

A Deep Belief Network (DBN) approach based on a Deep Neural Network (DNN) has been proposed by (Thamilarasu & Chawla, 2019) to detect network attacks. They created a dataset using the Cooja simulator, which is trained to detect sinkhole attacks, DDoS, Blackhole, and Wormhole attacks. Their deep-learning model utilized supervised training and binary classification for identifying abnormal activities. The proposed intrusion detection system can detect real-world intrusions effectively. They achieved an average precision rate of 95% and a recall rate of 97% for different attack scenarios.

Next, an Optimum-Path Forest (OPF) based on the ML method using graph theory has been proposed to detect suspicious nodes of SF, sinkhole, and wormhole attacks (Bostani & Sheikhan, 2017). In this method, specification and anomaly-based agents were assigned in the router and root nodes, respectively, to analyze the behavior of the host node and incoming data packets. They achieved a detection accuracy of 96.02%. In (Vinayakumar et al., 2019), a Scale-Hybrid-IDS-AlertNet based on the MLP-DNN model was compared with various existing datasets. The hybrid alert technique applied a highly scalable DL architecture to analyze the network and host-level activities. The proposed framework provides better accuracy than traditional machine learning classifiers.

Furthermore, a simple DL algorithm was deployed to train the IRAD dataset, which was created using Cooja simulator to detect Version Number, Blackhole, and Hello Flood attacks (Yavuz et al., 2018). After pre-processing, the datasets were labelled and mixed with attack and benign data. These datasets were then fed to a deep learning algorithm. The model achieved very high training accuracy of up to 99.5% and F1-scores up to 99%.

The authors (Aminanto et al., 2017) presented a three-layer architecture to detect impersonation attacks using the AWID dataset. First, the feature extraction is carried out using stacked sparse AE, and feature selection is made using SVM, DT, and ANN algorithm. Finally, the normal or abnormal traffic is classified using the ANN algorithm. The experiment results showed that the SVM had better accuracy; however, it took the longest training time. Next, the authors used the same dataset to benchmark their detection results with (N. Gao et al., 2014) and the hybrid method (Jiao et al., 2015). Their experiments provided a high accuracy of 97.9% compared with (Jiao et al., 2015)and (N. Gao et al., 2014), which had an accuracy rate of 93.94% and 92.1%, respectively.

2.9.3.3 Countermeasures to Physical Layer Attacks

Q-learning and Dyna-Q-based on Reinforcement Learning (RL) are applied to detect physical-layer spoofing (L. Xiao et al., 2016). Moreover, this method is based on interactions between a receiver and Spoofers as a zero-sum spoofing detection game. Simulation results show that the spoofing detection is robust against environmental changes. Again, the authors (Erpek et al., 2018; Shi et al., 2018) initiated a Jamming attack using a deep neural network and proposed mitigation methods for this type of attack. However, this protection system does not adapt the information of the Jammer and permits the transmitter to regulate its protection level on the fly based on its attained throughput.

Likewise, dynamic watermarking (Ferdowsi & Saad, 2018a, 2018b) is an algorithm that is capable of detecting and preventing cyber-physical attacks such as Code Injection and Eavesdropping. In addition, the method is based on the Long Short-Term Memory (LSTM) framework that allows IoT devices to extract a set of stochastic properties from their produced signals and dynamically watermarks these features into the signal. Interestingly, this algorithm enables the IoT gateway to authenticate the reliability of the signals effectively. However, authentication requires high computational resources.

A scheme based on channel-based machine learning was proposed (Chen et al., 2020) to detect both Clone and Sybil attacks. Simulations and experiments have been carried out in real environments. Certainly, both results confirm that the accuracy rate of authentication of the method achieves 84% without requiring manual labeling. The authors (Sayakkara et al., 2019) recently proposed a learning-based algorithm to detect side-channel attacks and showed 82% and 90% detection accuracy on high-end and low-end IoT devices, respectively.

2.10 Limitations Associated with Present Security Solutions

Undoubtedly, this chapter has addressed the key security issues, presented existing advanced countermeasures, and emphasized the areas that require further research. The following subsections provide an analytical discussion, suggest the appropriate security schemes for IoT, and propose future research directions for the researchers.

2.10.1 Existing Security Approaches

Several learning-based, autonomous, symmetric, asymmetric security schemes or mechanisms are mentioned above. However, not all of them are suitable for IoT. This section analyzes and discusses the advantages and trade-offs among existing countermeasures.

- **Autonomic Approaches:** The autonomic approaches have the advantages of an automatic architecture where different modules accomplish different tasks to detect and mitigate attacks. It is encouraged to design security solutions where human physical intervention requirement is low instead of relying on a complete autonomic solution.

  Moreover, integration between software and network virtualization helps to achieve the CIA triad with self-healing and self-protecting capacity in the IoT environment (Nasiri et al., 2019). Some autonomic systems demand complex cognitive structures to provide a self-repair mechanism. However, due to the resource limitation, it is encouraged to design a lightweight and energy-efficient autonomic system for the IoT. Furthermore, IoT devices transfer data to other devices or a central location; therefore, autonomic security solutions should be compatible with dynamic communication protocols and heterogeneous environments.

  Indeed, designing autonomic security without considering the complexity level is a roadblock to evaluating and implementing them in the IoT system. Existing

self-securing standards require a constant power supply to keep them operational. An intelligent power monitor and control system is necessary to keep the autonomic system running without energy exhaustion.

However, developing a fully automated security solution remains a shared vision among researchers. Contemporary researchers are still working towards designing a complete, portable, and robust self-securing system. Currently, a fully autonomic solution does not exist, and such an anticipated solution remains under continuous research consideration. Therefore, there is a need for more research in this vital field in order to develop a holistic, dynamic, and robust autonomic security solution for current and future IoT architecture.

Encryption Algorithms: Whenever asymmetric cryptography is used, the light-duty nodes will experience performance inefficiency. On the other hand, the heavy-duty nodes will lose the opportunity for better security implementation using symmetric cryptography. In order to resolve this dilemma, a security system should be able to adapt automatically to the cryptographic capabilities (Holzer & de Meer, 2011). Generating suitable small keys is challenging using public-key cryptography. The existing cryptosystems are designed to provide security for a specific security goal. However, achieving all the security goals at the time using conventional encryption-based countermeasure may not be possible. Therefore, research work has been initiated to discover quantum cryptography. Indeed, quantum cryptography is still in its infancy stage. Designing and developing such cryptosystems should consider the compatibility issue that might arise with the diverse IoT technology and protocols. The main criteria for evaluating IoT key management schemes include computational, communicational, energy and storage complexity; connectivity; scalability, and security resilience. These measures are usually used to validate the effectiveness of

security schemes. Communication capacity refers to the number and size of packets transmitted and received by IoT nodes. Connectivity refers to the probability of connection for a pair of nodes with the same pre-distributed key or sets up a key path among them. The applied key scheme must be scalable so that the network supports adding or removing IoT nodes anytime. Finally, resilience refers to the probability of an attacker compromising a link or whole network depending on the number of nodes captured by an attacker.

The above are important factors in evaluating the performance of the cryptographic schemes (J. Zhang & Varadharajan, 2010). Due to less computational complexity, the symmetric-key techniques are commonly used as they are appropriate for the resource-limited characteristics of the IoT networks. However, the efficient symmetric key cryptography shortages for IoT are also obvious. Moreover, there is still some weakness in the existing approaches, such as security resilience, connection probability, and scalability.

Learning-Based Countermeasures: The efficiency of learning-based approaches depends on attack detection accuracy, true and false-positive rates, F1-score, and some other performance metrics. These metrics tell how efficiently a model can detect an intrusion. True and false positive rates represent the rate of intrusions identified as true intrusions or normal traffics identified as intrusions by the model, respectively. F1-score is also a critical accuracy measure of a learning model on an intrusion dataset (E. Gyamfi & Jurcut, 2022; Lansky et al., 2021). The training time of the model also plays a vital role in the selection of the model. There are trade-offs among ML/DL-based algorithms. Deep learning algorithms can be trained on devices with relatively high processing and memory capabilities because they require large datasets, and the structure of neural networks are complex.

Conventional machine learning algorithms, on the other hand, can be trained on devices with somewhat lower processor and memory properties.

Notwithstanding, in terms of performance, the DL approaches provide higher accuracy and reliability compared to ML algorithms. Due to their superiority in terms of accuracy when trained with enormous amounts of data, DL algorithms are becoming increasingly popular. Deep learning can extract useful information from structured and unstructured data more efficiently than machine learning. Some learning algorithms are less computationally costly; some are complex in terms of their structures. For example, a Decision Tree algorithm can be constructed with only a few or several trees for either simple or complex classification. Naive Bayes classifiers are incapable of finding relationships among features to be learned from. Consequently, they classify the intrusions inaccurately. RNN algorithms suffer from vanishing gradients.

Meanwhile, some learning-based algorithms (e.g., CNN and SVM) are capable of breaking cryptographic implementations (Y. Yu et al., 2021). Further research is required to investigate these algorithms in terms of their purposes and performances. The structure of DL algorithms is more complex than that of ML algorithms and requires a larger dataset to be trained. DL methods' training time and computational complexity depend on how complex the structure is. Various tools and inbuilt libraries are available such as Keras, Tensorflow, and so on, to automate the training process. Ensemble-based and stack-based DL algorithms are computationally costly. The deployment of these methods may create bottlenecks during real-time implementation. Therefore, designing and developing a learning-based algorithm must be considered when adapting it to real implementation.

Additionally, learning-based methods depend on the existing data or information

from where the models learn and classify the incoming traffic as normal or abnormal. These datasets can be either smaller or larger. However, finding a real-world IoT-dedicated dataset to train learning-based algorithms is challenging. Machine learning algorithms require smaller datasets to train the model compared to deep learning algorithms. Finding publicly available intrusion detection datasets is another challenge, as very few datasets are available on public platforms.

Moreover, ML and DL algorithms may produce a higher false-positive rate if the dataset used in training is not realistic. Therefore, high-quality real-world and comprehensive IoT training datasets are required to train these methods. However, generating a high-quality training dataset remains challenging for contemporary scholars in IoT-related academic investigations.

2.10.2 Data Imbalanced Problem

Recent advancements in deep learning have helped to establish intrusion detection systems as an essential part of IoT applications. Despite the success of these learning models in solving intrusion detection in real-world applications, learning from an imbalanced dataset is still challenging. Most learning models suffer from low detection accuracy due to the highly skewed class distributions where there are only a few intrusion samples for specific classes in collected network traffic. Due to this, the rare instances, unknown and low-frequency attacks aren't easily detected, and hence, it becomes a challenging task for standard intrusion detection techniques to attain high accuracy in detecting the minority class attacks. Different approaches to tackling data imbalance problems can be grouped into three main categories: data-driven, algorithm-based and hybrid approaches. Among them, the data-driven models now offer promising performance. The application to conventional ML/DL algorithms fails to provide efficient performance. Indeed, when data samples are limited to a certain class, the intrusion detection models tend to be biased towards

the majority class. This results in increasing the probability of misclassification of the low-frequency attacks. In the context of IoT applications, this bias in classification models leads to the majority of unknown intrusions remaining unnoticed, and this significantly impacts the standard IoT services. The impact of not detecting the attacks is much more detrimental to the quality of service than misclassifying the attacks.

2.10.3 Lightweight Solutions

As mentioned in earlier sections, IoT devices are often deployed in remote areas and might be unattended, which may result in physical layer attacks in particular. The sensor-equipped connected things are often battery operated, embedded with small memory chips, and limited computation and communication capabilities. Therefore, there exists a roadblock in implementing complex and robust security protocols. Designing a lightweight solution with all security features is also a challenge. The communication may take place via popular wireless technologies, which are easier to compromise and vulnerable to interference and interception attacks. DoS attack may result in a single point of failure and severe service unavailability due to centralized communication. Finally, providing a complete self-securing and autonomic security architecture is necessary, but it is incredibly challenging to implement because of the IoT features like resource-limited characteristics.

2.11 Related Work on Intrusion Detection

In recent years, the use of the machine and deep learning algorithms to identify different types of network intrusion has become the prime research interest. Despite the promising outcomes that can be achieved by both ML and DL approaches, the DL algorithms outperform the shallow ML method due to their ability to automatic and high-level feature extraction. The DL algorithms are capable of performing complex intrusion detection tasks. Therefore, intrusion detection using deep learning techniques

has attained widespread research attention among researchers in recent days. Neither ML/DL methods nor cryptography is an individual algorithm, as they do not follow a single specified process to perform an intrusion analysis. Specifically, the learning-based approach is a family of techniques. ML/DL approaches learn from real network traffic and find the attack patterns using the learned model. The learning-based techniques are to be explored, evaluated, and compared in this research. Hence, this section provides some state-of-the-art related works based on learning algorithms and identifies the research gap for us to address in the next chapter.

A lot of solutions have been proposed for providing innovative and efficient intrusion detection for IoT. Some of them utilized different conventional machine learning algorithms, whereas others proposed deep learning methods. The authors (Shone et al., 2018) proposed an AutoEncoder-based deep intrusion detection model named Stacked Non-symmetric Deep AutoEncoders (S-NDAE). Their model consists of two main parts: 1) S-NDAE is used for feature extraction, and 2) trained S-NDAE and Random Forest (RF) are used for intrusion classification. The proposed S-NDAE experimented on NSL-KDD and KDD Cup'99 datasets. The model showed promising intrusion detection rates and achieved as high as 85.42% accuracy.

Ma et al. in (T. Ma et al., 2016) proposed a hybrid IDS called Spectral Clustering Deep Neural Network (SCDNN). SCDNN uses Spectral Clustering (SC) to cluster the training and testing dataset into multiple subsets to train and evaluate the train SCDNN model. Authors (Alrawashdeh & Purdy, 2016) utilized the Restricted Boltzmann Machine (RBM) algorithm to detect DoS, U2R, and probing attacks. They used an RBM method for feature learning and then forwarded the weighted result to next RBM layer to form a Deep Belief Network (DBN). Finally, multi-class intrusion detection was performed with a softmax activation function.

Lopez-martin et al. (Lopez-Martin et al., 2017) reported an intrusion detection approach using Conditional VAE called ID-CVAE. The proposed ID-CVAE is an encoder-decoder network and is based on unsupervised learning. ID-CVAE achieves 80.10% intrusion detection accuracy on the NSL-KDD dataset. Although the model performs well in intrusion detection, it is not suitable for resource-constrained IoT devices as the model performs high level feature reconstruction. This makes the model computationally costly and heavy for a low-memory device. Yin et al. in (Yin et al., 2017) proposed an RNN-based intrusion detection model called RNN-IDS. They experimented with different hyper-parameters such as learning rates and the number of hidden nodes to obtain optimal training time and detection accuracy. The model was evaluated using KDDTest+ and KDDTest-21 datasets (Tavallaee et al., 2009b) and obtained 83.28% and 68.55% accuracy, respectively.

Li et al. (Z. Li et al., 2019) experimented on a different number of hidden layers on Long Short-Term Memory (LSTM and) Gated Recurrent Unit (GRU) based deep RNNs approach. The model consists of an extended learning system to perform intrusion classification. The experiments on two benchmark datasets, namely NSL-KDD and BGP, showed the significance of hidden layers in detection accuracy for the proposed neural network. The model obtained significant detection accuracy and F1-score. Despite the promising overall detection accuracy, the proposed model is heavyweight and complex. The proposed model is memory and CPU inefficient for IoT edge devices. The authors, Vinayakumar et al. (Vinayakumar et al., 2019) proposed a scale-hybrid-IDS-AlertNet (SHIA) model based on deep neural networks to monitor network traffic. The proposed system can identify the malicious events for both network and host levels to further alert network administrators. SHIA model evaluated on multiple intrusion datasets and performed better than state-of-the-art machine learning models.

Most network traffic in a real environment is uneven, meaning the attack traffic is considerably lower compared to normal network traffic. This leads to a class imbalance problem which degrades classification accuracy and escalates the FPR of the learning model (Abdulhammed et al., 2018; Vu et al., 2017; Zuech et al., 2021). Some recent research has focused on addressing the data imbalance problem to improve detection accuracy. Many oversampling methods exist, such as ROS (Hayaty et al., 2020), SMOTE (Chawla et al., 2002), ADASYN (H. He et al., 2008), GAN (Alotaibi, 2020; Creswell et al., 2018; Goodfellow et al., 2020), AE (Albahar & Binsawad, 2020; Shone et al., 2018) to solve data/class imbalance problem.

The authors (X. Xu et al., 2020) proposed a deep learning-based intrusion detection model called Log-cosh Conditional Variational AutoEncoder (LCVAE). The model is capable of capturing the complex distribution of original input and generating new samples for specific classes. They utilized the log hyperbolic cosine (log-cosh) loss function in the proposed model. The authors utilized on Convolutional Neural Network (CNN) for intrusion detection. The results show that the proposed model outperforms several state-of-the-art intrusion detection methods. However, the working principle of the log-cosh loss function is similar to mean squared error, which does not strongly affect the occasional wildly wrong prediction. The intrusion detection by the proposed LCVAE model lacks minority attack class detect on rates, and the overall detection rates can further be improved.

The authors (Y. Yang et al., 2019) explored the significance of CVAE to augment data and solve data imbalanced issues in order to improve intrusion classification. An improved CVAE (ICVAE) is used to augment new data samples, and DNN is utilized for classifying intrusion in the system. The ICVAE-DNN model outperforms in detecting minority attack categories. However, they may neglect the cost sensitivity of imbalance intrusion data to generate high-quality synthetic data. The traditional Cross-entropy (CE) loss in ICVAE

may not be able to optimize the latent distribution and may lead to degrading the quality of decoded samples. Therefore, the generated data deviate from observed data, leading the classifier to perform poorly in terms of detecting minority class attacks. For instance, the proposed model obtains only 11.00% and 44.41% of U2R and R2L minority class attack detection rates, respectively.

Although the aforementioned intrusion detection approaches, including data generation methods, succeeded with satisfactory performance, they yet suffer from inferior detection rates, high FPR, and low detection performance of low-frequent, minority, and unknown attack classes. Apart from the minority attack class detection rates, most of the approaches mentioned above did not consider the resource-constrained nature of IoT devices. The proposed models lack lightweight evaluation.

To overcome the data imbalance issues and build a suitable lightweight classification model for IoT, this work proposes a novel intrusion detection Framework, called CFLVAE-LDNN. The CFLVAE-LDNN framework inherits the strengths of VAE and utilizes improved Class-wise Focal Loss (CFL) as an objective function instead of the traditional CE to train the CFLVAE model. CFLVAE-LDNN framework consists of two models and phases: 1) CFLVAE model is trained to generate realistic synthetic data, and 2) Lightweight Deep Neural Network (LDNN) classification model is developed to classify the attack categories. The motive is to improve the intrusion detection accuracy for minority class attacks and make the classification model lightweight and suitable for resource-constrained IoT.

2.12 Chapter Summary

This chapter studied and presented an overview of IoT and its enabling technologies and compared the factors related to implementing a comprehensive security approach in IoT with the traditional Internet. A focus has been given on security attacks based on

IoT architecture. Attack taxonomy and comparisons have been provided. It is crucial to consider IoT architecture, its limitations, and diversity when providing comprehensive protection. Furthermore, the chapter discussed the different factors related to the capacity and limitations of IoT in the design of security solutions. In this regard, the chapter considered the need for IoT security, including the conventional Confidentiality, Integrity, and Availability (CIA) triad.

Unlike other studies, this research aggregated and discussed various advanced security countermeasures, including cryptography, autonomic, and learning-based schemes ensuring secure IoT communication. In contrast, the existing reviews considered only certain types of countermeasures. This review study in this chapter will serve as a useful manual for researchers to access a wide range of security attacks and solutions that may be of benefit to them. Finally, we discussed existing security approaches and their implementation challenges and provided future research directions. For example, although many researchers have proposed lightweight IoT schemes, more research in this field is needed to design holistic, unified, and well-suited security countermeasures for the IoT as a whole.

This chapter also reported how this research derived the research gap by studying a range of the existing literature. This research will provide an in-depth analysis and demonstrate how the data imbalance problem in network traffic affects the performance of learning algorithms in the subsequent chapters.

CHAPTER 3: METHODOLOGY

This chapter proposes a novel intrusion detection framework called CFLVAE-LDNN. The CFLVAE stands for Class-wise Focal Loss Variational Autoencoder, which inherits the strengths of Variational AutoEncoder (VAE) and utilizes improved Focal Loss as an objective function instead of the traditional reconstruction loss (CE). Class-wise FL (CFL) is used to train the VAE model by focusing on the minority class and adjusting weights for each class sample individually.

This chapter reports the finding of a suitable dataset, data preprocessing steps, development of the data generation model and proposing an intrusion detection model. First, a data generation model is developed and trained to balance the intrusion dataset. The proposed data generation technique generates realistic synthetic data for the classifier to provide high detection accuracy. Then, lightweight intrusion detection model is developed to classify and detect intrusion in IoT.

Section 3.1 provides the research process where a summary of the proposed research phases are described. Then, research design and development with its derivative equations are elaborated in section 3.2, followed by the intrusion detection model is proposed and detailed in section 3.3. Finally, section 3.4 summarizes this chapter.

3.1 Research Process

The processes involved in this research in three main phases are described as follows:

Phase one: Problem Definition

In this phase, an extensive studies of the existing research efforts on the problem domain were studied. Security attacks, vulnerabilities and their current countermeasures are studied, analyzed and reported in chapter two.

Phase two: Research Design ad Development

This phase involves developing a security attack model for IoT in the field of study. This phase involved defining CFLVAE-LDNN framework, which consists of four (4) stages: 1) develop and train CFLVAE, 2) data collection and preprocessing, 3) generate realistic synthetic data from trained CFLVAE and balance the training dataset, 4) develop and evaluate lightweight DNN classifier to classify the attack categories. Research design and development are explained in the current chapter.

Phase three: Experimentation, Evaluation and Reporting

This phase involves experimentation and evaluation. The experimentation and implementation details are elaborated on and reported in chapter four. The evaluation performance of the proposed CFLVAE-LDNN model is reported, and comparative studies with the state-of-the-art techniques are reported and shown in chapter five. The evaluation is made in terms of accuracy, precision, recall, F1-score, FPR, class-wise detection rate, AUC_ROC measure, CPU time, memory and energy consumption. The following tools and methods will be used to validate our research results:

- Dataset: Many recent studies relied on the well-known NSL-KDD dataset to validate the proposed CFLVAE-LDNN framework. NSL-KDD is a highly imbalanced network intrusion dataset.
- Environment: This study utilized Python programming language using Keras framework and Tensorflow in the backend. The implementation is done in a Goggle Colaboratory environment in 12GB of RAM. The lightweight DNN classification model is evaluated using the Keras library.

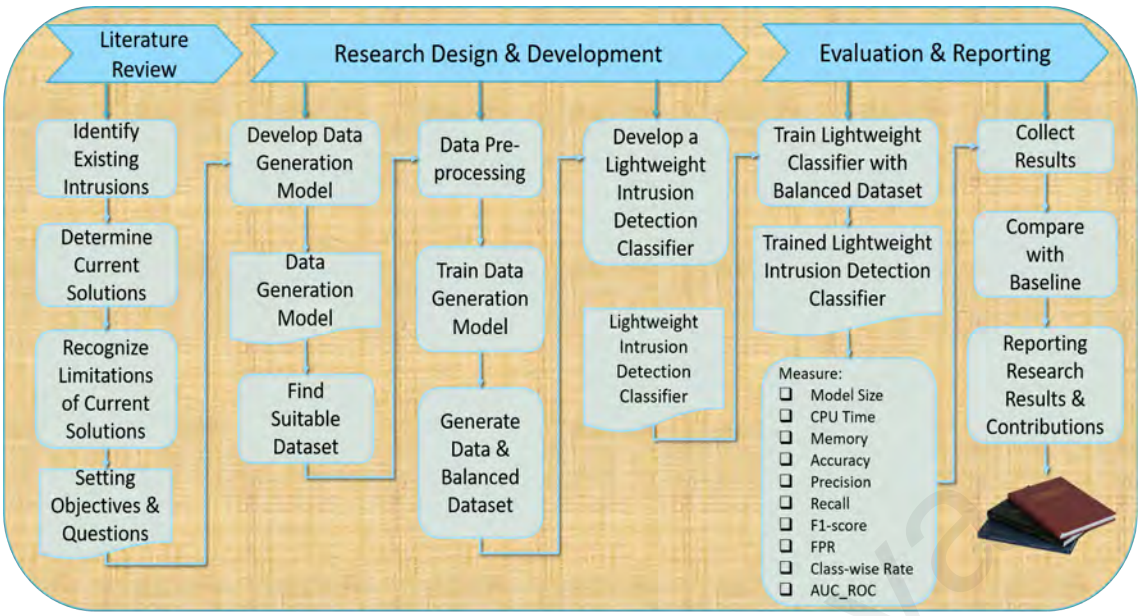


Figure 3.1: Research process.

3.2 Research Design and Development

The proposed CFLVAE-LDNN framework inherits the property of VAE for data generation. However, the VAE is improved by adding CFL as an objective function. The CFL objective function assigns different weight properties to the different target class, leading to generate high quality, diverse and realistic data for minority class attacks. The following sections explain the VAE and how the proposed CFL is incorporated with VAE.

3.2.1 AutoEncoder (AE) and Variational AutoEncoder (VAE)

AutoEncoder (AE) is an artificial neural network that learns how to compress and encode data effectively before learning how to reconstruct data as near to the original input data as feasible (Tschannen et al., 2018). AutoEncoders are used to generate new data samples from existing samples. AutoEncoder (AE) is a type of unsupervised deep learning feature extraction or data generative algorithm, which works by learning best parameters, encoding or compressing input data, utilizing an activation function and finally decompressing/decoding the data back (Shone et al., 2018; Tschannen et al., 2018). The architecture is based on the encoder-latent space-decoder paradigm, which is conventionally

used for dimensionality reduction. The latent space is a fixed vector or data point with a compressed lower dimension (Dong et al., 2018). AE and its variations have been used as a new content generator and a hot research topic for solving data imbalance problems. However, there is no effective way to generate new samples from it as the latent core of AE is not regularized enough.

VAE is a variation of AutoEncoder, which can also generate synthetic data, however, from a probability distribution instead of a fixed data point (Shone et al., 2018). A VAE is different from AE in the sense that the VAE infers the input data samples have some form of latent probability distribution, and then it aims to uncover the properties of the distribution. In other words, VAE learns the input's probability distribution, while autoencoders map each data source directly to a value. A VAE architecture consists of an probabilistic encoder $Q_\phi(Z|X)$ with the learnable parameter ϕ, a latent space Z and a probabilistic decoder $P_\theta(X|Z)$ with the learnable parameter θ (Kingma & Welling, 2013; Y. Yang et al., 2019).

The architecture is based on the encoder-latent space-decoder paradigm. The latent space of VAE is a distribution with mean and variance. Figure 3.2. depicts the VAE architecture with traditional cross-entropy (CE) loss function. In VAE, the encoder transforms the input data to a lower dimension with a probability distribution.

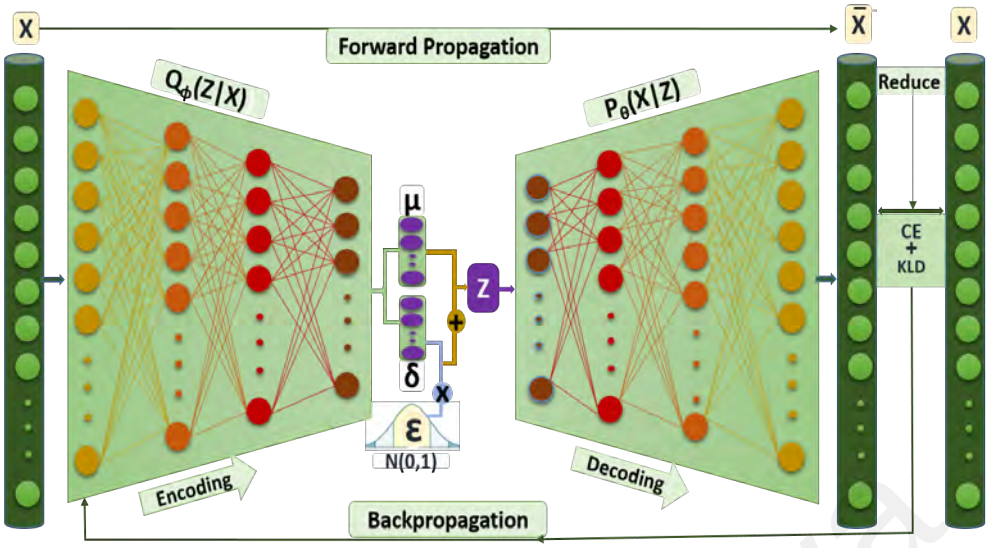


Figure 3.2: Variational AutoEncoder with CE loss.

The working principle of VAE is shown in detail in figure 3.2. Input X has a distribution, and the encoder tends to produce latent code that looks as if they were sampled from Gaussian distribution ($\mathcal{N}(0, I)$) with a mean (μ) and variance (σ). That is, the encoding process of input X takes place through the encoder in the means of forward-propagation. After encoding, we get the latent space Z which is Gaussian distribution with the property of mean (μ) and variance (σ). The Z is then passed to the decoder for decoding it to reconstruct the original dimension $\bar{X}$. During training, the sum of loss (CE + KLD) is backpropagated in order to reduce the total loss through several iterations.

For the latent space Z to have a meaningful abstract property to reconstruct the observed data, the distribution is regularized, and VAE learns variational inference during the training. The encoder network's weight parameter ϕ is learned to encode the input samples to produce encoded feature representation Z. In contrast, the decoder network's weight parameter θ is trained to reproduce new samples by mapping the encoded space Z. However, during the training process, some information is lost and not recovered while decoding. The main goal is to obtain the best encoder-decoder pair that ensures maximum information gain during encoding and has minimum reconstruction error during decoding.

VAE model is widely used to generate data by passing sampled Z to the decoder. During the forward propagation, the reconstruction error (e.g., CE loss) and Kullback–Leibler (KL) divergence loss, $D_{KL}[Q(Z|X)||P(Z)]$ is computed, and the network back-propagates the computed error value.

Hence, the lower bound loss function of VAE is expressed as the equation below (Doersch, 2016; Kingma & Welling, 2013; Y. Yang et al., 2019):

$$\mathcal{L}_{vae}(\phi, \theta, X) = \mathbb{E}[\log P(X|Z)] - D_{KL}[Q(Z|X)||P(Z)] \tag{3.1}$$

where, $\mathbb{E}[logP(X|Z)]$ is the reconstruction loss which is traditionally CE loss used in VAE. Hence, the variation lower bound of VAE can be re-written as:

$$\mathcal{L}_{vae}(\phi, \theta,\ X) = -\log(p_t) - D_{KL}[Q(Z|X)||P(Z)] \tag{3.2}$$

The first part, $(-\log(p_t))$ is the CE loss and the second part is the KL divergence loss. The CE loss is further elaborated on in the next section.

3.2.2 Proposed Class-wise Focal Loss Variational AutoEncoder (CFLVAE)

As mentioned in section 3.2.1, the VAE is used to sample data points from a probability distribution, which matches the prior distribution $P(X)$, where X is a random variable of input data. VAE aims to reconstruct the input data with minimum reconstruction error and reduce the log-likelihood probability of $P(X)$. The improved version of VAE is known as CVAE. The only difference between VAE and CVAE is that in CVAE, the class label y (shown in figure 3.3) is added to train the model for encoding to get the latent space Z for the particular class.

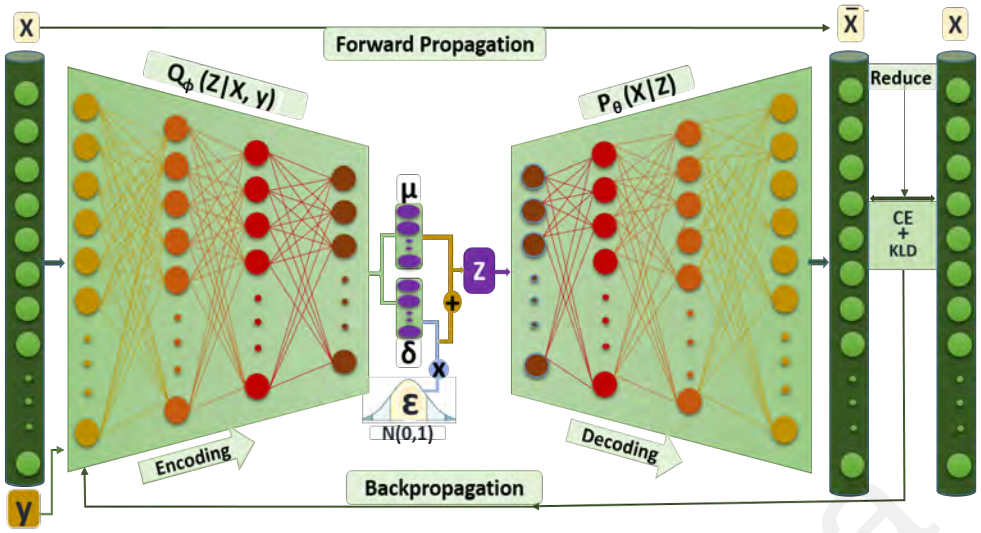


Figure 3.3: Conditional Variational AutoEncoder with CE loss.

The encoder of CVAE can then be expressed as $Q_\phi(Z|X, y)$ and decoder as $P_\theta(X|Z)$ (Kingma et al., 2014; Sohn et al., 2015; Y. Yang et al., 2019). During training, the network learns to encode the best latent distribution Z for a specific class label y. The latent vector Z is then passed to the decoder to reconstruct a new attack vector X for class label y. The loss function of CVAE is computed using the following equation (Kingma et al., 2014; Y. Yang et al., 2019):

$$\mathcal{L}_{cvae}(\phi, \theta, X, y) = -\log(p_t) - D_{KL}[Q(Z|X, y)||P(Z|X)] \tag{3.3}$$

where, $\mathcal{L}_{cvae}(\phi, \theta, X, y)$ is the variation lower bound of CVAE. The first term is called reconstruction loss $\log(X|Z, y)$, which is the typical cross-entropy loss (X. Li et al., 2019) that makes the decoder network learn to recreate back the input data. The second item is known as Kullback–Leibler divergence, $D_{KL}[Q(Z|X, y)||P(Z|X)$. It lowers the distance between encoder $Q_\phi(Z|X)$ and the prior $P(Z)$ distribution. In other words, D_{KL} encourages the learned distribution $Q_\phi(Z|X)$ to be as close to the prior distribution $P(Z)$.

Therefore, the objective of CVAE is to escalate the probability of data reconstruction $\log P_\theta(X|Z, y)$ and reduce the difference between the prior and posterior distribution. The

cross-entropy (CE) loss (X. Li et al., 2019) is defined as follows:

$$CE(p,y) = \begin{cases} -\log(p), & \text{if } y{=}1 \\ -\log(1-p), & \text{otherwise} \end{cases} \tag{3.4}$$

Modifying by refactoring the above CE loss function in simplistic terms, we get p_t:

$$p_t = \begin{cases} -p, y{=}1 \\ -(1-p), & \text{otherwise} \end{cases} \tag{3.5}$$

Finally, the CE loss is therefore expressed as follows by putting eq. 3.5 into eq. 3.4:

$$CE(p_t) = -\log(p_t) \tag{3.6}$$

While using CE as reconstruction loss, the majority attack class in an imbalanced dataset dominates the loss and governs the gradient. Hence, the reconstruction of low-frequency attack class samples differ significantly from the original samples. Consequently, this may result in degrading the quality of generated samples. The generated data combined with the original data is used to train ML/DL classifiers. When the generated data deviates from the original data, these classifiers cannot identify a particular sample belonging to which class. As a result, the learning-based classifier performs poorly in detecting low-frequency attacks.

This research utilized Class-wise Focal Loss CFL to train the VAE in a supervised manner which we termed CFLVAE. To better apprehend the representation and the property in the observed intrusion data and its minority class, this chapter designs a novel objective function called the CFL function for the proposed VAE generative model. In other words, we aim to reconstruct data for a specific minority class and hence, the VAE model is trained

by adding sample data with the class label y using Class-wise Focal Loss. In our proposed model, we passed class label y to both the encoder and decoder. Thus the model encoder of CFLVAE can then be expressed as $Q_\phi(Z|X, y)$ and decoder as $P_\theta(X|Z, y)$.

The architecture and working principle of the proposed CFLVAE is shown in figure 3.4 in detail. Input X has a distribution, and the encoder tends to produce latent code that looks as if they were sampled from Gaussian distribution ($\mathcal{N}(0, I)$) with a mean (μ) and variance (σ). That is, the encoding process of input X for a particular class label y takes place through the encoder in the means of forward propagation. After encoding, we get the latent space Z for each class label y, which is Gaussian distribution with the property of mean (μ) and variance (σ). The Z is concatenated with class label y and then passed to the decoder for decoding it to reconstruct the original dimension $\bar{X}$ for individual class label y. During training, the sum of loss (CFL + KLD) is backpropagated in order to reduce the total loss through several iterations. This process minimizes the loss, and minority attack samples are learned effectively.

As mentioned above, the traditional CE loss in CVAE may not be able to optimize the latent distribution. By using CE as reconstruction loss, the majority class in an imbalanced dataset dominates the loss and governs the gradient. On the other hand, the CFL loss function focuses on the minority class and adjusts weights for each class sample individually, which allows VAE to generate realistic and diverse data to solve data imbalance problems for intrusion detection.

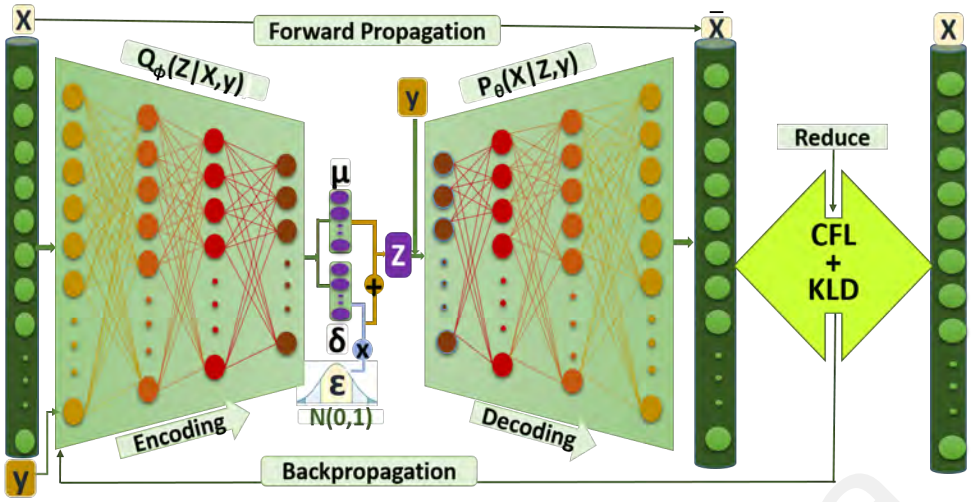


Figure 3.4: Proposed Class-wise Focal Loss Variational AutoEncoder (CFLVAE).

We added a modulating factor $(1-p_t)^\gamma$ with tune-able parameter γ to overcome the issues with CE loss, which is called FL loss (T.-Y. Lin et al., 2017). $(1-p_t)$ is used to consider the hard/misclassified and easy/true negative samples. When p_t is small, $(1-p_t)$ is close to 1 and the loss is unaffected. As $p_t \to 1$, $(1-p_t)$ goes to 0 and the loss for well-classified examples is down-weighted. Formally, the mathematical expression of FL (T.-Y. Lin et al., 2017) is as follows:

$$FL(p_t) = -\alpha_t(1-p_t)^\gamma \log(p_t) \tag{3.7}$$

where, α_t term is added to handle the class imbalance problem where,

$$\alpha_t = \begin{cases} -\alpha & if \;\; y = 1 \\ -(1-\alpha) & \text{otherwise} \end{cases} \tag{3.8}$$

α_t is a weighted term whose value is $-(\alpha)$ for positive class and $-(1-\alpha)$ for negative class. The term α balances the significance of majority/minority examples.

We set different values of $\gamma > 0$ for different classes depending on their imbalance nature to minimize the relative errors for minority classes by paying more attention to

them. The hyper-parameter γ regulates the nature of the loss curve. A larger value of γ leads to a lower loss for minority class samples. We considered several values of $\gamma \epsilon [0, 10]$, as shown in chapter 4, table 4.1, along with other implementation details. The focusing parameter γ smoothly adjusts the rate at which easy examples are down-weighted.

The idea behind the FL loss is to minimize error input from well-recognized examples and maximize the error value for the examples which accept a low loss. Hence, the final CFL loss equation of our proposed CFLVAE model is formulated as below:

$$\mathcal{L}_{cflvae}(\phi, \theta, X, y) = -\alpha_t (1 - p_t)^{\gamma} \log (p_t) - D_{KL}[Q(Z|X, y)||P(Z|y)] \tag{3.9}$$

The first term is the CFL loss $(-\alpha_t (1 - p_t)^{\gamma} \log (p_t))$, which is the reconstruction loss of our proposed CFLVAE.

It is worth mentioning that the effectiveness of Focal Loss has been applied and tested for object detection and computer vision in an imbalanced dataset and attained incredible performance (T.-Y. Lin et al., 2017). The authors implemented FL for imbalance object detection and showed the superiority to CE loss. FL loss is used for cost-sensitive learning to stabilize cross-entropy loss, so rare examples are learned efficiently. Figure 3.5. depicts the comparison between CE and FL loss taken from (T.-Y. Lin et al., 2017). However, the usefulness of FL is not restricted to only computer vision; it is also applied to intrusion detection for imbalanced data issues (Z. Cheng & Chai, 2020).

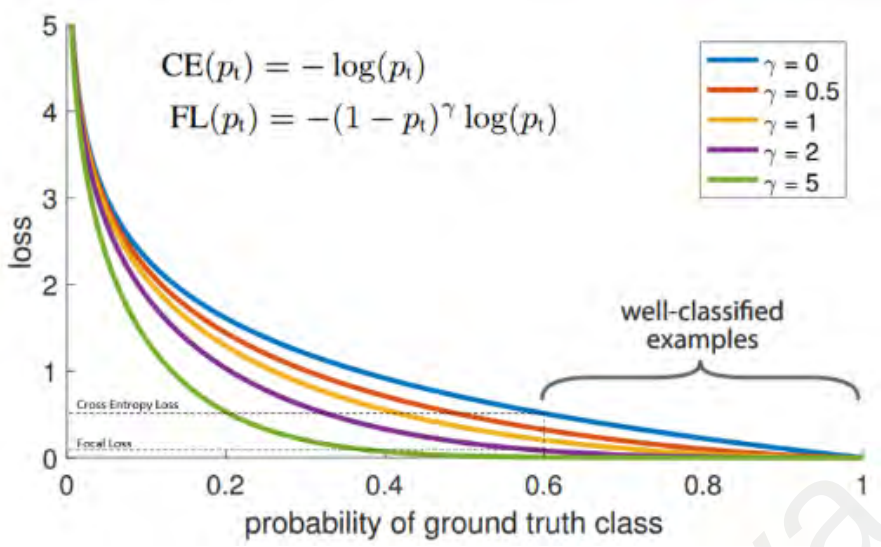


Figure 3.5: Focal Loss vs Cross Entropy loss (T.-Y. Lin et al., 2017).

3.3 Proposed Intrusion Detection Model

The architecture of the proposed CFLVAE-LDNN framework is presented in figure 3.6. The CFLVAE-LDNN framework is mainly comprised of four stages:

- **Data preparation:** Firstly, discrete features are converted to numeric values. Secondly, the features with mostly zeros are eliminated and then, data are normalized between 0 and 1. Finally, feature space is reduced using Mutual Information (MI) technique.
- **Training CFLVAE:** Class-wise FL is added to VAE for cost-sensitive learning to better model the minority class intrusion data. The model is trained to learn a better representation of minority class samples.
- **Data generation:** Generating realistic and diverse synthetic samples for specified minority classes using trained CFLVAE and balancing the dataset.
- **Intrusion detection:** Using the balanced dataset generated from the CFLVAE to train the Lightweight DNN (LDNN) classifier to classify the intrusions.

All these four stages are detailed in the following sections.

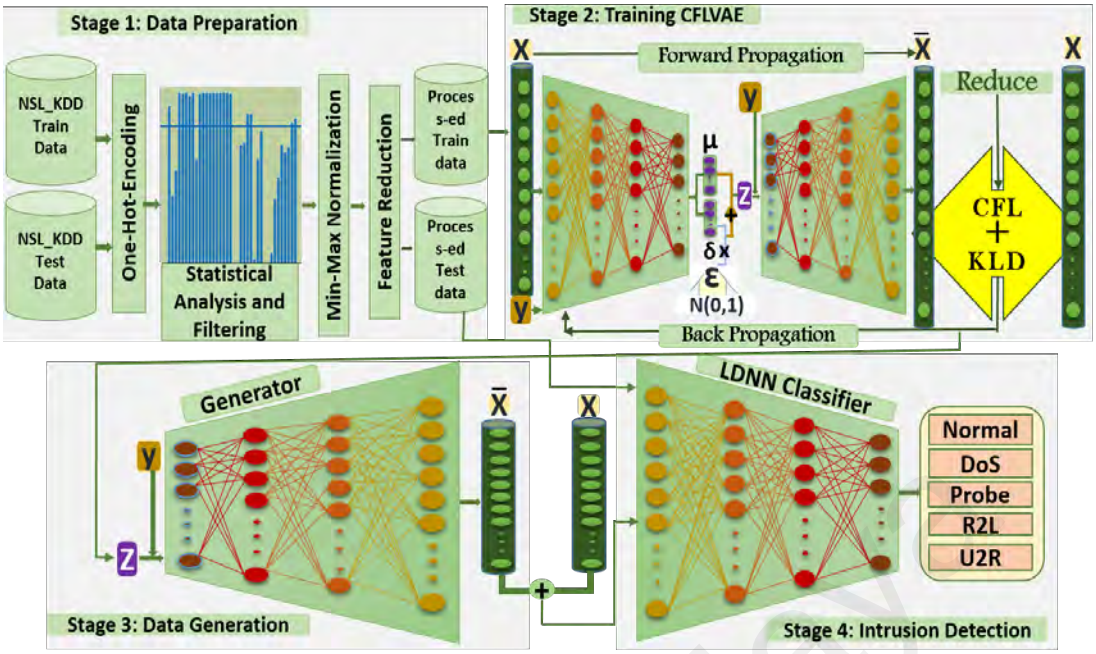


Figure 3.6: Proposed CFLVAE-LDNN framework.

3.3.1 Data Preparation

Data preparation is required for the learning model to be trained efficiently. As shown in figure 3.6, the first stage of CFLVAE-LDNN framework is to prepare the imbalanced data. The dataset is preprocessed using the following steps using Algorithm 1.

Algorithm 1 Data Preparation

Input: Imbalanced raw dataset
Output: Preprocessed dataset

Function:
    Numeration ← One-Hot-Encoding to convert data
    Feature filtering ← filters out features with 90% of zeros
    Normalization ← perform min-max normalization
    Feature reduction ← MI technique to select the best features
    Return scaled dataset with important features
End of the function

- **Feature numeration**: One-Hot encoding (Al-Shehari & Alsowail, 2021; Cassel & Lima, 2006; L. Yu et al., 2022) is one of the most simple, effective and widely used techniques to convert categorical or discrete features to numerical features. It transforms the categorical values to binary vectors with 0s and 1s. 1 corresponds to the existence of a particular categorical value. In the NSL-KDD dataset, there

are three discrete features such as protocol type, service, and flag. We utilized the strength of One-Hot encoding to convert all discrete values to numeric values.

- **Feature filtering**: We eliminated all irrelevant features. The ratio of zeros is computed for each numerical feature, and the features with more than 90% of zero value are removed. The first stage in figure 3.6 depicts the percentage of zeros of each feature in the NSL-KDD dataset, which has been eliminated.
- **Data normalization**: It is important to scale the values to a certain range for the deep learning models to be trained efficiently. NSL-KDD datasets include values with dynamic range. In the linear conversion of the original input, all feature values are scaled to the range [0 – 1] using min-max normalization (Patro & Sahu, 2015) as the following equation:

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{3.10}$$

where, x' is the normalized value, and x is the original value of a dataset.

- **Dimensionality reduction:** Reduction of feature dimension not only simplifies the model but also makes the model lightweight in terms of size, memory and energy consumption during intrusion detection. Additionally, it may help to find the most important features which contribute the most to intrusion detection. Moreover, it minimizes the training time of the model. Feature reduction is made by selecting a subset of the most important features from a large set of related features.

  In this research, a filer method has been used in ranking features according to their level of relevance via several criteria, such as the value of information and correlation. Furthermore, due to the low space and complexity reduction of an IoT device, a common feature selection approach called Mutual Information (MI) has been utilized for feature/dimensionality reduction on the basis of the information

value. According to the authors (Beraha et al., 2019; Dhindsa et al., 2021), The MI between two random variables, X and Y, can be defined as:

$$MI(X;Y) = H(X) - H(X|Y) \tag{3.11}$$

where, $MI(X;Y)$ is the mutual information value for variable X and Y, $H(X)$ denotes the entropy for variable X, and $H(X|Y)$ represents the conditional entropy for X given Y. The output is denoted as the units of bits. MI is an estimation of mutual dependency between two random variables. As such, the measure is symmetrical, meaning that $MI(X;Y) = MI(Y;X)$. The final 87 features are selected to train both CFLVAE and LDNN networks.

3.3.2 Training CFLVAE

The second stage of the CFLVAE-LDNN framework is to train the CFLVAE model for data generation. The dataset is processed and transformed into 87-dimensional features in section 3.3.1. The converted dataset is used to train the CFLVAE model. The architecture of CFLVAE consists of an encoder and a decoder, as shown in figure 3.5. For the encoder $Q_\phi(Z|X,y)$ and decoder as $P_\theta(X|Z,y)$ we use a multivariate Gaussian distribution. The training of the CFLVAE model consists of the following processes. Firstly, the encoder is trained to obtain the best distribution of latent code Z. The encoding process of input X for particular class label y takes place through the encoder in the means of forward propagation. After encoding, we get the latent space Z for each class label y, which is Gaussian distribution with the property of mean (μ) and variance (σ). Secondly, the Z is concatenated with class label y and then passed to the decoder for decoding it to reconstruct the original dimension $\bar{X}$ for individual class labels y. The decoder is trained to recreate the data sample from learned latent distribution Z. The main objective is to

train the proposed CFLVAE to reduce the divergence between the reconstructed sample $\overline{X}$ and the observed sample X, that is, to decrease the D_{KL} loss to recreate data from the Gaussian prior $P(Z)$ and to reduce the CFL loss by learning weights for each class. In other words, the sum of loss (CFL + KLD) is backpropagated in order to reduce the total loss through several iterations. Through this training process, the minority class samples are trained efficiently. In addition, the CFLVAE measures the disparity between the reconstructed samples and the observed sample during the encoding and decoding processes. This leads the CFLVAE to generate high-quality, realistic and diverse data samples for minority classes in the data generation phase.

The training procedure is done in a number of mini-batches and epochs in order for the weight parameters ϕ and θ of the CFLVAE networks to be converged effectively. This research utilized rectified linear unit 6 (ReLU6) (H. Kim et al., 2021; Yarotsky, 2017) as an activation function and Adam optimizer (Kingma & Ba, 2014) to train the CFLVAE generation model. ReLU6 is an improved version of the ReLU function where the maximum size of activation output is limited to 6. This increases robustness when used with low-precision computation. Adam optimizer manually changes the learning rate for each network weight.

3.3.3 Data Generation

After completing the training of the CFLVAE model, the third stage is to generate the new attack samples in order to balance the intrusion dataset. This research made use of a random sampling method to sample data points from the trained CFLVAE. The target minority class label y is concatenated with data points from Z from the trained encoder and fed into the decoder network. In other words, after training the CFLVAE with CFL loss, we pass the encoded Z distribution to the decoder along with its respective class level y to generate the desired number of synthetic attack samples. In the encoder network,

standard normal distribution $Q_\phi(Z|X, y)$ is used to obtain latent space Z. A point from Z is then passed to the decoder $P_\theta(X|Z, y)$, added with standard normal distribution $\mathcal{N}(0, I)$ for respective minority class label y to generate a new training attack sample $(\hat{x}, \hat{y})$, that is, the generated attack sample corresponds to specific attack class y.

Algorithm 2 CFLVAE for generating synthetic data samples

Input: Imbalanced training dataset $X_{train} = x_1, x_2, \ldots..x_n$, hidden layer $h = h_1, h_2 \ldots., h_m$, weight matrix W, latent variable Z, learning rate lr, training epochs ep, batch size m, hyper-parameters α and γ, class label y.

Output: Balanced dataset $\hat{X}$

init: W_{ij}, b_i, for $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$.
init: define CFLVAE network architecture
Train CFLVAE with Gaussian normal distribution:
Repeat
  for training epochs $ep = 1, 2, \ldots, T$
    for divisible batches $m = 0, 1, \ldots\ldots., k - 1$
      Calculate $\mathcal{L}_{cflvae}(\phi, \theta, X, y)$ according to eq. (3.9).
      Optimize CFLVAE by back-propagating $\mathcal{L}_{cflvae}(\phi, \theta, X, y)$
      according to eq. (3.9) and update weights of the CFLVAE network.
    end
  end
Return converged $\mathcal{L}_{cflvae}(\phi, \theta, X, y)$ of eq. (3.9).
Generate new intrusion sample $(\hat{x}, \hat{y}) \epsilon \overline{X}$ from trained CFLVAE.
Merge generated data with original imbalance data to obtain final balanced training dataset $\hat{X}$.

3.3.4 Lightweight DNN for Intrusion Detection

The fourth or final stage of our proposed CFLVAE-LDNN framework to detecting intrusions using the balanced dataset generated from the previous stage. This work proposes a Lightweight Deep Neural Network (LDNN) model as a classifier with a customized architecture for intrusion detection, as shown in figure 3.6, stage four. LDNN is a neural network model comprised of one input, one output and several hidden layers (Aleesa et al., 2021). Our proposed LDNN model is a fully connected feedforward neural network. Apart from the input and output layers, the LDNN architecture initially consists of six hidden layers. We utilized ReLU6 (H. Kim et al., 2021; Yarotsky, 2017) as the activation function of all hidden layers and softmax for the output layer. Figure 3.7 is the elaboration of figure

3.6, stage four, which depicts the LDNN model and its steps in intrusion detection. The first step is to define the model, find the best architecture of the model and train it using the balanced train dataset. In the second step, we compress the network and train again for a few epochs. Then network compression technique is elaborated in section 3.3.4.2. The third and final step is to classify the attacks by feeding the test data into the trained LDNN model.

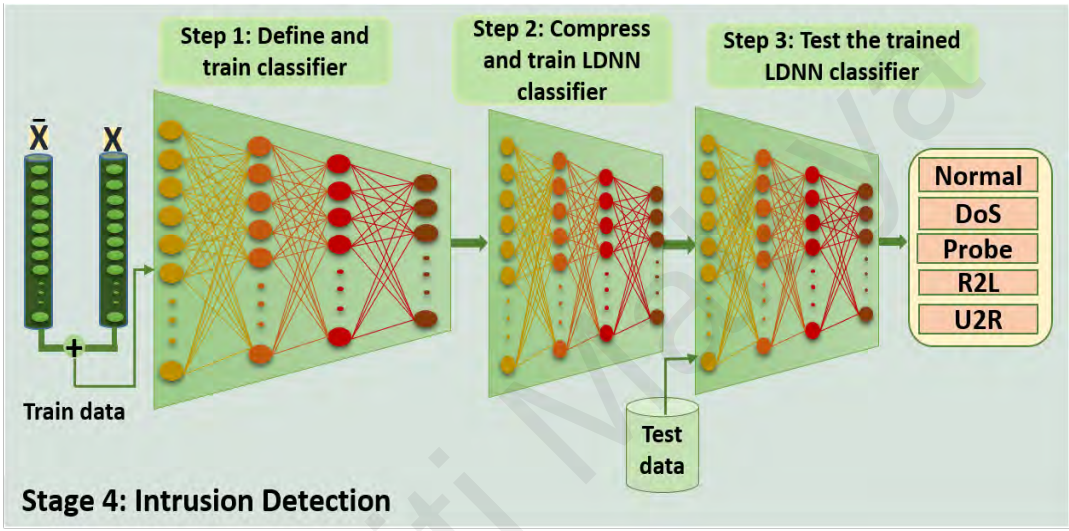


Figure 3.7: Proposed LDNN Model.

As the input dimension of LDNN classifier classifier is the same as the CFLVAE networks, our LDNN model is expected to perform well. The LDNN classifier is able to extract the most relevant attributes automatically. The classifier's weight initialization is done in the same way as CFLVAE networks. The generated minority class samples merged with observed samples are fed into LDNN to train the classifier. However, in order to achieve a lightweight model suitable for IoT in addition to obtaining an optimal intrusion detection performance, finding suitable parameters is necessary when modelling the LDNN classifier (Bisong, 2019; Harrington, 2012). The proposed architecture of our LDNN classification model is formulated as below:

Input layer: The input layer is the first layer of the LDNN model, which takes a number of features of the dataset.

Fully connected hidden layer: The full connectivity of all units in the preceding layer to all units of the succeeding layer is referred to as a fully connected layer. Extraction of high features is achievable using a fully connected layer which leads the classifier to output higher detection performance. In order to achieve a lightweight model suitable for IoT and obtain an outstanding intrusion detection performance, we have experimented with several hidden layers in search of an optimal architecture. The activation function used is the ReLU6 for fully connected hidden layers.

Batch normalization and regularization: This experiment employs dropout probability with a 0.01 value and batch normalization for the fully connected layers to obviate overfitting and to ensure a speedup training process of the model (T. Kim, 2021). Since the proposed LDNN is a relatively small network, 10% dropout is used as is a weight constraint on the hidden layers to regularize and approximate the neural network (T. Kim, 2021).

Classification (output) layer: The last layer is fully connected, and it outputs the classification of multi-class attacks by making use of the *softmax* activation function.

Cost function: The most frequently implemented loss function for multi-class classification task is the categorical cross-entropy (CE_c) loss function (Jayasinghe et al., 2021; Koidl, 2013). For our LDNN classifier, the CE_c loss function is defined as follows (Jayasinghe et al., 2021; Koidl, 2013):

$$CE_c = \sum_{i}^{n} y_i . \log \hat{y}_i \tag{3.12}$$

where, $\hat{y}$ is the predicted class label.

The proposed LDNN classifier is elaborated in Algorithm 3.

Algorithm 3 LDNN Classifier

Input: Balanced train dataset $\hat{X}$, learning rate lr, training epochs ep, batch size m, test dataset

Output: Classification results

split balanced data set into train and validation sets
init: $W_{ij}, b_i, for\ \ i = 1, \ldots., \mathrm{m},\ j = 1, \ldots, n$
init: define LDNN network architecture
Repeat
  for training epochs ep = 1, 2, . . . , T
    for divisible batches m = 0, 1,. . . . . , k -1
      Train LDNN network
      Calculate loss according to Eq. (3.12)
      Optimize LDNN network by back-propagating the according to Eq. (3.12)
      Validate LDNN using validation dataset X_{valid}
    end
  end
until Eq. (3.12) gains convergence
Input test data to trained LDNN model to evaluate the model performance
Return classification report

3.3.4.1 Finding Best LDNN Architecture

The aim is to find an optimal classifier for efficient memory, energy and testing time in order to maintain the lightweight criteria. Our LDNN classifier contains one input, one output, and several hidden layers. They are fully connected layers, which is a hyperparameter. This research implemented different network architecture by changing the number of hidden layers from six (6) layers to one (1). The optimal network architecture is essential to achieve the optimal detection accuracy and fit the model into resource-constrained IoT devices. The selected model should be as light as possible, taking a low detection time, energy, and memory efficiency simultaneously. In addition it should provide an optimal detection accuracy.

3.3.4.2 Network Compression to Reduce Complexity

The IoT device is energy, memory and processor constrained. However, the DNN model is significantly large to fit into IoT devices. Therefore deploying and running this large model in an IoT device may not be suitable. Hence, the model needs to be optimized to

reduce the model size. Once the model size is minimized, it fits the requirements of limited resources of IoT devices, and the inference becomes much faster (Lei et al., 2020). This section proposes a network compression technique for deploying the detection model on IoT devices. Several ways of neural network compression are available, such as Network Pruning and Network Quantization (Gholami et al., 2021; Liang et al., 2021). Pruning and Quantization are network conversion/compression techniques that minimize model size and improve CPU and hardware accelerator latency. Thus, the model becomes lightweight in terms of size, memory, and CPU consumption in order for the model to be deployed in resource-constrained devices (Chung et al., 2020; Shang et al., 2020; Sudharsan et al., 2020).

Network pruning is an essential network compression method utilize for memory and bandwidth reduction (S. Gao et al., 2021; Liang et al., 2021). Network pruning techniques were introduced to convert a large network into a much smaller one. Network pruning is applied to a trained network which does not require retraining the model (S. Gao et al., 2021; Liang et al., 2021; Molchanov et al., 2019). This technique gets rid of insignificant parameters which do not contribute much to the model performance. Pruning, therefore, minimizes the computational complexity, and as the network size gets smaller, it helps the machine learning model to be deployed in resource-constrained IoT devices (Gholami et al., 2021; Liang et al., 2021). While training, the Network Pruning (also known as weight pruning) gradually zeroes out model weights to reduce the model size (Z. Liu et al., 2018). Network pruning is done via network compression. With this technique, the model can be improved in terms of weights and latency up to 6 times with a minimum amount of degradation in detection accuracy.

Network Quantization on the other hand, compresses the learning model by reducing the width of the datatype (Gholami et al., 2021; Liang et al., 2021). Approximation

of continuous signal with a collection of discrete or numeric values is referred to as quantization. For instance, the compression is performed via converting 32-bits floating points (FP32) with 16-bits or 8-bits integers. The information is preserved through encoding. Quantization of neural network was introduced in the early 1990s (Fiesler et al., 1990). FP32 numbers have traditionally been used to train most neural networks (Sze et al., 2017). However, it has more precision than is required. Quantization can convert this extra precision into lower bits to save computation, energy, memory and storage costs. There are two types of quantization available: 1) Quantization Aware Training (QAT) and 2) Post Training Quantization (PTQ). QAT refers to the quantization that takes into account during the model's training. PTQ is, on the other hand, performed after a model's training. PTQ is a compression technique that can reduce model size with some degradation in model accuracy. In contrast, QAT results in non-negligible accuracy loss (Gholami et al., 2021; Liang et al., 2021). Hence, in this research, we utilized QAT of the LDNN model, which compresses the precision values from 32 bits floating points to 8 bits integers. Thus, the model becomes much more lightweight in terms of size and computational complexity, reducing memory and CPU consumption. The compressed and lightweight classifier is created with the help of Tensorflowlite. As a result, the compressed model gets four times smaller.

3.4 Chapter Summary

In this chapter, an intrusion detection framework called CFLVAE-LDNN was designed and developed. The CFLVAE-LDNN framework consists of two models: CFLVAE data generation model and LDNN intrusion detection model. CFLVAE data generation model combines VAE and Class-wise Focal Loss to reconstruct the observed data samples better and balance the intrusion dataset. The CFLVAE model was proposed to generate diverse and realistic samples for the undetectable minority-class attacks. A Lightweight

Deep Neural Network (LDNN) classification model was designed on generated balanced dataset. Additionally, to meet lightweight criteria for resource-constrained devices, a further improvement is proposed by utilizing the dimensionality reduction and network compression techniques for the LDNN classifier.

CHAPTER 4: EXPERIMENTATION

The proposed model in chapter 3 is extensively experimented with by utilizing the dataset and parameters in this chapter. This study selected a highly imbalanced NSK-KDD intrusion dataset to evaluate the proposed CFLVAE-LDNN model. This chapter presents the details about the benchmark dataset, its preprocessing steps and the implementation details. The dataset and parameters are detailed in sections 4.1 and 4.2. The performance or evaluation metrics of the proposed model are outlined in section 4.3. Finally, section 4.4 summarizes this chapter.

4.1 Imbalanced Dataset

Many recent studies relied on the well-known NSL-KDD dataset (Jianhong, 2015; J. Liu et al., 2020; Lopez-Martin et al., 2017; Ravipati & Abualkibash, 2019; Shone et al., 2018; Tavallaee et al., 2009b; Thomas & Pavithran, 2018) to validate Network IDS (NIDS) and its ML algorithms. NSL-KDD is a highly imbalanced network intrusion dataset. This thesis studies specific intrusions present in this dataset that can impact devices and networks in IoT settings. The dataset comprises of four attack vectors (DoS, Probe, R2L, U2R) and normal network traffic. However, the total attack techniques are not limited to these four. The class imbalance of this dataset is shown in figure 4.1.

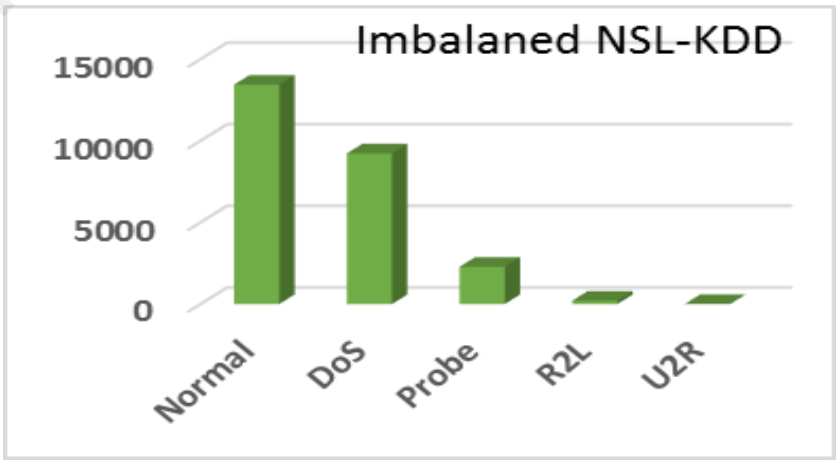


Figure 4.1: Imbalanced original records of NSL-KDD dataset.

Moreover, NSL-KDD has a variety of useful information to detect and mark malicious network traffic. Some of the important features comprise the ability to extract data from the packet header, thereby uncovering the required information. Its content features carry the information about the actual payloads. Time-dependent features enable the study of the traffic request over two seconds. The host-based features access the dynamic behavior over a sequence of active connections. The IPv4&6, TCP, and UDP are widely used protocols in WSNs, whereas FTP, SNMP, ARP, and XTerm are uncommon in WSN environments. Furthermore, few attacks are created for Windows and Linux Operating Systems only. More precisely, DoS and Probe attacks are interesting to be tested in resource-constrained environments.

NSL-KDD dataset is an upgraded version of the KDD-99 (Tavallaee et al., 2009a), aimed to address the redundant records problem of the earlier. The NSL-KDD dataset comprises 125973 samples in total, and there are 25192 (20%) training samples and 22544 (KDDTest+) and 11850 (KDDTest-21) test samples. We utilized 25192 (20%) training samples to train and both test datasets to evaluate our model. NSL-KDD dataset has 41 features: 38 continuous and 3 categorical (discrete values). This study has performed additional data transformation as well.

Furthermore, the skewness of several attack classes present in the NSL-KDD dataset makes it harder to examine the attacks by just using the original class labels. Some intrusion vectors only exist in the test dataset but not in the train dataset, which makes the classifier perform inefficiently. The following section defines the DoS, Probe, R2L and U2R attacks (J. Liu et al., 2020; Tavallaee et al., 2009a) in detail:

DoS – the invader exhausts available computational power or memory space, making the system victim of resource shortage and users are unable to handle routine requests and features.

Probe - this attack enumerates the possible flows or defenselessness of the target network that it leverages to initiate further attacks.

R2L - invader lacks direct access to the target system, so it attempts to obtain local/remote access to a device of the system.

U2R – an intruder tries to enter the network as a benign user and utilizes the weakness of such a system to obtain root access.

4.2 Implementation Details

The proposed CFLVAE-LDNN framework was implemented in Python using TensorFlow[1] as backend with Keras[2] higher-level framework on the GPU-enabled Google Colab[3] with 12 GB RAM. In our proposed CFLVAE, we used fully connected networks for both the encoder and decoder. Apart from the input and output layers, we defined three hidden layers. We implemented the RELU6 (H. Kim et al., 2021; Yarotsky, 2017) activation function to avoid vanishing gradient issues for all hidden layers of encoder and decoder networks. However, Sigmoid is implemented as an activation function for the final layer of the decoder network. The hyper-parameters are defined in table 4.1.

[1] TensorFlow: https://www.tensorflow.org/

[2] Keras: https://keras.io/

[3] Google Colab: https://colab.research.google.com/

Table 4.1: Hyperparameters

Parameters		**Value**
CFLVAE architecture		87-40-20-10-20-40-87
LDNN architecture		87-40-20-10-5
Latent space dimension (Z)		10
Weight initializer		GlorotNormal
Optimizer		Adam
Learning rate(lr)	Value(lr):	10^{-3} to 10^{-5}
	Scheduler name:	Polynomial Decay
	Decay step:	10
	Power:	0.5
Focal loss (Gamma value)		0.50, 1.00, 1.30, 1.50, 2.00, 5.00, 10.00
Focal loss (Alpha value)		0.5 and 0.6
Batch size m		64
Epochs ep (CFLVAE and LDNN)		500 and 200

The optimal network architecture of the proposed generator CFLVAE network is 87-40-20-10-20-40-87 with two hidden layers (e.g., 40 nodes and 20 nodes) for the encoder and two hidden layers (e.g., 20 nodes and 40 nodes) for the decoder and a latent space Z (e.g., 10 nodes). The architecture of the LDNN network is 87-40-20-10-5 with three (e.g., 40, 20 and 10 nodes) hidden layers. In order for the classifier to be lightweight, we have selected three hidden layers after several experiments with multiple hidden layers. Both networks consist of one input (e.g., 87 nodes fore CFLVAE and LDNN) and one output layer (87 nodes for CFLVAE and 5 nodes for LDNN). We proposed novel CFL as the reconstruction objective function and the optimal value of hyper-parameter Gamma (γ) and Alpha (α). The initial value of γ was set to 0.5; according to eq. 7. Seven datasets are created using seven different γ values and used to classify the intrusion detection efficiency for each dataset. After several experiments, the optimal values of γ in the CFL function are obtained as 1.30, which fits for two top minority classes (DoS and Probe) samples, and 1.50 for bottom minority classes (R2L and U2R). The α values in the CFL loss function are set to 0.5 for DoS and Probe and 0.6 for R2L and U2R minority classes.

Thereafter, for both generator and classifier, we use the Adam (Kingma & Ba, 2014)

algorithm with an initial learning rate of 0.001. The learning rate is scheduled with a polynomial decay function with decay steps 10 and a power of 0.5 to optimize the learning parameters of the optimizer (Bukhari & Mohy-ud Din, 2021; Gupta et al., 2019). We utilized GlorotNormal[4] as a weight initializer. The GlorotNormal initializer works fine for our networks as it eliminates the need to guess proper values of fixed limits. The weight matrix is obtained randomly from the normal distribution. Next, we utilized a bias regularizer and the learning is optimized by Adam optimization algorithm (Kingma & Ba, 2014). Adam optimizer has various advantages that make it popular. It has been used as a benchmark for deep learning research and is suggested as the default optimization approach. Furthermore, the method is simple to use, has a shorter running time, consumes less memory, and requires less tuning than other optimization techniques. The value of the bias regularizer is set to 0.0005 for all layers in both generator (CFLVAE) and classifier (LDNN). To evaluate the classifier, we fed the NSL-KDDTest+ and NSL-KDDTest-21 test datasets into trained LDNN to obtain intrusion detection performance.

This research implemented three-fold cross-validation to validate our LDNN classifier. We divide the training dataset into three subsets with an equal fraction of every target class of data. During each training procedure of the classifier, one subset holds out for a testing purpose, and the rest 2 subsets are utilized for training the model. By training the LDNN classifier three times, each subset of the sample takes part in both training and testing.

The learning behaviour of the CFLVAE model and LDNN classifier in the proposed CFLVAE-LDNN framework are depicted in figure 4.2, 4.3 and 4.4. It can be observed that the CFLAVE network converges considerably faster with a minimum number of epochs. The loss reaches very close to four (4) for the CFLVAE and close to 0.05 for the LDNN classifier. The training of the LDNN model also reaches high accuracy faster and converges

[4] Layer weight initialization: https://keras.io/api/layers/initializers

at only 200 epochs.

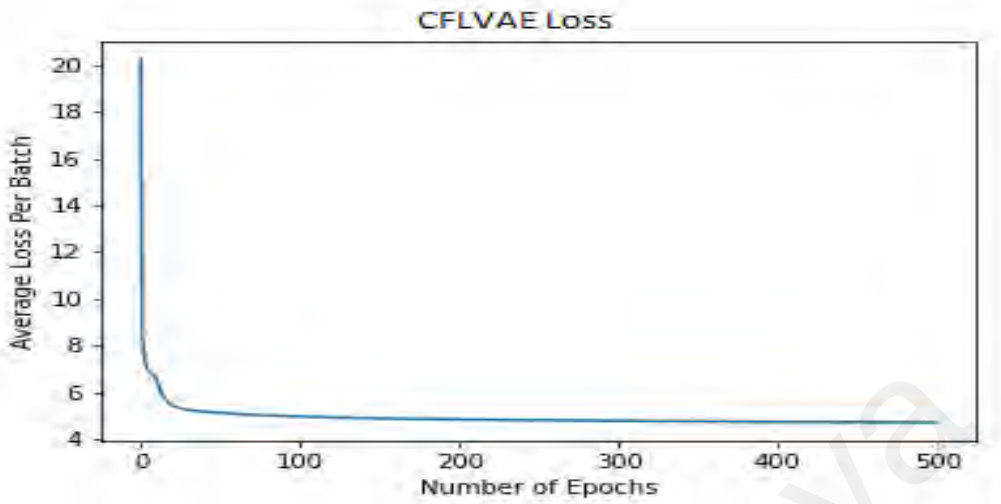


Figure 4.2: CFLVAE average loss.

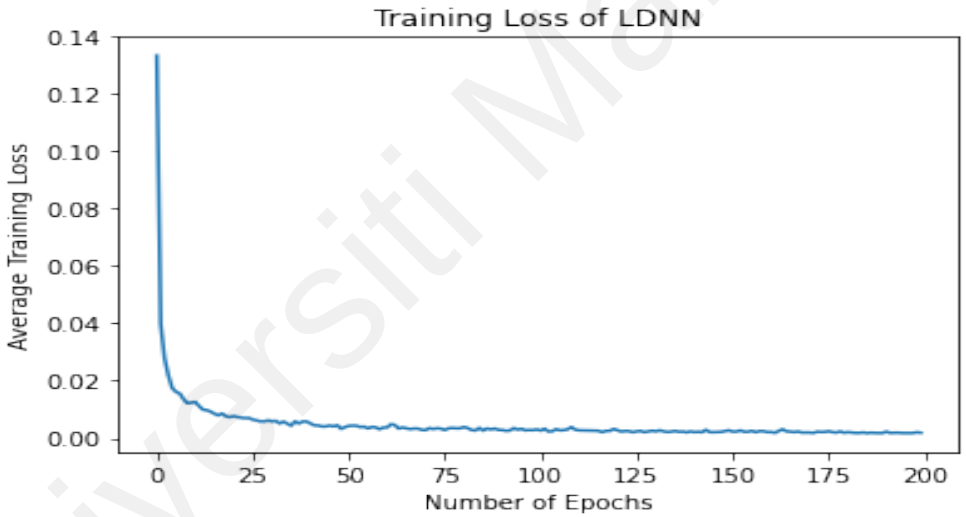


Figure 4.3: LDNN average loss.

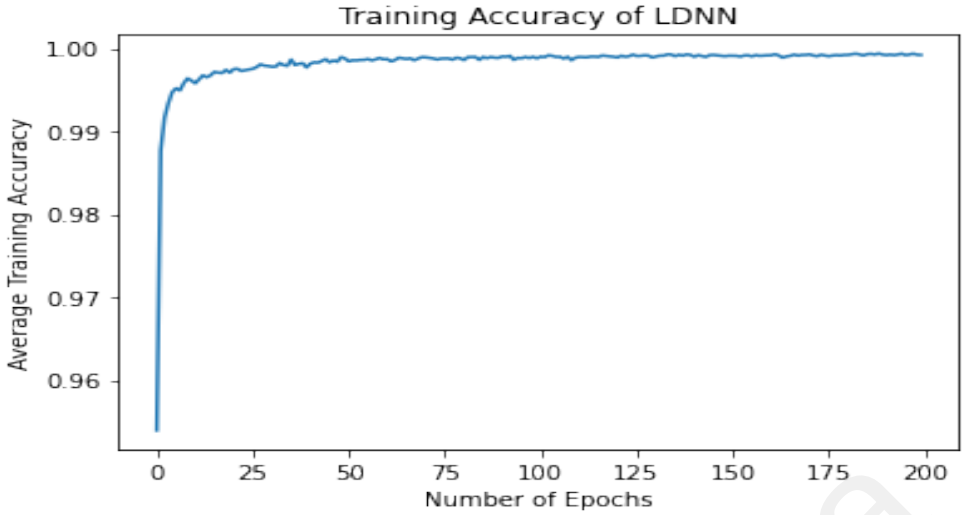


Figure 4.4: LDNN average accuracy.

4.3 Evaluation Metrics

For an adequate evaluation of our proposed intrusion detection model, we have considered the seven most widely used performance metrics, including accuracy, precision, recall, F1-score, False Positive Rate (FPR), and Receiver Operating Characteristic (ROC). In addition, Area Under the ROC Curve (AUC) is also measured to evaluate the performance of our proposed model. The parameters are mainly obtained out of the confusion matrix of detection algorithms (Rácz et al., 2019; Tharwat, 2020).

Likewise, the confusion matrix is formed based on the true positive (tp), true negative (tn), false positive (fp), and false negative (fn) matrix. Correctly predicted traffic is called tp; meanwhile, tn is the number of benign network traffic, which is correctly classified, fp is the number of misclassified traffic and finally, fn the number of traffic incorrectly predicted as benign traffic (Rácz et al., 2019). Indeed, the higher the accuracy, precision, recall, and F1-score, the better the performance of the intrusion detection algorithm. Similarly, the lower value of the FPR is expected for better performance of the detection algorithm.

- **Accuracy:** Accuracy is defined as the ratio of the number of accurately classified

attacks and benign traffic to the total traffic. The higher the accuracy, the better performance of the intrusion detection algorithm. Accuracy is mathematically expressed as follows:

$$\textbf{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn} \tag{4.1}$$

- **Recall:** The recall or Detection Date (DR) is defined as the percentage of correctly predicted actual attacks. The recall is also known as sensitivity or True Positive Rate (TPR). The higher the recall, the better. The mathematical expression of DR is as follows:

$$\textbf{DR/Recall} = \frac{tp}{tp + fn} \tag{4.2}$$

- **Precision:** Precision is the probability of all classified attack traffic, which are true attack traffic. Like accuracy and recall, the higher value of precision confirms the better intrusion detection performance. The precision can be expressed as below:

$$\textbf{Precision} = \frac{tp}{tp + fp} \tag{4.3}$$

- **F1-score:** Another performance metric we are considering to evaluate our model is the F1-score. The F1-score is computed as the harmonic averages of accuracy and detection rate. The F1-score is used to observe the overall performance of the detection model. The higher value of the F1-score ensures a better intrusion detector, with 0 being the worst possible and 1 being the best. The equation of F1-score is defined as:

$$\textbf{F1-score} = \frac{tp}{tp + fp + fn} \tag{4.4}$$

- **False Positive Rate (FPR):** FPR is the measure of the probability of incorrectly predicted benign data traffic. The lower value of the FPR is expected for better detection algorithm performance. The equation of FPR is expressed as:

$$\textbf{FPR} = \frac{fp}{tn + fp} \tag{4.5}$$

- **ROC-AUC measures:** The Receiver Operating Characteristic curve (ROC) is a graphical representation demonstrating a classification model's efficiency over all diverse threshold values. The ROC is a two-dimensional curve of FPR and TPR with possible thresholds for the transition of observation to a particular target variable. The AUC refers to area under ROC curve (Bowers & Zhou, 2019; Narkhede, 2018). A higher value of AUC ensures the superior performance of a classifier. The ideal value of AUC is between 0.5 and 1 for an excellent classifier (Sauka et al., 2022). AUC is expressed as:

$$\textbf{AUC} = \int_0^1 \frac{tp}{tp + fn} d\frac{fp}{tn + fp} \tag{4.6}$$

While ROC denotes of a probability curve, the AUC refers to the degree of separability. It is also can be termed as AUROC (Area Under the ROC). It measures the model's capability to distinguish among various classes in multiclass classification problems. The steeping rate of ROC curve is very important in order to maximize TPR and minimize FPR.

4.4 Chapter Summary

This chapter presented the implementation details, including hyperparameters derived and utilized for this research. A highly imbalanced NSL-KDD intrusion dataset is employed

for the experiments. This chapter also elaborated the evaluation metrics which are used to evaluate the model performance. Since, the proposed CFLVAE-LDNN framework consists of two models: data generation and intrusion detection models. The data generation model is trained with 500 epochs, and the LDNN classification model is trained with 200 epochs.

CHAPTER 5: RESULTS AND DISCUSSION

The proposed CFLVAE-LDNN model has experimented on a highly imbalanced dataset in a Python environment. This chapter presents and discusses the detection performance of CFLVAE-LDNN and comparative studies. Section 5.1 illustrates the model performance, including data generation and intrusion detection performance. The lightweight assessments of the proposed model are also elaborated on with obtained results in this section. Section 5.2 outlines and discusses the comparative hlstudies of our proposed CFLVAE-LDNN with popular data generation models, machine learning algorithms and state-of-the-art approaches. Finally, section 5.3 summarizes this chapter.

5.1 Model Performance

The proposed model generates diverse and realistic data samples for minority classes using Class-wise Focal Loss Variational Autoencoder (CFLVAE) to balance the uneven intrusion dataset. The diverse and realistic generated data samples are merged with the original dataset, and then the diverse, balanced dataset is utilized for training a deep learning-based classifier called Lightweight Deep Neural Network (LDNN) model. The LDNN is a fully connected model with a minimum number of hidden layers and nodes, as mentioned in chapter 3. The following subsections explicitly illustrate the data generation performance of the CFLVAE model and intrusion detection performances of the LDNN model.

5.1.1 Data Generation to Balance Intrusion Dataset

The proposed CFLVAE data augmentation model successfully generates high-quality, diverse and realistic data samples for the minority attack classes. Figure 5.1(a) depicts the severely imbalanced NSL-KDD dataset. It can be seen that the minority classes (i.e., R2L and U2R) contain only a few data samples compared majority class (i.e., Normal) which

(a) Imbalanced original records (b) Generated records (c) Balanced dataset

Figure 5.1: NSL-KDD dataset.

contains a large number of data samples. The data samples are generated by CFLVAE for each minority class and is shown in figure 5.1(b). Finally, the generated data samples are merged with the original data samples to creatre a balanced dataset. Figure 5.1(c) presents the balanced datasets.

5.1.2 Intrusion Detection on Lightweight DNN Model

The balanced dataset is used to train the LDNN classifier for intrusion detection. The performance is extracted with regard to the performance matrix mentioned in the section 4.3. The evaluation performance for accuracy, recall, F1-score, precision and FPR of our proposed lightweight DNN classifier is illustrated in this section. Table 5.1 presents the overall performance of our proposed lightweight DNN classification model on generated data using CFLVAE. The overall performance (in %) of our model is demonstrated in table 5.1 as follows: accuracy of 88.08%, recall of 88.02%, precision of 88.73% and F1-score of 87.69% when tested with KDDtest+ test dataset and accuracy of 76.22%, recall of 76.21%, precision of 80.16% and F1-score of 76.66% when tested by KDDtest-21 test dataset. As we mentioned in chapter four that these datasets are dedicated test datasets belonging to the NSL-KDD intrusion detection dataset. Our lightweight deep neural network (LDNN) classification model achieved significantly lower FPR as 3.77% and 6.51% KDDtest+ KDDtest-21 test datasets, respectively.

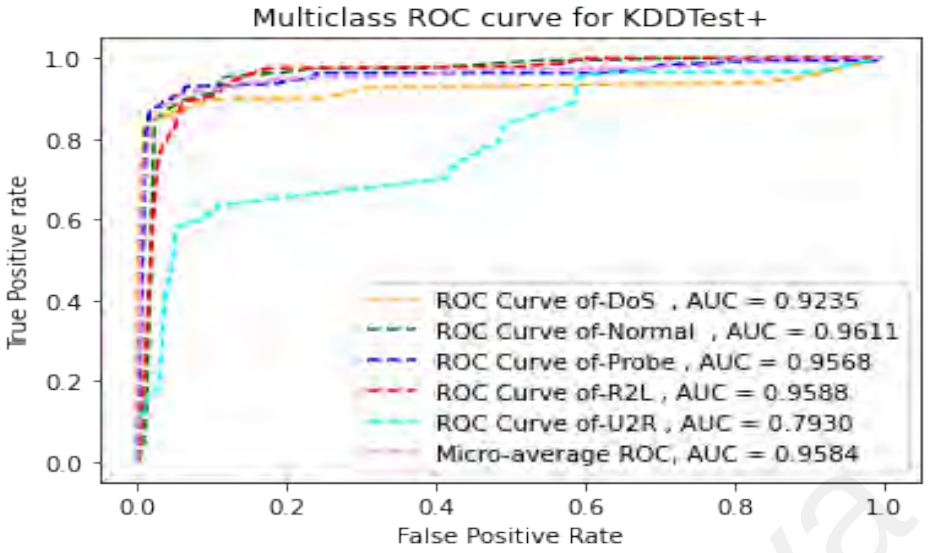
One of the primary objectives of this research is to address and improve minority or

unknown attack detection rates. This research proposed a minority class data generation model which successfully generated diverse and realistic data samples. The generated minority class data samples significantly improved the low frequency attacks detection rates when classifying using the proposed LDNN model. It is interesting to observe that the CFLVAE-LDNN improved the overall performance of minority attack classes. The class-wise detection scores for minority classes are 88.87%, 87.01%, 79.26%, 67.5% for DoS, Probe, R2L and U2R, respectively for the KDDtest+ dataset and 72.28%, 82.82%, 79.25%, 66.00% for the same minority attacks while tested the trained model with KDDtest-21 test dataset. The CFLVAE-LDNN significantly improved the low-frequency attack class detection rates.

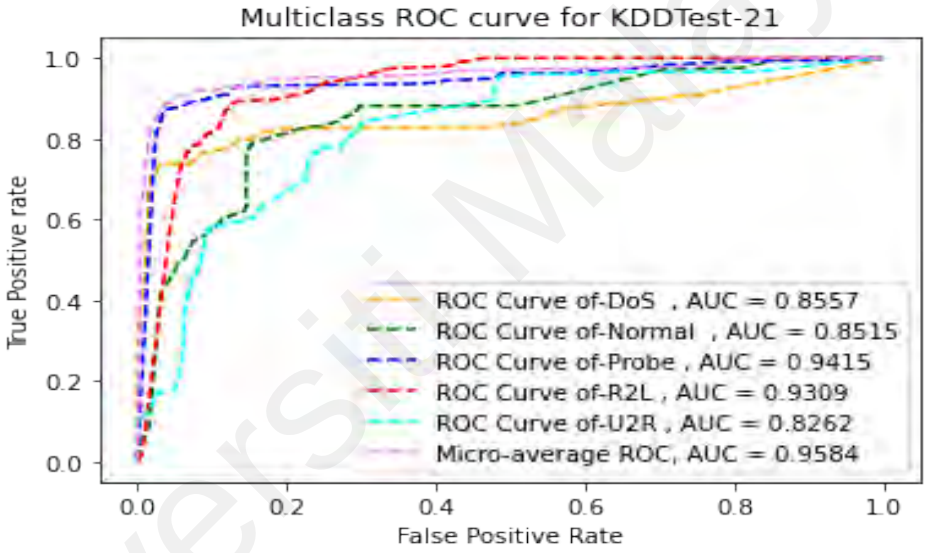
Table 5.1: Intrusion detection performance (in %) of our proposed CFLVAE-LDNN model

Test data	Accuracy	Precision	Recall	F1-score	FPR	Normal	DoS	Probe	R2L	U2R
KDDtest+	88.08	88.02	88.25	87.69	3.77	95.28	88.87	87.01	79.26	67.5
KDDtest-21	76.22	80.16	76.21	76.66	6.51	79.18	72.28	82.82	79.25	66.00

In addition, the ROC curves and AUC values are shown figures 5.2(a) and 5.2(b) for KDDtest+ and KDDtest-21 datasets, respectively. These values play a vital role in order to analyze the overall performance of learning models. ROC is a graphical representation of FPR on the X-axis versus TPR on the Y-axis, which demonstrates the efficiency of a classification model over diverse threshold values. The area under the ROC curve is known as AUC. A higher value of AUC ensures the better performance of the classifier. The ideal value of AUC is between 0.5 and 1 for a good classifier. It is shown in the figures that the AUC values of all classes range between 0.79% and 0.95%, which validates that the



(a) AUC-ROC curve on the KDDTest+



(b) AUC-ROC curve on the KDDTest-21

Figure 5.2: AUC-ROC curve on NSL-KDD test datasets.

proposed CFLVAE-LDNN generates a high-level classification outcome. Hence, the area under ROC curve is inside 0.79% and 0.95% area. This indicates that the classifier with generated data obtained a high TPR and a low FPR.

5.1.2.1 Detection Performance on Different Gamma Values

This research proposed novel CFL as the reconstruction objective function and the optimal value of hyper-parameter Gamma (γ) and Alpha (α) to generate high-quality, diverse and realistic data samples for low-frequency attacks. The initial value of γ was

set to 0.5, according to eq. 3.2 the optimal value of γ in the CFL function is obtained as 1.30, the value fit for two top minority classes (DoS and Probe) samples, and 1.50 for bottom minority classes (R2L and U2R). This research obtained the (γ) values with trial and error experiments. Figure 5.3 shows the detection performance of the LDNN classifier on different datasets generated using different γ values. It is worth mentioning that, we managed to generate seven (7) different datasets using multiple γ values in our proposed CFL loss function for the CFLVAE data generation model. Similarly, the α values in the CFL loss function are set to 0.5 for DoS and Probe and 0.6 for R2L and U2R minority classes. It is interesting to observe that the LDNN classification model obtained very close to 89% and 88% overall accuracy, recall, precision and F1-score when data was generated using 1.30 and 1.5 γ values. The figure also clearly shows that the FPR values are the lowest using the mentioned γ values.

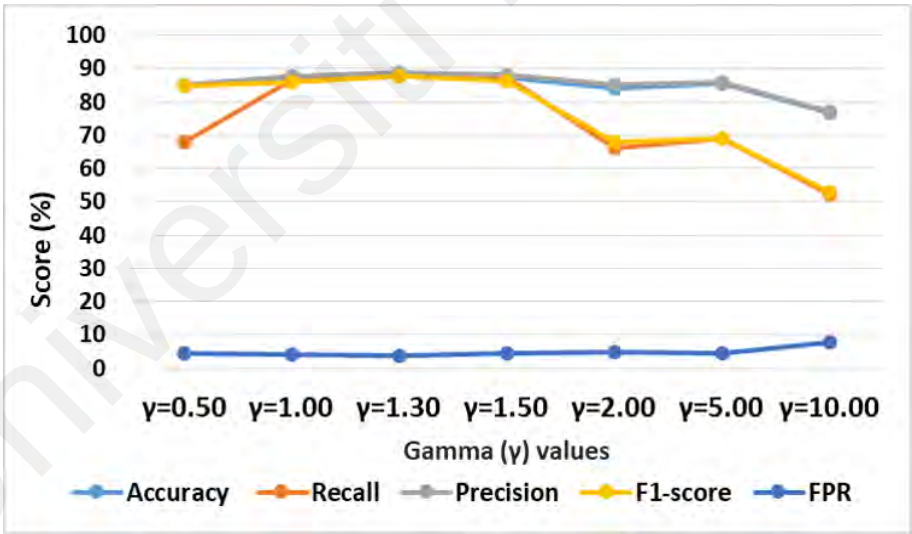
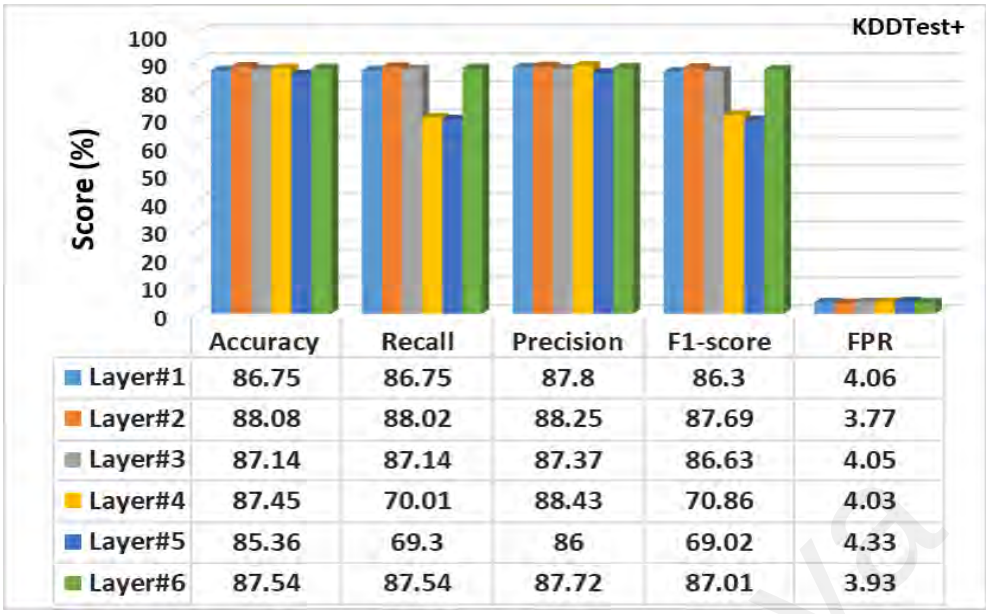


Figure 5.3: The result of intrusion detection performance with different Gamma (γ) values of CFL loss function.

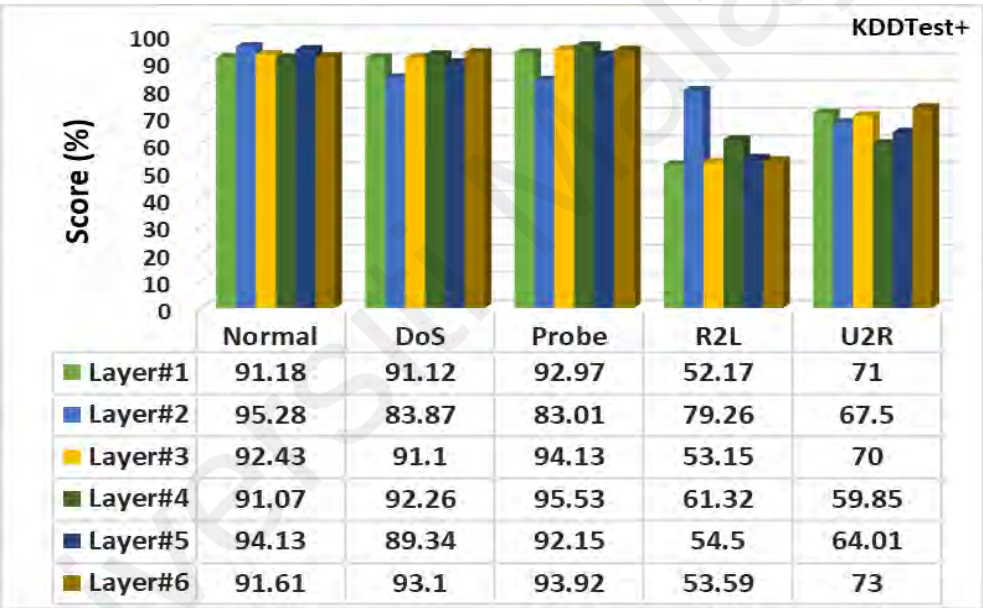
5.1.2.2 Detection Performance on Reduced Network Layers

In this section, the overall intrusion detection performance of the different network architectures has been presented. This research considered several network architectures of the LDNN classification model. The base architecture consists of one input, one output



	Accuracy	Recall	Precision	F1-score	FPR
Layer#1	86.75	86.75	87.8	86.3	4.06
Layer#2	88.08	88.02	88.25	87.69	3.77
Layer#3	87.14	87.14	87.37	86.63	4.05
Layer#4	87.45	70.01	88.43	70.86	4.03
Layer#5	85.36	69.3	86	69.02	4.33
Layer#6	87.54	87.54	87.72	87.01	3.93

(a) Overall performance



	Normal	DoS	Probe	R2L	U2R
Layer#1	91.18	91.12	92.97	52.17	71
Layer#2	95.28	83.87	83.01	79.26	67.5
Layer#3	92.43	91.1	94.13	53.15	70
Layer#4	91.07	92.26	95.53	61.32	59.85
Layer#5	94.13	89.34	92.15	54.5	64.01
Layer#6	91.61	93.1	93.92	53.59	73

(b) Class-wise detection performance

Figure 5.4: Comparison of (a) Overall detection rates and (b) Class-wise detection performance on different numbers of hidden layers used in LDNN classification model (in %).

and six (6) hidden layers. The results of different hidden layers are demonstrated in figure 5.4. To find an optimal network architecture which obtains the highest intrusion detection accuracy while meeting the lightweight criteria, we experimented the classifier model six times with a different number of hidden layers.

Overall detection rates are shown in figure 5.4(a), and Class-wise detection performance

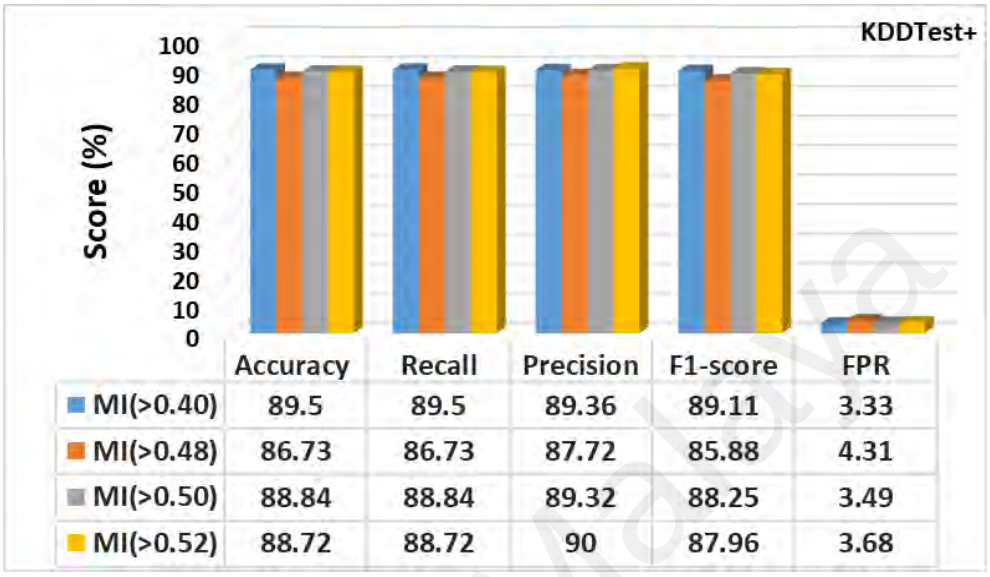
is shown in 5.4(b) on different numbers of hidden layers used in the LDNN classification model. It is interesting to observe that the intrusion detection performance changes with a different number of hidden layers of the LDNN classifier. The highest overall detection performance was achieved using two hidden layers, and the lowest performance was obtained using only one hidden layer on generated data using CFLVAE. The classifier with two (2) hidden layers obtained 88.08% overall intrusion detection accuracy while ensuring the higher low-frequency attacks detection rates of 83.87%, 83.01%, 79.26%, and 67.5% for DoS, Probe, R2L, and U2R attacks, respectively. This also contributes to the proposed LDNN model to become lightweight and suitable for resource-constrained IoT devices.

5.1.2.3 Detection Performance on Reduced Features

In this section, the performance of the feature reduction and feature selection methods are discovered in terms of overall intrusion detection. The Mutual Information (MI) feature selection method (as discussed in section 3.3.1) is utilized to keep the most important features and thus eliminate irrelevant features. The detection results of different values are demonstrated in figure 5.5.

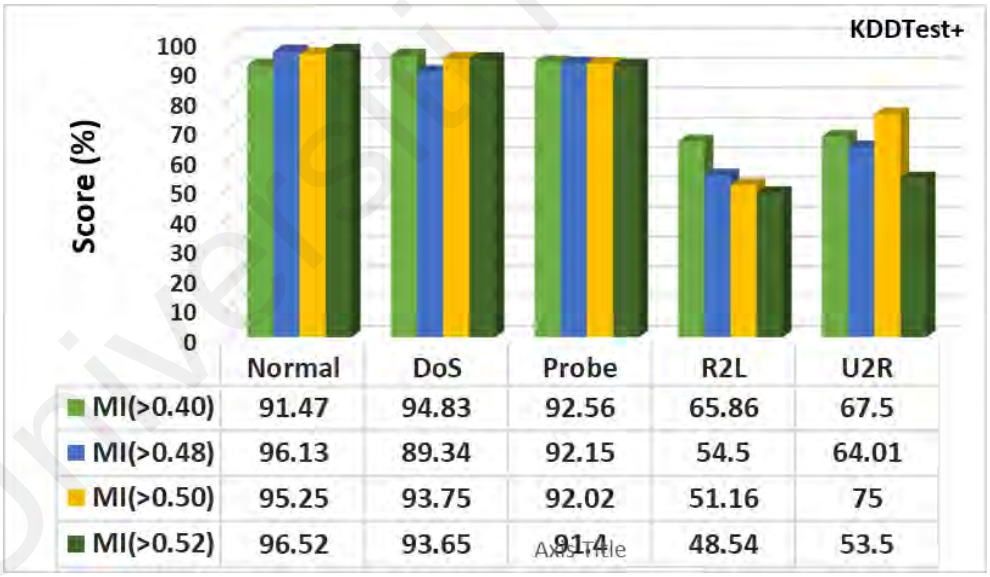
Overall detection rates are shown in figure 5.5(a), and Class-wise detection performance is shown in 5.5(b) on different values of MI used in the LDNN classification model. Remarkably, the intrusion detection performance changes with varying values of MI of the LDNN classifier. This is because different number of feature sets are selected using MI values. For instance, an MI value greater than 40% selects 108 features, an MI value greater than 48% selects 95 features, an MI value greater than 50% selects 87 features, and MI value greater than 52% selects 80 features for our proposed LDNN architecture on generated data using CFLVAE.

The highest overall detection performance (89.5% accuracy) was achieved using MI value greater than 40%, and the lowest performance (86.73% accuracy) was obtained using



	Accuracy	Recall	Precision	F1-score	FPR
MI(>0.40)	89.5	89.5	89.36	89.11	3.33
MI(>0.48)	86.73	86.73	87.72	85.88	4.31
MI(>0.50)	88.84	88.84	89.32	88.25	3.49
MI(>0.52)	88.72	88.72	90	87.96	3.68

(a) Overall performance



	Normal	DoS	Probe	R2L	U2R
MI(>0.40)	91.47	94.83	92.56	65.86	67.5
MI(>0.48)	96.13	89.34	92.15	54.5	64.01
MI(>0.50)	95.25	93.75	92.02	51.16	75
MI(>0.52)	96.52	93.65	91.4	48.54	53.5

(b) Class-wise detection performance

Figure 5.5: Comparison of (a) Overall detection rates and (b) Class-wise detection performance on different values of Mutual Information(MI) used in LDNN classification model on generated data using CFLVAE (in %).

an MI value greater than 48%. The model also performed significantly well when the features selected using an MI value greater than 50%. In this case, the overall accuracy is obtained as 88.84%. It is worth mentioning that the hidden layers of the LDNN architecture are selected as two (2) layers since the architecture provides the highest detection performance, as stated in the previous section. Therefore, considering the analysis, it can be deduced that the model can perform better, provided it contains all the features. However, due to the lightweight requirements, we would like to keep 87 (with MI value greater than 50%) features for our proposed LDNN classifier without compromising the accuracy much. The lightweight assessments are done in section 5.1.3.

5.1.2.4 Detection Performance on Compressed Model

The deep learning model is heavy and not suitable for IoT devices. This section demonstrated the intrusion detection performance of the lightweight version of our classifier. The study utilized the Quantization Aware Training (QAT) (Gholami et al., 2021; Liang et al., 2021) technique (as mentioned in section 3.3.4) to compress our proposed LDNN classifier in order to make it suitable for resource-constrained IoT devices. Table 5.2 shows the detection performance of our lightweight DNN classifier on generated data using CFLVAE. The detection performance is presented for a compressed LDNN classifier using the QAT technique. Overall detection rates and class-wise detection performance is shown for both KDDest+ and KDDTest-21 test datasets.

Table 5.2: Detection performance of Quantization Aware Training (QAT) used in LDNN classification model on generated data from CFLVAE (in %)

Test data	Accuracy	Precision	Recall	F1-score	FPR	Normal	DoS	Probe	R2L	U2R
KDDtest+	88.15	88.73	88.15	87.77	3.75	95.97	88.8	93.14	56.53	68.8
KDDtest-21	76.22	80.16	76.21	76.66	6.51	79.18	72.28	82.82	79.25	66.00

It is worthwhile to state that this performance is deduced after applying the feature reduction method, reducing the number of LDNN hidden layers, and finally utilizing the network compression technique. Hence, this research concludes the model's intrusion detection performance as depicted in table 5.2.

5.1.3 Model Size, Memory Consumption, and CPU Time

Evaluation of lightweight parameters is as illustrated in the subsequent sections. Evaluation of model size, CPU usage and Memory consumption are considered. The analysis takes place in explorations of components that help achieve a lightweight intrusion detection classifier for resource-constrained IoT devices.

5.1.3.1 Model Size

This research analyzed the size of our model based on hidden layers, feature numbers, and compression techniques to evaluate lightweight requirements to make the intrusion detection classifier suitable for IoT devices.

This research experimented on different network architectures for our proposed LDNN classifier to evaluate lightweight parameters. Figure 5.6 shows that the model size (in KB) reduces with the reduced number of hidden layers of the LDNN classifier. The model contains the largest size of 295KB, which has six hidden layers, and the smallest size of 102KB, which has only one hidden layer. However, we observed from section 5.1.2.2 that the model architecture with two hidden layers provides the most promising intrusion detection performance. Hence, to satisfy both lightweight and intrusion detection criteria, we select the model with two hidden layers.

Moreover, different MI values are utilized to obtain the important features to reduce model size. MI values greater than 40% results in 108 features, whereas MI>48% returns 95 features, MI>50% returns 87 important features, and finally, MI>52% returns 80

important features. From figure 5.7, it is evident that the model size in kilobytes (KB) reduces with the reduced number of features. Utilizing an MI value greater than 48% provides 146KB of model size and an MI value greater than 50% provides a model size 140KB, as shown in figure 5.7. However, we observed from section 5.1.2.3 that the model with 108 features (MI>0.40) obtains the highest overall detection accuracy. To meet lightweight criteria, we choose the model with 87 features. The reason is the model with MI>0.50 (87 features) also obtains satisfactory intrusion detection performance with a reduced model size.

Figure 5.8 demonstrates the effect of reducing the model size significantly using the Quantization techniques. Interestingly, the model size was reduced to 19KB using Post Training Quantization (PTQ) and only 7KB using Quantization Aware Training (QAT). In contrast, the model size without compression is 141KB. Thus, the conclusion can be drawn from the above analysis and considering figures 5.6, 5.7, and 5.8 that the combination of two hidden layers and 87 features and the QAT compression technique is the optimal lightweight DNN model size with better intrusion detection performance.

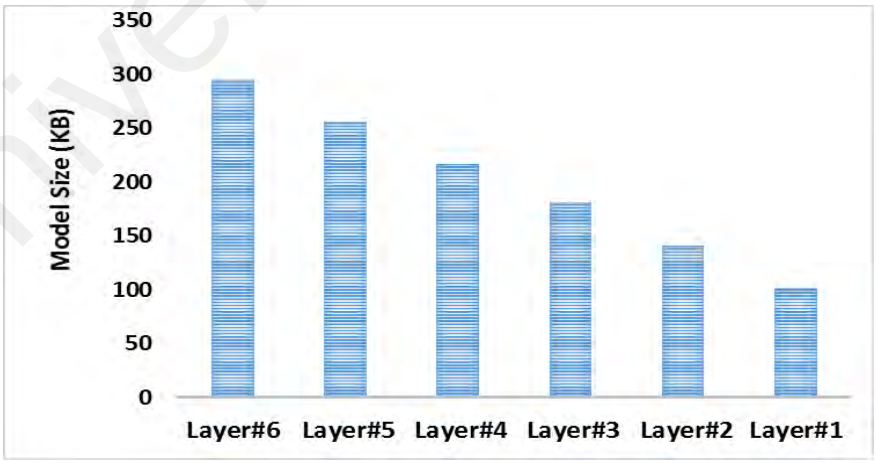


Figure 5.6: Model size with different LDNN architecture.

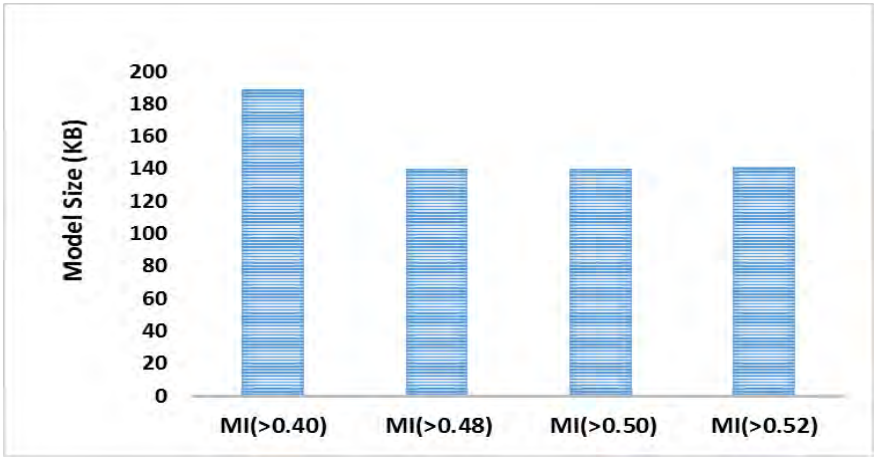


Figure 5.7: Model size on reduced features using MI technique.

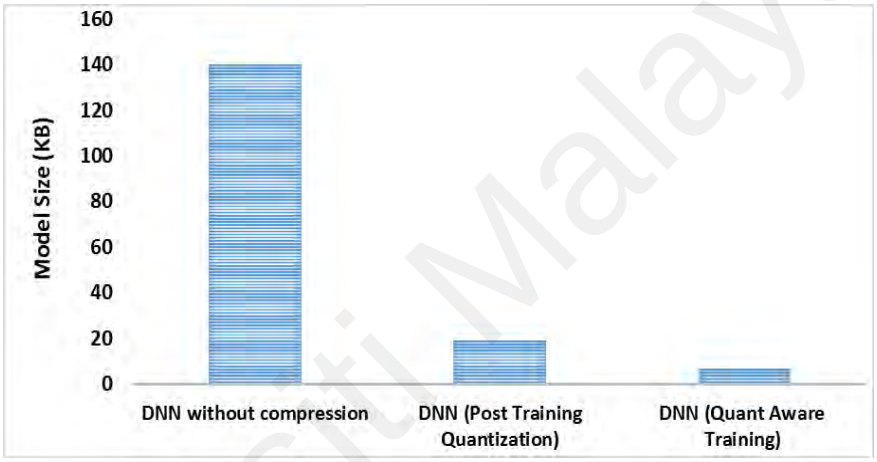


Figure 5.8: Model size using network compression technique.

5.1.3.2 Memory Consumption

In addition, for evaluating lightweight requirements, this research also analyzed the memory consumption by our model during the intrusion detection phase. Figures 5.9, 5.10 and 5.11 presents the memory consumption for intrusion detection by the LDNN classifier using different hidden layers, features, and network compression techniques.

In figure 5.9, it is clearly shown the highest memory consumed by the model, which utilized six (6) hidden layers which is near to 4000 Bytes. The lowest amount of memory is used by the model with only one (1) hidden layer, which is about 2000 Bytes. It is interesting to observe that the memory consumption reduces gradually with the reduction of hidden layers in the LDNN model. In the same fashion, in figure 5.10, it is clearly shown

the highest memory (2300 Bytes) consumed by the model, which utilized 108 features (MI>40%), and the lowest amount of memory (1700 Byes) is consumed by the the model with 86 features (MI>52%). In this case, the memory consumption also reduces with the reduced number of features.

Furthermore, figure 5.11 presents the memory consumption by the LDNN model before and after compression. We compared two most popular network compression techniques: Post Training Quantization (PTQ) and Quantization Aware Training (QAT). A significant improvement in memory consumption is seen after network compression. Before compression, with the reduced number of features and hidden layers, the model consumes approximately 2000 Bytes memory during testing of the trained model. The LDNN consumes around 1700 Bytes memory when applying PTQ compressing technique. Using the QAT compression technique, the model only consumes an average of 1500 Bytes of memory during the detection of an intrusion.

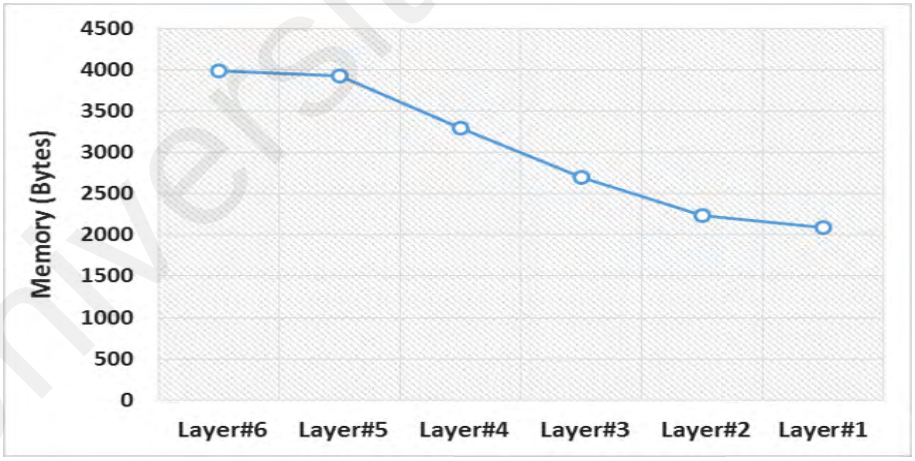


Figure 5.9: memory consumption using different LDNN architecture.

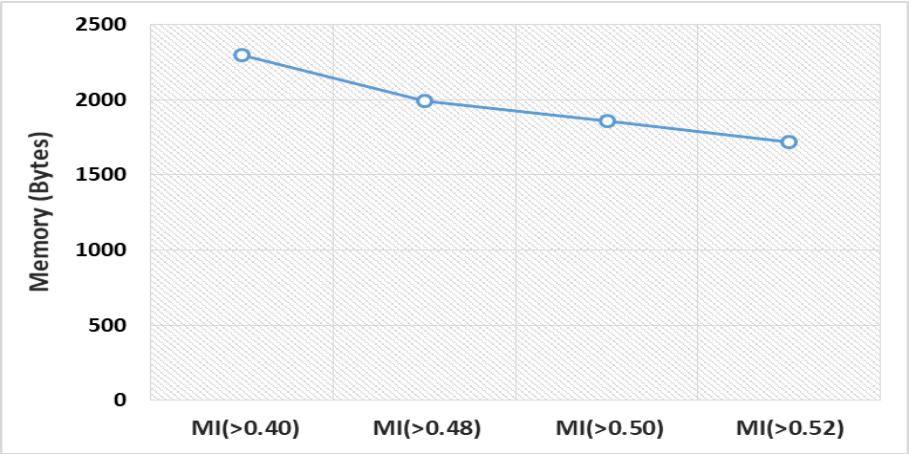


Figure 5.10: memory consumption on reduced features using MI technique.

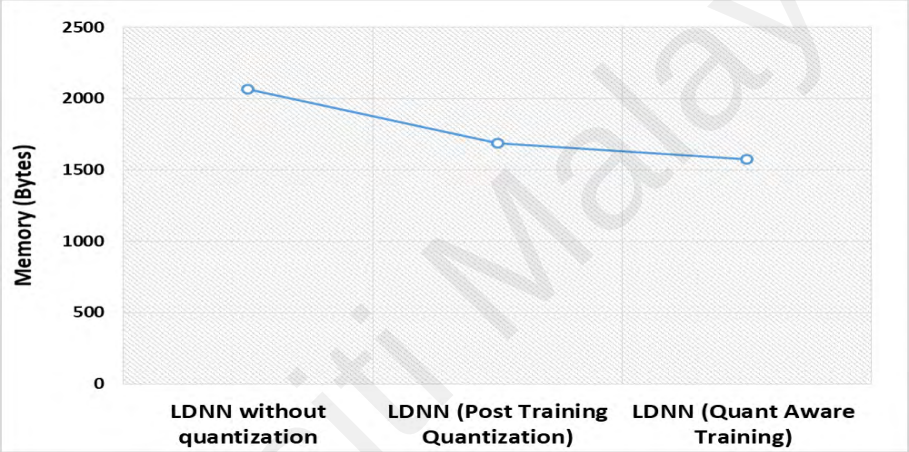


Figure 5.11: Memory consumption using network compression technique.

5.1.3.3 CPU time

This study considers the analysis of CPU time (testing time) of our classification model to evaluate of lightweight requirements of the intrusion detection classifier. Figures 5.12, 5.13, and 5.14 denote the evaluation of CPU time of the LDNN classifier based on hidden layers, feature numbers, and compression techniques. For example, figure 5.13 indicate the CPU time taken by a different number of hidden layers of the LDNN classifier. The model takes a minimum testing time (0.17 seconds) while using one hidden layer and a maximum testing time (0.32 seconds) using six hidden layers.

In contrast, figure 5.13 depicts slightly different results on the test time (CPU time). Different MI values are utilized to obtain the important features to reduce CPU time.

The model takes a minimum testing time (0.163 seconds) with 80 features (MI>52%). The model obtains a maximum testing time (0.18 seconds) with 108 features (MI>40%). However, due to the highest intrusion detection rates obtained from 87 features, we consider utilizing a MI value greater than 50% (MI>50%).

Furthermore, figure 5.14 demonstrates the effect on the CPU time by the proposed LDNN model before and after compression. It is observed from the figure that the CPU time improved slightly during intrusion detection by the model after model compression. The test time of the LDNN model without network compression is 0.15 seconds and 0.13 after utilizing the QAT model compression technique.

Figure 5.12: CPU time (testing) using different LDNN architecture.

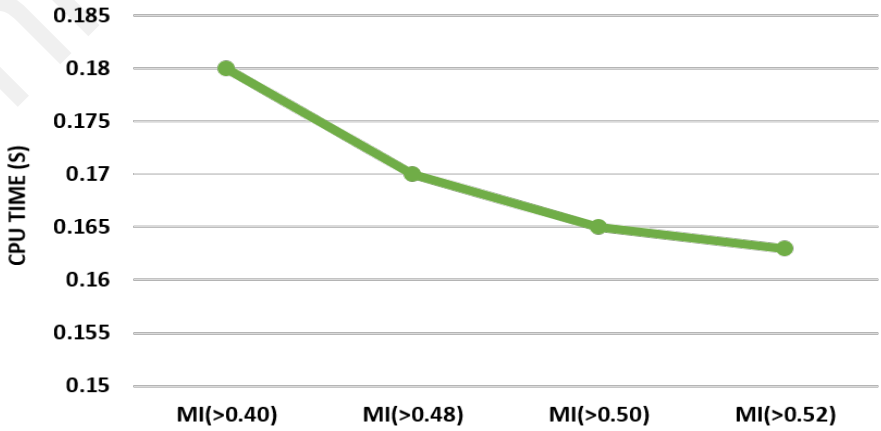


Figure 5.13: CPU time (testing) on reduced features using MI technique.

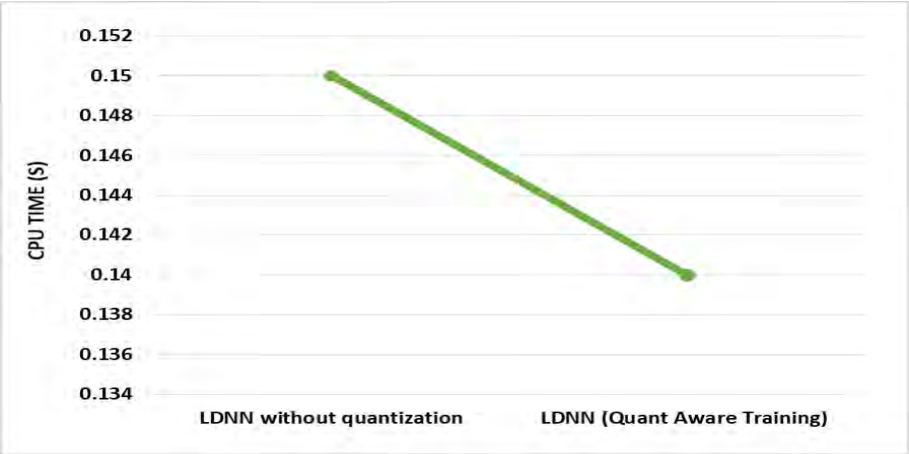


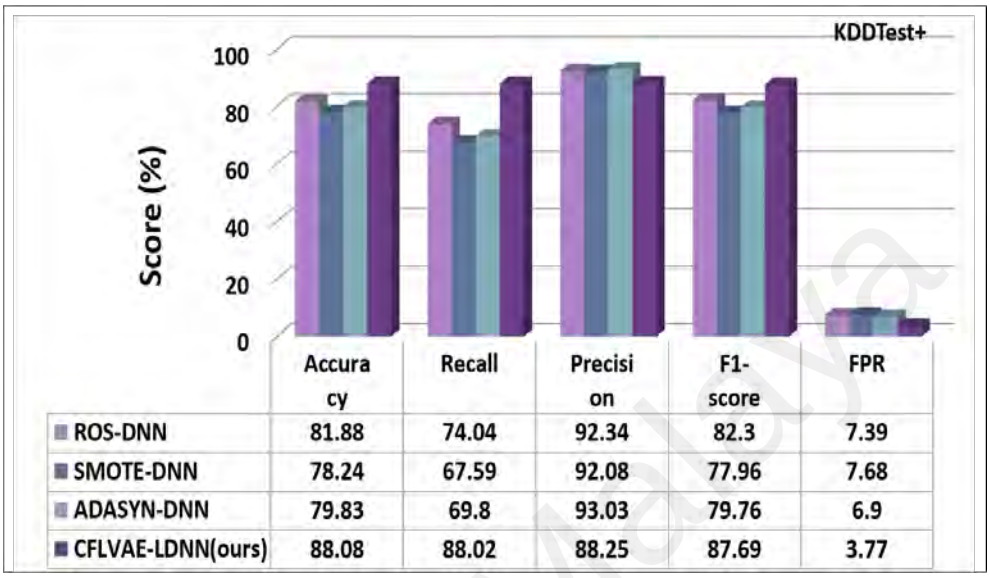
Figure 5.14: CPU time (testing) on network compression technique.

To summarize, it can be concluded that the classifier with two hidden layers, 87 features, and the QAT compression technique provides better overall memory, CPU time efficiency, and acceptable model size without penalizing much overall intrusion detection accuracy and is an optimal lightweight LDNN intrusion detection model.

5.2 Comparative Study

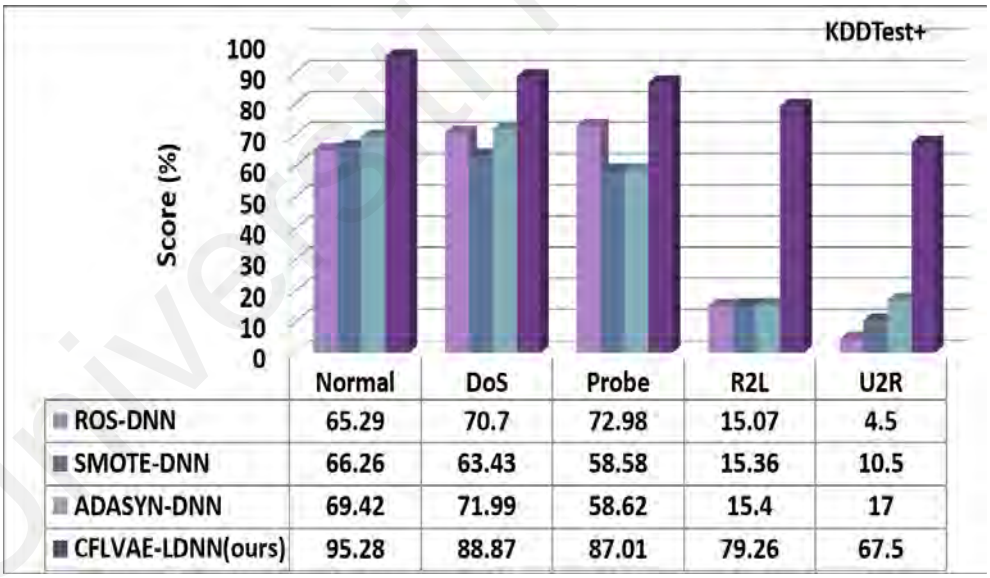
5.2.1 Comparative Study with Data Generation Methods

As mentioned above, the data generation method solves data imbalance issues, resulting in improved overall classification accuracy, including detection rates of the minority class attacks. Random Over Sampler (ROS) (Hayaty et al., 2020), Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al., 2002), and Adaptive Synthetic (ADASYN) (H. He et al., 2008) are the most popular oversampling/data generation methods which have shown significant performance improvement in recent years. Our proposed CFLVAE-LDNN model augment samples for minority and low-frequency attack classes to improve the intrusion detection performance of a deep neural network-based classifier. This research utilized the same LDNN model as the classifier to compare the overall classification result of the proposed CFLVAE-LDNN with the above three most popular data augmentation methods.



	Accuracy	Recall	Precision	F1-score	FPR
ROS-DNN	81.88	74.04	92.34	82.3	7.39
SMOTE-DNN	78.24	67.59	92.08	77.96	7.68
ADASYN-DNN	79.83	69.8	93.03	79.76	6.9
CFLVAE-LDNN(ours)	88.08	88.02	88.25	87.69	3.77

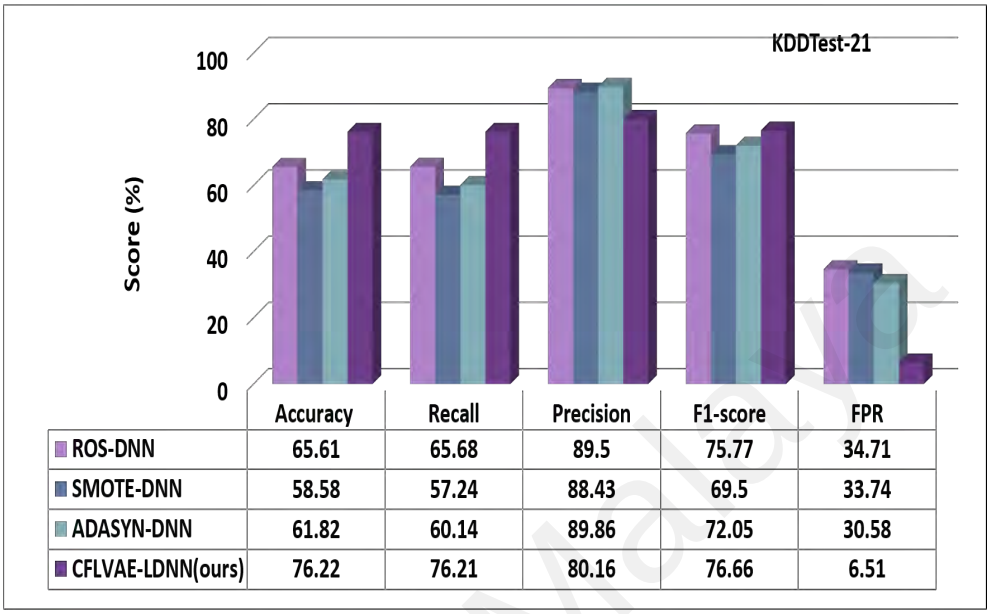
(a) Overall performance



	Normal	DoS	Probe	R2L	U2R
ROS-DNN	65.29	70.7	72.98	15.07	4.5
SMOTE-DNN	66.26	63.43	58.58	15.36	10.5
ADASYN-DNN	69.42	71.99	58.62	15.4	17
CFLVAE-LDNN(ours)	95.28	88.87	87.01	79.26	67.5

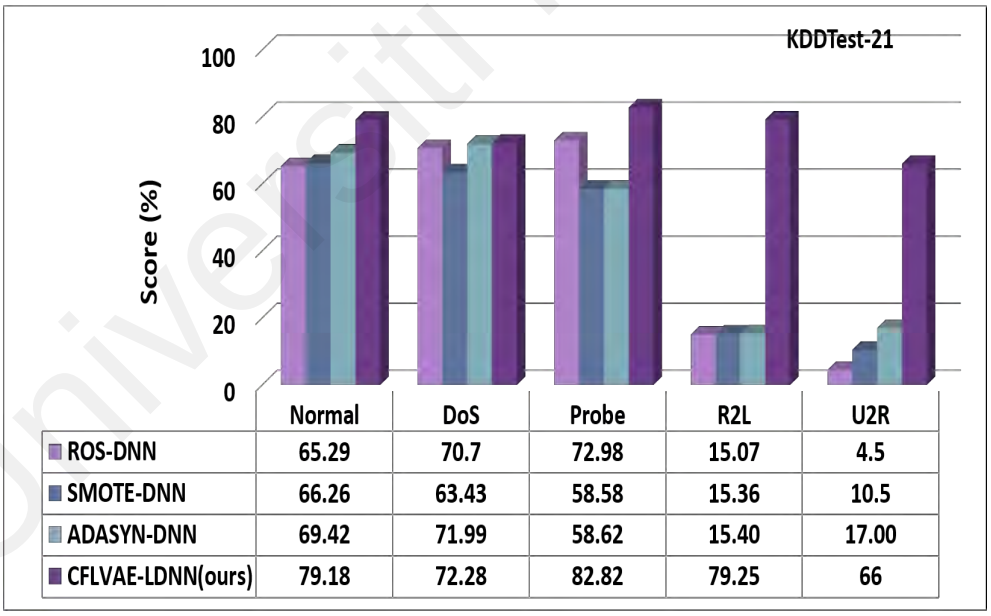
(b) Class-wise detection performance

Figure 5.15: Comparison of (a) Overall detection performance and (b) Class-wise detection performance of popular data generation techniques on the KDDTest+ dataset (in %).



	Accuracy	Recall	Precision	F1-score	FPR
ROS-DNN	65.61	65.68	89.5	75.77	34.71
SMOTE-DNN	58.58	57.24	88.43	69.5	33.74
ADASYN-DNN	61.82	60.14	89.86	72.05	30.58
CFLVAE-LDNN(ours)	76.22	76.21	80.16	76.66	6.51

(a) Overall performance



	Normal	DoS	Probe	R2L	U2R
ROS-DNN	65.29	70.7	72.98	15.07	4.5
SMOTE-DNN	66.26	63.43	58.58	15.36	10.5
ADASYN-DNN	69.42	71.99	58.62	15.40	17.00
CFLVAE-LDNN(ours)	79.18	72.28	82.82	79.25	66

(b) Class-wise detection performance

Figure 5.16: Comparison of (a) Overall detection rates and (b) Class-wise detection performance of data generation techniques on the KDDTest-21 dataset (in %).

Figures 5.15 and 5.16 depict the comparative results of all three methods with the proposed CFLVAE-LDNN. Figure 5.15(a) provides overall performance accuracy, 5.15(b) provides the class-wise detection performance for the KDDTest+ test dataset, figure 5.16(a) provides overall performance accuracy, figure 5.16(b) provides the class-wise detection performance for KDDTest-21 test dataset. It is observed that the CFLVAE-LDNN has achieved the highest overall accuracy, recall, precision, F1-score, and detection rates of minority classes. The proposed CFLVAE-LDNN model achieved approximately 8%, 10%, and 9% higher overall accuracy compared to the ROS-DNN, SMOTE-DNN, and ADASYN-DNN models when testing with the KDDTest+ dataset. Notably, our model achieved approximately 14%, 53%, and 57% higher minority attack class detection rates for Probe, R2L, and U2R classes, respectively, compared with the three mentioned data generation methods. In the case of the KDDTest-21 dataset, our model also achieved the highest overall accuracy, minority class detection rates for all classes, and lowest FPR. Our model has also achieved the lowest FPR (e.g., 3.77% & 6.51% for KDDTest+ and KDDTest-21, respectively). These comparative studies demonstrate that the CFLVAE generates more quality and diverse synthetic samples for the minority attack classes.

The most significant difference between the mentioned benchmark data generation techniques and our proposed CFLVAE is the capability to reconstruct intrusion features from particular attack samples and produce diverse and realistic samples for them. The CFLVAE can generate a corresponding intrusion sample with its properties. The experimental results confirm that data augmented from CFLVAE using class-wise focal loss are more diverse and realistic than the data generated from the benchmark techniques.

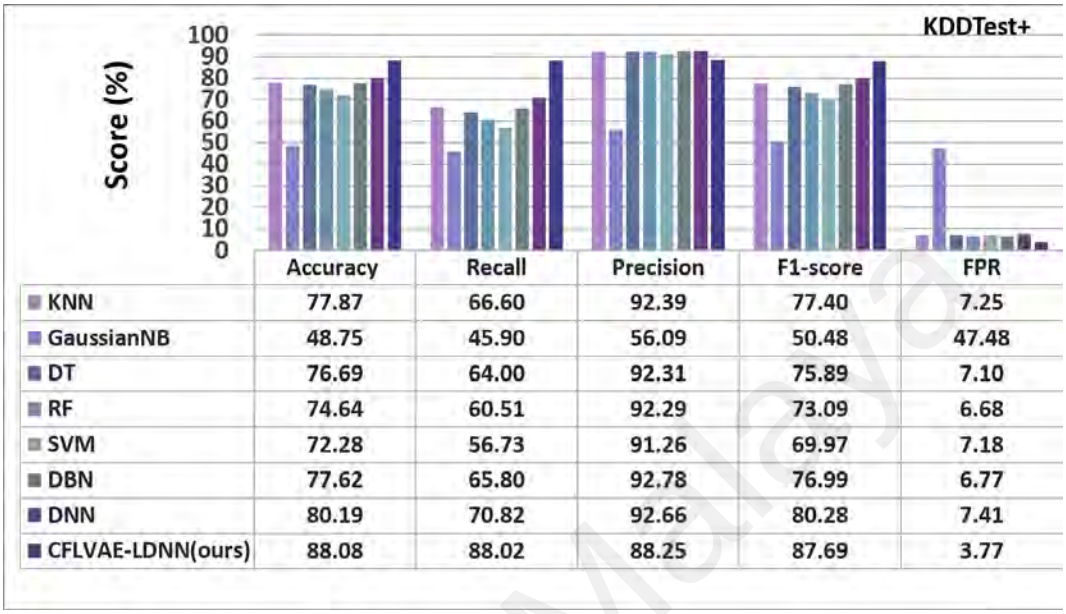
5.2.2 Comparative Study with Learning-Based Classifiers

This research compares the performance of the proposed CFLVAE-LDNN with seven popular and frequently used ML and DL based classifiers, namely, K-Nearest Neighbor

(KNN), Gaussian Naive Bayes (GaussianNB), Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), Deep Belief Network (DBN), and Deep Neural Network (DNN) (Chang et al., 2017; Jianhong, 2015; Y. Yang et al., 2020; Zaman & Lung, 2018). These algorithms are well-established classifiers for their promising performance in intrusion detection and can be found in several pieces of literature. It is worth mentioning that these classifiers are tested on the NSL-KDD original imbalanced dataset. Remarkably, the DNN model is tested without network compression being applied. The DNN model contains six (6) hidden layers.

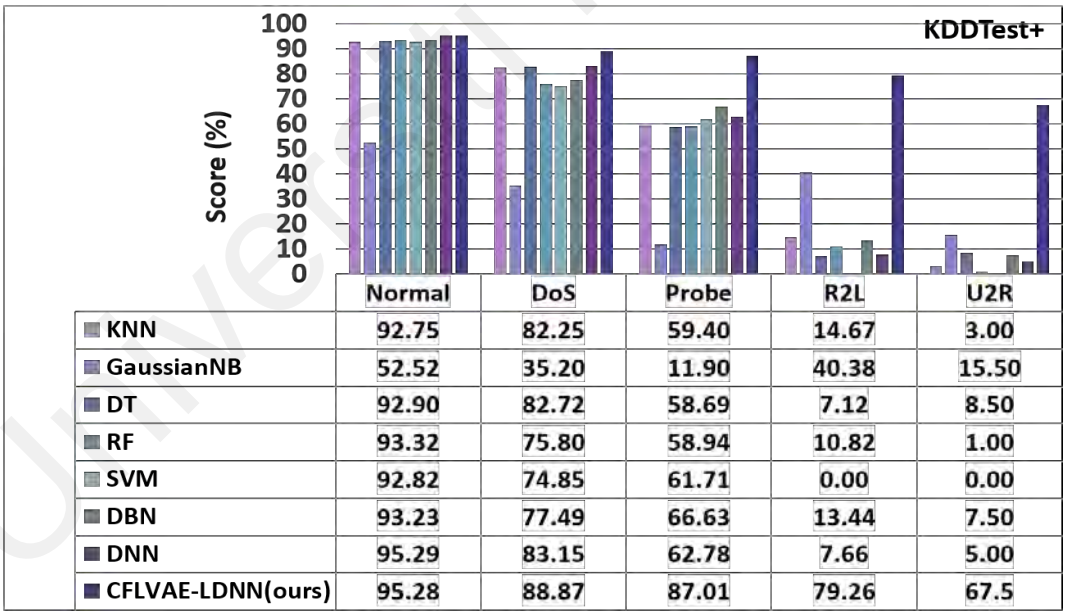
The summary of the comparative studies is presented in figures 5.17 to 5.18. As is observed from figures 5.17(a) and 5.18(a), the CFLVAE-LDNN has a superior detection accuracy (88.08% and 76.22%) and lower FPR (3.77% and 6.51%) among all the well-known classifiers on both the KDDtest+ and KDDtest-21 test datasets. Figure 5.17(a) demonstrates that the proposed CFALVAE-LDNN model achieves the highest recall (by ~18% higher than benchmark) and F1-score (by ~7% higher than benchmark). The precision is slightly higher in KNN (by ~4%), SVM (by ~3%), and DBN (by ~4%) algorithms than our proposed model. The original DNN model without data generation tested on an imbalanced dataset achieved almost 8%, 18%, and 7% lower accuracy, recall and F1-score, respectively, compared to our data generation and classification CFLVAE-LDNN model when tested with the KDDtest+ test dataset. In the KDDtest-21 data in figure 5.18(a), our model obtains the highest overall accuracy, recall, and F1-score. The precision is slightly lower (by ~10%). The FPR of our model is the lowest as 3.77% and 6.51% among all the well-known classifiers for both the KDDtest+ and KDDtest-21 test datasets.

Likewise, figures 5.17(b) and 5.18(b) show that the CFLVAE-LDNN obtains the highest class-wise detection rates for minority attack classes in both the NSL-KDDtest+ and



	Accuracy	Recall	Precision	F1-score	FPR
KNN	77.87	66.60	92.39	77.40	7.25
GaussianNB	48.75	45.90	56.09	50.48	47.48
DT	76.69	64.00	92.31	75.89	7.10
RF	74.64	60.51	92.29	73.09	6.68
SVM	72.28	56.73	91.26	69.97	7.18
DBN	77.62	65.80	92.78	76.99	6.77
DNN	80.19	70.82	92.66	80.28	7.41
CFLVAE-LDNN(ours)	88.08	88.02	88.25	87.69	3.77

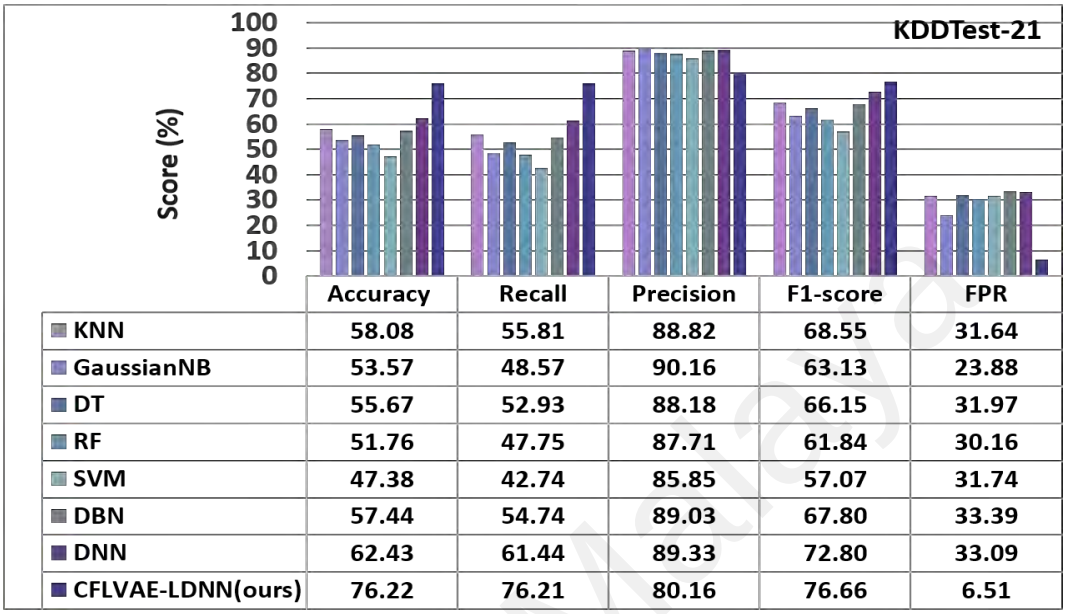
(a) Overall performance



	Normal	DoS	Probe	R2L	U2R
KNN	92.75	82.25	59.40	14.67	3.00
GaussianNB	52.52	35.20	11.90	40.38	15.50
DT	92.90	82.72	58.69	7.12	8.50
RF	93.32	75.80	58.94	10.82	1.00
SVM	92.82	74.85	61.71	0.00	0.00
DBN	93.23	77.49	66.63	13.44	7.50
DNN	95.29	83.15	62.78	7.66	5.00
CFLVAE-LDNN(ours)	95.28	88.87	87.01	79.26	67.5

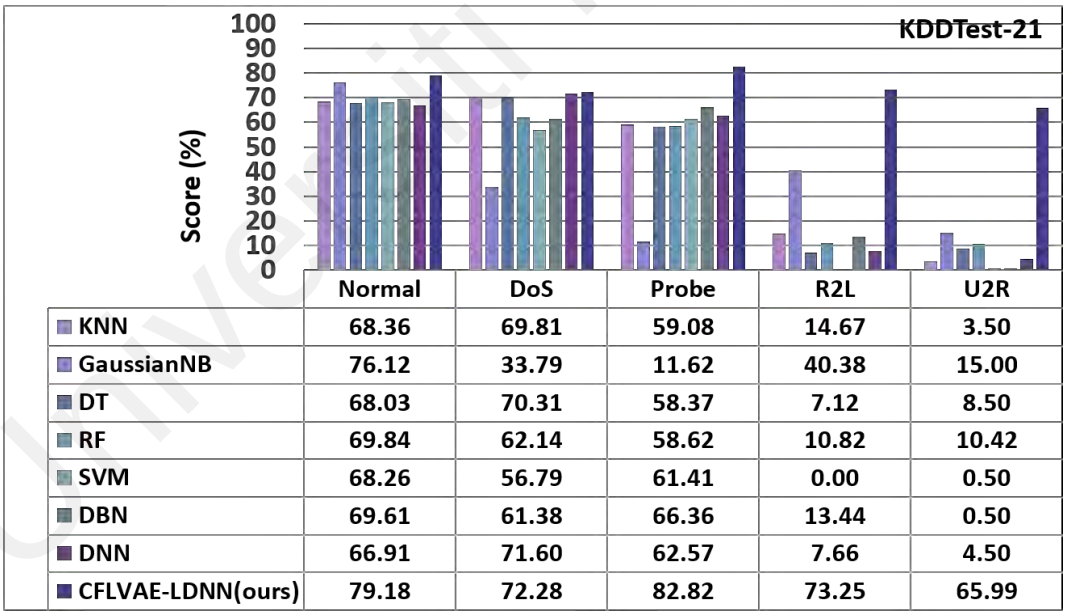
(b) Class-wise detection performance

Figure 5.17: Comparison of (a) Overall performance and (b) Class-wise detection rates of learning-based classifiers on the NSL-KDD (KDDTest+) dataset (in %).



	Accuracy	Recall	Precision	F1-score	FPR
KNN	58.08	55.81	88.82	68.55	31.64
GaussianNB	53.57	48.57	90.16	63.13	23.88
DT	55.67	52.93	88.18	66.15	31.97
RF	51.76	47.75	87.71	61.84	30.16
SVM	47.38	42.74	85.85	57.07	31.74
DBN	57.44	54.74	89.03	67.80	33.39
DNN	62.43	61.44	89.33	72.80	33.09
CFLVAE-LDNN(ours)	76.22	76.21	80.16	76.66	6.51

(a) Overall performance



	Normal	DoS	Probe	R2L	U2R
KNN	68.36	69.81	59.08	14.67	3.50
GaussianNB	76.12	33.79	11.62	40.38	15.00
DT	68.03	70.31	58.37	7.12	8.50
RF	69.84	62.14	58.62	10.82	10.42
SVM	68.26	56.79	61.41	0.00	0.50
DBN	69.61	61.38	66.36	13.44	0.50
DNN	66.91	71.60	62.57	7.66	4.50
CFLVAE-LDNN(ours)	79.18	72.28	82.82	73.25	65.99

(b) Class-wise detection performance

Figure 5.18: Comparison of (a) Overall performance and (b) Class-wise detection rates of learning-based classifiers on the NSL-KDD (KDDTest-21) dataset (in %).

NSL-KDDtest-21 datasets. Compared with other detection models, the proposed CFLVAE-LDNN obtains the highest detection accuracy (in %) on all minor and significant attack types, namely DoS (88.80), Probe (93.14), R2L (56.53) and U2R (68.80) NSL-KDDtest+ and DoS (72.28), Probe (82.82), R2L (79.25) and U2R (66.00) NSL-KDDtest-21 datasets. Hence, the CFLVAE-LDNN model achieves higher detection performance for all classes by synthesizing diverse and realistic data for unknown/minority attack types.

5.2.3 Comparative Study with Related Works

Last but not least, this research has compared the detection performance of our proposed CFLVAE-LDNN with some recently reported intrusion detection techniques to demonstrate the performance of CFLVAE-LDNN. The selected state-of-the-art IDS that are reported in the following research: Improved Conditional Variational AutoEncoder (ICVAE-DNN) (Y. Yang et al., 2019), intrusion detection method based on a Conditional Variational AutoEncoder (ID-CVAE) (Lopez-Martin et al., 2017) Scale-Hybrid-IDS-AlertNet (SHIA) framework (Vinayakumar et al., 2019), Recurrent Neural Network (RNN-IDS) (Yin et al., 2017), Stacked Non-symmetric Deep AutoEncoders (S-NDAE) (Shone et al., 2018), and Log-cosh Conditional Variational AutoEncoder (LCVAE) (X. Xu et al., 2020).

Table 5.3 demonstrates the performance comparisons based on the NSL-KDDTest+ dataset, as the majority of the reported state-of-the-art techniques did not consider the NSL-KDDTest-21 dataset for the evaluation of their models. Therefore, the comparison is made with regard to the performance metrics. It can be concluded that, our proposed obtains the highest overall detection and minority attacks detection rates among all of the related intrusion detection models.

It can be derived from the table that our CFLVAE-LDNN obtains the best detection results in terms of overall accuracy, recall and F1-score among all of the mentioned intrusion detection models. One of the most important evaluation metrics is F1-score

Table 5.3: Comparative study (in %) of CFLVAE-LDNN with the state-of-the-art techniques on the KDDTest+ dataset (NA means not available, *ranked first, **ranked second).

Model	Accuracy	Recall	Precision	F1-score	FPR	Normal	DoS	Probe	R2L	U2R
ICVAE-DNN (Y. Yang et al., 2019)	85.97	77.43	97.39	86.27	2.74*	97.26	85.65	74.97	44.41	11.00
ID-CVAE (Lopez-Martin et al., 2017)	80.1	80.1	81.59	79.00	8.18	91.8	84.41	72.78	33.59	0.057
SHIA (Vinayakumar et al., 2019)	78.5	78.5	80.1	76.5	NA	97.4	76.6	66.3	67.2	24.2
RNN-IDS (Yin et al., 2017)	83.28	73.125	NA	83.22	3.44**	NA	83.49	83.4	24.69	11.5
LCVAE (X. Xu et al., 2020)	85.51	68.9	97.61**	80.78	NA	NA	NA	NA	NA	NA
S-NDAE (Shone et al., 2018)	85.82	85.82	100*	87.37	14.58	99.49	99.79	98.74	9.31	NA
CFLVAE-LDNN (ours)	**88.08***	**88.02***	**88.25**	**87.69***	**3.77**	**95.28**	**88.87****	**87.01****	**79.26***	**67.50***

which is the harmonic mean between precision and recall. Although the precision of our model is negligibly inferior (by 9%) to S-NDAE (Shone et al., 2018) model, the proposed CFLVAE-DNN achieved the highest F1-score among all the cited models.

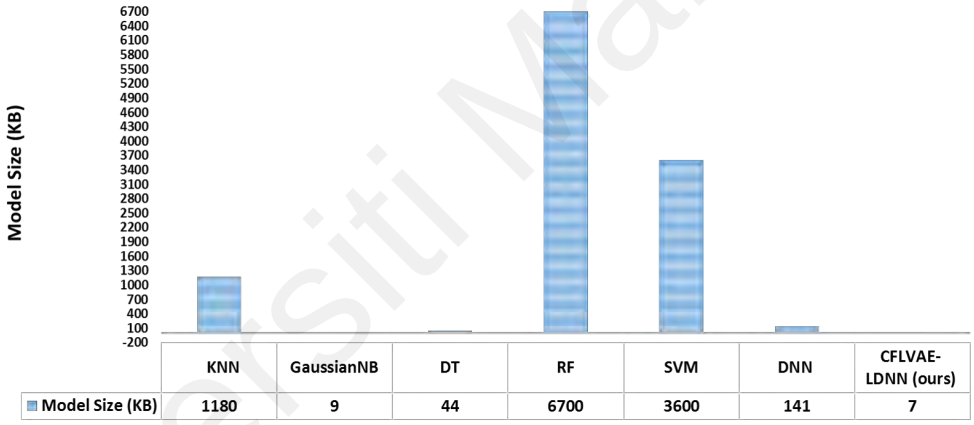
The main aim of CFLVAE-LDNN is to improve the minority attacks defection rates, in addition to improving overall detection performance by solving the data imbalance problem. Our proposed CFLVAE-LDNN achieved the highest detection rates for the two rarest unknown attack vectors. The proposed CFLVAE-LDNN obtained the minority attacks class detection rates of 79.26% and 67% against 44.41% achieved by ICVAE-DNN and 24.2% achieved by SHIA models, for R2L and U2R attacks, respectively. It is observed from the table that, by generating a high-quality sample by the proposed CFLVAE model, our DNN algorithm obtains the highest minority attacks detection rates among all other benchmark models.

The ICVAE-DNN (Y. Yang et al., 2019) scored slightly lower FPR (only 1.01% difference) than our CFLVAE-LDNN model. However, the ICVAE-DNN reported inferior detection accuracy, recall, and F1-score compared to our proposed model. To sum up, the comparative studies demonstrate that the CFLVAE-LDNN intrusion detection is superior in detecting network intrusion effectively. This indicates that the proposed CFLVAE model generates diverse, high-quality, realistic data samples for minority classes to balance the dataset. Furthermore, this led the LDNN classifier to achieve the highest overall accuracy,

F1-score, and minority attack detection rates.

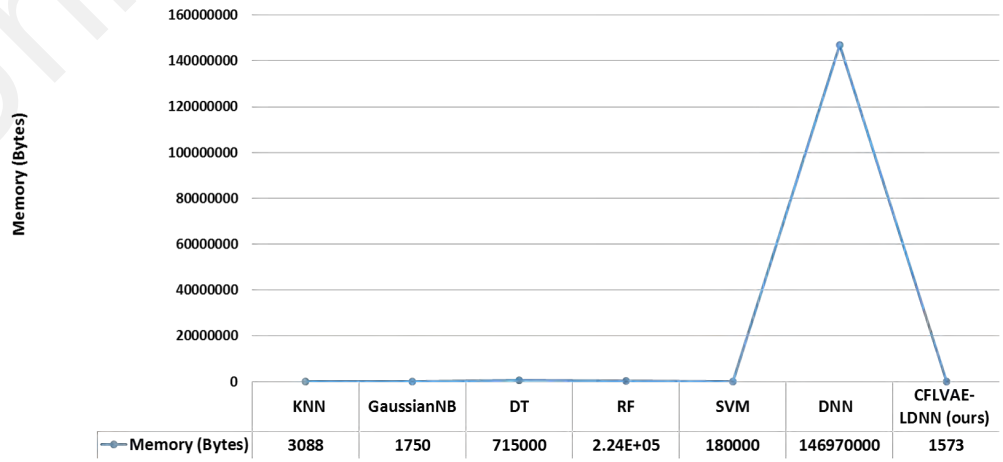
5.2.4 Comparative Study of Model Size, Memory and CPU Time Consumption

To make the model suitable for resource-constrained IoT devices, this research analyzed the model size, memory consumption, and testing time (CPU time). The subsequent sections demonstrate these parameters and comparative study in order for our proposed model to be suitable for IoT. Figures 5.19, 5.20, and 5.21 present the comparative studies of different learning-based algorithms with our LDNN model in terms of model size, memory consumption, and CPU time consumption while utilizing the balanced data from CFLVAE model. The reported results are the averages of 10 instances.



	KNN	GaussianNB	DT	RF	SVM	DNN	CFLVAE-LDNN (ours)
Model Size (KB)	1180	9	44	6700	3600	141	7

Figure 5.19: CPU time (testing).



	KNN	GaussianNB	DT	RF	SVM	DNN	CFLVAE-LDNN (ours)
Memory (Bytes)	3088	1750	715000	2.24E+05	180000	146970000	1573

Figure 5.20: Comparative Study of CPU time (testing).

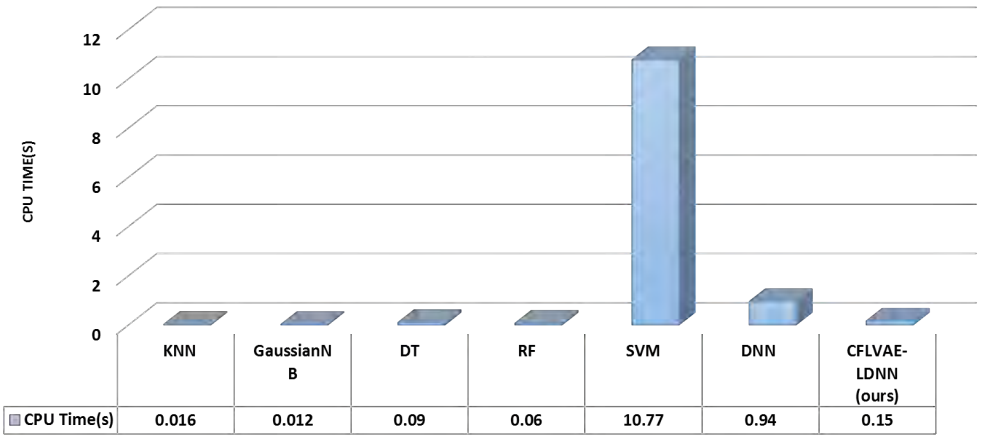


Figure 5.21: CPU time (testing) on reduced features using MI technique.

It is interesting to observe in figure 5.19 that depicts the smallest model size is achieved by our proposed CFLVAE-LDNN, which is only 7KB. This proves that the model is very lightweight for resource-constrained IoT. On the contrary, Random Forest (RF) algorithm seems very heavyweight (6700KB) and Support Vectior Macine (SVM) contains 3600KB of the model size. In terms of memory consumption, figure 5.20 shows the different memory consumption by other algorithms. Interestingly, our proposed model only consumes 1.5KB of memory during intrusion detection, while the conventional DNN model consumes as high as 146970KB of memory while testing the trained model with an instance. According to figure 5.21, the lowest CPU usage is reported by the GaussianNB model (0.012 seconds) followed by the KNN model (0.016 seconds). However, our model consumes a bit higher CPU time of 0.15 seconds. The highest CPU usage is reported by the SVM model.

It can be concluded from figures 5.19, 5.20, and 5.21 that our proposed LDNN model is suitable for resource-constrained IoT systems.

5.3 Chapter Summary

This chapter reported the experimental results of the proposed CFLVAE-LDNN framework. The performance of intrusion detection and lightweight analysis are thoroughly

investigated and illustrated in the figures and tables in terms of performance metrics. This chapter also reported the comparative studies of the proposed CFLVAE-LDNN framework with several existing data generation models, learning-based models, and state-of-the-art intrusion detection models. It is interesting to conclude from the findings that the proposed framework improved the intrusion detection accuracy and minimized FPR. The model also improved the low-frequency attack detection rates by many folds. In terms of lightweight assessments, the proposed CFLVAE-LDNN is lightweight and suitable for the Internet of Things (IoT).

CHAPTER 6: CONCLUSIONS AND FUTURE WORK

This chapter concludes the thesis, highlights the important findings, and suggests some future works. Section 6.1 provides a summary of how this research was conducted. The main finding in the concluding remarks is presented in Section 6.2, followed by the contributions and novelty of this research in section 6.3. Section 6.4 outlines some key strengths and limitations. Subsequently, some possible further research directions are suggested in section 6.5. The chapter ends with some final words in section 6.6.

6.1 Summary

In the first part of the thesis, the literature review was conducted and uncovered the limitations reported in the existing literature. IoT suffers from numerous intrusions, threats, and vulnerabilities. This research outlined those potential intrusions and vulnerabilities. Moreover, numerous studies adopted many intrusion detection techniques, including learning-based and cryptography approaches. This research illustrated the present intrusion detection techniques and reported their advantages (e.g., high overall intrusion detection accuracy) and limitations (e.g., inferior minority-class detection rate) in chapter two.

Various methodological aspects were discovered while conducting investigations on existing literature. One of those aspects is considered for further analysis: the impact of the data imbalance problem in intrusion detection. An experimental study was conducted to confirm that this is undeniably a vital challenge in detecting intrusions, particularly detecting low-frequency attacks. As a matter of fact, data imbalance is one of the many reasons that many traditional machine learning algorithms are inefficient in spotting a specific class of intrusion. Hence, this necessitates us to design and develop a data generation model by combining VAE and Focal Loss to reconstruct the observed data samples better and balance the dataset.

The proposed intrusion detection framework is detailed in chapter three. The proposed model, referred to as CFLVAE, was proved to be effective in generating diverse and realistic samples for undetectable minority-class attacks. Furthermore, the generated balanced dataset is utilized for training and testing a Lightweight Deep Neural Network (LDNN) classification model. However, to meet lightweight criteria for resource-constrained devices, further improvements are implemented by investigating the different architecture of the LDNN model and utilizing the dimensionality reduction and network compression techniques. However, to maintain a lightweight model, there is a negligible trade-off in intrusion detection performances of the lightweight DNN model.

6.2 Findings and Conclusions

The research aims to establish a data generative model using Class-wise Focal Loss Variational Autoencoder and to propose and evaluate a lightweight deep neural network model for Intrusion Detection for IoT. This section outlines the findings and conclusions related to the objectives of this thesis. The findings on RO1 are highlighted in section 6.2.1, RO2 is summarized in section 6.2.2, RO3 in section 6.2.3, and RO4 is recapped in section 6.2.4.

6.2.1 RO1: To identify existing security threats, attacks, intrusions, and vulnerabilities, and to recognize current solutions used for intrusion detection associated with the Internet of Things (IoT) and their limitations.

To achieve RO1, this research has studied and presented security intrusions/attacks based on IoT architecture, and their taxonomy and comparative studies have been discussed in chapter two, section 2.7. Furthermore, this thesis illustrated the aspects related to the capacity and limitations of IoT in the design of intrusion detection approaches. Thus, the research considered the need for IoT security outlined the methods of security attacks and analyzed the actual attacks/intrusions regarding IoT. Likewise, this thesis will serve as

a useful manual for researchers to retrieve a comprehensive list of intrusions that may interest them. Security attacks taxonomy and comparisons have been provided for IoT.

Furthermore, this thesis investigated and reported various intrusion countermeasures, including cryptography, autonomic, and learning-based schemes, detailed in chapter two, section 2.8. Moreover, a discussion on existing intrusion detection approaches, their advantages, and their limitations is provided. Finally, implementation challenges of such intrusion detection algorithms in resource-constrained IoT systems are outlined. Chapter two also reported some state-of-the-art security solutions by studying the range of the existing literature.

6.2.2 RO2: To develop a data generation model to balance a intrusion detection dataset.

This thesis proposes a unique data generative model called CFLVAE for intrusion detection to achieve RO2, elaborated in chapter three, sections 3.2 and 3.3. The model can reconstruct and generates samples for continuous and discrete features present in the NSL-KDD intrusion dataset. CFLVAE incorporates the strength of Variational AutoEncoder and the benefits of Class-wise Focal Loss (CFL) cost-sensitive learning to synthesize the data similar to original data with enough diversity. By implementing the CFL loss function, the minority class attack samples receive more attention, and the CFLVAE data generative model is able to extract the high-level feature distribution of observed samples.

Interestingly, the unique aspect of the model is the ability to generate samples for specific intrusion classes to which the new samples should belong. This improves the intrusion detection performance for minority-class intrusions of deep learning-based classifiers. As a result, the model achieved as high as 88.08% intrusion detection accuracy; 88.87%, 87.01%, 79.26%, and 67.5% for DoS, Probe, R2L, and U2R minority-class attack detection rates, respectively. The high detection rates of minority-class attacks signify that the

proposed CFLVAE is able to produce diverse, quality, and realistic intrusion data samples for specific attack classes.

6.2.3 RO3: To establish a lightweight deep learning model for intrusion detection in IoT.

Chapter three, section 3.3.4, established a Lightweight Deep Neural Network (LDNN) based classifier with unique architecture to achieve RO3 to achieve superior detection performance. The diverse, balanced dataset is used to train the LDNN classifier, which enables the classifier to achieve higher overall detection performance, higher class-wise detection rates, and lower false-positive rates. The experimental results showed that the proposed CFLVAE-LDNN model achieved improved minority-class intrusions detection rates and overall superior performance compared with state-of-the-art data generation and traditional machine learning models on the NSL-KDD dataset.

Moreover, this research experimented with different architectures to make the classifier lightweight and utilized dimensional reduction and network compression techniques. The empirical results demonstrated that the classifier with one input, one output, and two hidden layers. The 87 features and QAT compression provide an overall lightweight model size, memory usage, and CPU time without much penalizing the overall intrusion detection performance. The LDNN achieved 88.08% of overall intrusion detection accuracy, 88.02% recall, 88.25% precision, 87.69% F1-score and as low as 3.77% false positive rate (FPR). The model size, memory, and energy consumption ensure that the model is suitable for IoT devices.

6.2.4 RO 4: To evaluate the performance of the proposed lightweight intrusion detection model for IoT.

This research carried out and reported comparative studies of the intrusion detection performance of the proposed CLFVAE-LDNN model with three popular data over-sampling

algorithms, namely, ROS, SMOTE, and ADASYN. The research also studied comparative analysis using the most common learning-based classifiers such as SVM, DT, RF, NB, DBN, and Gaussian Naive Bayes. The experimental results showed that the synthetic data generated by the proposed CFLVAE offer better overall intrusion detection performance in terms of accuracy, recall, F-measure, and FPR when using the LDNN model as an intrusion classifier. The results also showed that the CFLVAE-LDNN model outperformed the existing over-sampling, learning-based and state-of-the-art classifiers in terms of minority-class intrusion detection rates. This indicates that the data generated with the proposed model is closer to the original data and can better reproduce the probability distribution of its features.

Moreover, the proposed LDNN with different architectures, feature numbers, and network compression techniques is trained with the balanced dataset. The results demonstrated that the model is lightweight in terms of size, memory, and CPU time consumption and is suitable for resource-constrained IoT systems.

6.3 Contribution and Novelty

To bring up the rear, the contributions of this research are manifold:

- The research investigated the security threats, intrusions, and vulnerabilities of IoT.
- It studied present security solutions, including learning-based and encryption-based countermeasures for IoT systems.
- It presents an insightful comparative studies of the current intrusions and their countermeasures.
- This research designed and developed a novel data generation model called Class-wise Focal Loss (CFL) Variational AutoEncoder (CFLVAE) in order to solve the data imbalance issue. The CFL objective function focuses on the different minority

class samples differently and learns the best distribution of observed data, leading the CFLVAE to generate more realistic, diverse, and quality intrusion data.

- This research fine-tuned and optimized the Alpha (α) and Gamma (γ) parameters for the proposed CFL in our intrusion dataset.
- This thesis utilized the strength of Deep Neural Network (DNN) to learn the features of high-dimensional balanced intrusion data to achieve high attack detection performance and inferior FPR.
- This research utilized feature reduction and network compression techniques to reduce the complexity of the LDNN classifier to maintain lightweight criteria for resource-constrained IoT.
- Finally, this research evaluated our proposed CFLVAE-LDNN model on the NSL-KDD dataset, provided a detailed comparative analysis with relevant state-of-the-art intrusion detection techniques, and verified superior performance in detecting minority-class attacks in addition to overall detection accuracy.

6.4 Strengths and Limitations

Based on the results of this research, the strengths of the proposed CFLVAE-LDNN model are highlighted below:

1. The CFLVAE is able to produce realistic and diverse intrusion data samples
2. The CFLVAE-LDNN improves overall intrusion detection accuracy and minimizes the False Positive Rate.
3. The model improves minority-attacks detection rates.
4. The classifier is lightweight and suitable for resource-constrained IoT devices.
    a) The size of the proposed lightweight classifier is only 7 KB, which can perfectly be implemented in IoT devices.

b) The memory consumption of 1.5 MB and consumption of CPU time of 0.14 seconds ensure the model is suitable for resource-constrained IoT systems.

Based on the results of this research, the limitations of the proposed CFLVAE-LDNN model are highlighted below:

1. Including other IoT intrusion datasets to conduct extensive experiments might be interesting. It is difficult to find publicly available IoT intrusion datasets.
2. It is difficult to find publicly available IoT intrusion dataset.
3. The VAE model might not be as successful as other data generative models.
4. The findings outlined here prove that the model is suitable for IoT. However, it will be interesting to implement the proposed CFLVAE-LDNN in resource-constrained IoT.

6.5 Future Research Directions

The findings presented in this thesis are relevant to the deep learning and cyber security community. Hence, conducting further investigations on various IoT intrusion datasets will be worthwhile. NSL-KDD dataset has already been examined for imbalance learning and intrusion detection. Interestingly, it will be more advantageous to include various datasets with other properties of IoT and conventional Internet in the experiments. Based on the experimental results reported in this thesis, the CFLVAE-LDNN framework should be successful in intrusion detection with conflicting attack categories. There may be intrusion classification problems that do not have distinguished attack properties. For such cases, the CFLVAE-LDNN may not be as beneficial as other intrusion detection approaches.

The data generated using the cost-sensitive focal loss may improve intrusion detection. However, in the future, it will be interesting to investigate other objective functions and different data generation techniques to enhance data imbalance problems in intrusion

detection. The findings reported in this thesis ensure the suitability of the model for IoT. However, it will be interesting to implement the model in resource-constrained IoT devices.

6.6 Final Words

This thesis established and reported an intrusion detection framework called CFLVAE-LDNN. The CFLVAE-LDNN framework consists of CFLVAE data generation and LDNN intrusion detection phases. The CFLVAE data generation model combines VAE and Class-wise Focal Loss for better reconstructing the observed data samples to balance the intrusion dataset. The proposed CFLVAE model was proposed to generate diverse and realistic samples for the undetectable minority-class attacks. A Lightweight Deep Neural Network (LDNN) classification model was designed for intrusion detection; then trained and tested using generated balanced dataset. Additionally, to meet lightweight criteria for resource-constrained devices, a further improvement is proposed by utilizing the dimensionality reduction and network compression techniques for the LDNN classifier. Overall, it is interesting to present the proposed CFLVAE-LDNN framework to help the researchers and practitioners with intrusion detection in the IoT.

REFERENCES

Aazam, M., St-Hilaire, M., Lung, C.-H., & Lambadaris, I. (2016). Pre-fog: Iot trace based probabilistic resource estimation at fog. In *2016 13th ieee annual consumer communications & networking conference (ccnc)* (pp. 12–17).

Abbas, S., Merabti, M., Llewellyn-Jones, D., & Kifayat, K. (2012). Lightweight sybil attack detection in manets. *IEEE systems journal*, *7*(2), 236–248.

Abdulhammed, R., Faezipour, M., Abuzneid, A., & AbuMallouh, A. (2018). Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic. *IEEE sensors letters*, *3*(1), 1–4.

Abeshu, A., & Chilamkurti, N. (2018). Deep learning: The frontier for distributed attack detection in fog-to-things computing. *IEEE Communications Magazine*, *56*(2), 169–175.

Abualghanam, O., Qatawneh, M., & Almobaideen, W. (2019). A survey of key distribution in the context of internet of things. *Journal of Theoretical and Applied Information Technology*, *97*(22), 3217–3241.

Adnan, A., Muhammed, A., Abd Ghani, A. A., Abdullah, A., & Hakim, F. (2021). An intrusion detection system for the internet of things based on machine learning: Review and challenges. *Symmetry*, *13*(6), 1011.

Agyemang, J. O., Kponyo, J. J., & Acquah, I. (2019). Lightweight man-in-the-middle (mitm) detection and defense algorithm for wifi-enabled internet of things (iot) gateways. *Inf. Secur. Comput. Fraud*, *7*.

Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, *32*(1), e4150.

Ahmed, A., Omar, N. M., & Ibrahim, H. M. (2017). Modern iot architectures review: A security perspective. In *Proc. 8th annu. int. conf. ict: Big data, cloud secur.* (pp. 73–81).

Ahmed, N., & Khan, M. Z. R. (2021). A secure iot based grid-connected inverter using rsa

algorithm. In *2021 31st australasian universities power engineering conference (aupec)* (pp. 1–5).

Airehrour, D., Gutierrez, J., & Ray, S. K. (2016). Secure routing for internet of things: A survey. *Journal of Network and Computer Applications*, *66*, 198–213.

Alaba, F. A., Othman, M., Hashem, I. A. T., & Alotaibi, F. (2017). Internet of things security: A survey. *Journal of Network and Computer Applications*, *88*, 10–28.

Albahar, M. A., & Binsawad, M. (2020). Deep autoencoders and feedforward networks based on a new regularization for anomaly detection. *Security and Communication Networks*, *2020*.

Aldweesh, A., Derhab, A., & Emam, A. Z. (2020). Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowledge-Based Systems*, *189*, 105124.

Aleesa, A., Younis, M., Mohammed, A. A., & Sahar, N. (2021). Deep-intrusion detection system with enhanced unsw-nb15 dataset based on deep learning techniques. *Journal of Engineering Science and Technology*, *16*(1), 711–727.

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE communications surveys & tutorials*, *17*(4), 2347–2376.

Al-Garadi, M. A., Mohamed, A., Al-Ali, A. K., Du, X., Ali, I., & Guizani, M. (2020). A survey of machine and deep learning methods for internet of things (iot) security. *IEEE Communications Surveys & Tutorials*, *22*(3), 1646–1685.

Al Hayajneh, A., Bhuiyan, M. Z. A., & McAndrew, I. (2020). Improving internet of things (iot) security with software-defined networking (sdn). *Computers*, *9*(1), 8.

Ali, W., Din, I. U., Almogren, A., Guizani, M., & Zuair, M. (2020). A lightweight privacy-aware iot-based metering scheme for smart industrial ecosystems. *IEEE Transactions on Industrial Informatics*.

Aljohani, N. R., Fayoumi, A., & Hassan, S.-U. (2021). A novel focal-loss and class-weight-aware convolutional neural network for the classification of in-text citations. *Journal of Information Science*, 0165551521991022.

AlMajed, H., & AlMogren, A. (2020). A secure and efficient ecc-based scheme for edge computing and internet of things. *Sensors*, *20*(21), 6158.

Alnabulsi, H., Islam, R., & Talukder, M. (2018). Gmsa: Gathering multiple signatures approach to defend against code injection attacks. *IEEE Access*, *6*, 77829–77840.

Alonso, L., Milanés, V., Torre-Ferrero, C., Godoy, J., Oria, J. P., & De Pedro, T. (2011). Ultrasonic sensors in urban traffic driving-aid systems. *Sensors*, *11*(1), 661–673.

Alotaibi, A. (2020). Deep generative adversarial networks for image-to-image translation: A review. *Symmetry*, *12*(10), 1705.

Alrawashdeh, K., & Purdy, C. (2016). Toward an online anomaly intrusion detection system based on deep learning. In *2016 15th ieee international conference on machine learning and applications (icmla)* (pp. 195–200).

Al Salami, S., Baek, J., Salah, K., & Damiani, E. (2016). Lightweight encryption for smart home. In *2016 11th international conference on availability, reliability and security (ares)* (pp. 382–388).

Al-Shehari, T., & Alsowail, R. A. (2021). An insider data leakage detection using one-hot encoding, synthetic minority oversampling and machine learning techniques. *Entropy*, *23*(10), 1258.

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

Althubiti, S., Nick, W., Mason, J., Yuan, X., & Esterline, A. (2018). Applying long short-term memory recurrent neural network for intrusion detection. In *Southeastcon 2018* (pp. 1–5).

Aluvala, S., Sekhar, K. R., & Vodnala, D. (2016). An empirical study of routing attacks in mobile ad-hoc networks. *Procedia Computer Science*, *92*, 554–561.

Amaral, L. A., Hessel, F. P., Bezerra, E. A., Corrêa, J. C., Longhi, O. B., & Dias, T. F. (2011). ecloudrfid–a mobile software framework architecture for pervasive rfid-based applications. *Journal of Network and Computer Applications*, *34*(3), 972–979.

Aminanto, M. E., Choi, R., Tanuwidjaja, H. C., Yoo, P. D., & Kim, K. (2017). Deep abstraction and weighted feature selection for wi-fi impersonation detection. *IEEE Transactions on Information Forensics and Security*, *13*(3), 621–636.

Andrea, I., Chrysostomou, C., & Hadjichristofi, G. (2015). Internet of things: Security vulnerabilities and challenges. In *2015 ieee symposium on computers and communication (iscc)* (pp. 180–187).

Andrews, J. G. (2013). Seven ways that hetnets are a cellular paradigm shift. *IEEE communications magazine*, *51*(3), 136–144.

Ashraf, Q. M., & Habaebi, M. H. (2015). Autonomic schemes for threat mitigation in internet of things. *Journal of Network and Computer Applications*, *49*, 112–127.

Ashraf, Q. M., Habaebi, M. H., & Islam, M. R. (2016). Jammer localization using wireless devices with mitigation by self-configuration. *Plos one*, *11*(9), e0160311.

Ashraf, Q. M., Habaebi, M. H., Sinniah, G. R., & Chebil, J. (2014). Broadcast based registration technique for heterogenous nodes in the iot. In *International conference on control, engineering, and information technology (ceit 2014), sousse.*

Awalgaonkar, N. M., Zheng, H., & Gurciullo, C. S. (2020). Deeva: A deep learning and iot based computer vision system to address safety and security of production sites in energy industry. *arXiv preprint arXiv:2003.01196*.

Aziz, A., & Singh, K. (2019). Lightweight security scheme for internet of things. *Wireless Personal Communications*, *104*(2), 577–593.

Bairagi, A. K., Khondoker, R., & Islam, R. (2016). An efficient steganographic approach for protecting communication in the internet of things (iot) critical infrastructures. *Information Security Journal: A Global Perspective*, *25*(4-6), 197–212.

Baker, S. B., Xiang, W., & Atkinson, I. (2017). Internet of things for smart healthcare: Technologies, challenges, and opportunities. *IEEE Access*, *5*, 26521–26544.

Bala, T., & Kumar, Y. (2015). Asymmetric algorithms and symmetric algorithms: A review. *International Journal of Computer Applications (ICAET)*, 1–4.

Banupriya, S., Kottursamy, K., & Bashir, A. K. (2021). Privacy-preserving hierarchical deterministic key generation based on a lattice of rings in public blockchain. *Peer-to-Peer Networking and Applications*, 1–13.

Baskar, C., Balasubramaniyan, C., & Manivannan, D. (2016). Establishment of light weight cryptography for resource constraint environment using fpga. *Procedia Computer Science*, *78*, 165–171.

Battat, N., Seba, H., & Kheddouci, H. (2014). Monitoring in mobile ad hoc networks: A survey. *Computer Networks*, *69*, 82–100.

Beraha, M., Metelli, A. M., Papini, M., Tirinzoni, A., & Restelli, M. (2019). Feature selection via mutual information: new theoretical insights. In *2019 international joint conference on neural networks (ijcnn)* (pp. 1–9).

Bisong, E. (2019). *Building machine learning and deep learning models on google cloud platform*. Springer.

Boneh, D., & Franklin, M. (2001). Identity-based encryption from the weil pairing. In *Annual international cryptology conference* (pp. 213–229).

Borgia, E. (2014). The internet of things vision: Key features, applications and open issues. *Computer Communications*, *54*, 1–31.

Bostani, H., & Sheikhan, M. (2017). Hybrid of anomaly-based and specification-based ids for internet of things using unsupervised opf based on mapreduce approach. *Computer Communications*, *98*, 52–71.

Botta, A., De Donato, W., Persico, V., & Pescapé, A. (2016). Integration of cloud computing and internet of things: a survey. *Future generation computer systems*, *56*, 684–700.

Bowers, A. J., & Zhou, X. (2019). Receiver operating characteristic (roc) area under the curve (auc): A diagnostic measure for evaluating the accuracy of predictors of education outcomes. *Journal of Education for Students Placed at Risk (JESPAR)*, *24*(1), 20–46.

Briggs, B. (2019). *Hackers hit norsk hydro with ransomware*. Retrieved 2021-04-01, from `https://news.microsoft.com/transform/hackers-hit-norsk-hydro`

-ransomware-company-responded-transparency/

Bu, K., Xu, M., Liu, X., Luo, J., Zhang, S., & Weng, M. (2015). Deterministic detection of cloning attacks for anonymous rfid systems. *IEEE Transactions on Industrial Informatics*, *11*(6), 1255–1266.

Bukhari, S. T., & Mohy-ud Din, H. (2021). A systematic evaluation of learning rate policies in training cnns for brain tumor segmentation. *Physics in Medicine & Biology*, *66*(10), 105004.

Button, M., Wang, V., Klahr, R., Amili, S., & Shah, J. (2016). Cyber breaches survey 2016.

Bysani, L. K., & Turuk, A. K. (2011). A survey on selective forwarding attack in wireless sensor networks. In *2011 international conference on devices and communications (icdecom)* (pp. 1–5).

Cai, Y., Pelechrinis, K., Wang, X., Krishnamurthy, P., & Mo, Y. (2013). Joint reactive jammer detection and localization in an enterprise wifi network. *Computer Networks*, *57*(18), 3799–3811.

Campagnaro, F., Tronchin, D., Signori, A., Petroccia, R., Pelekanakis, K., Paglierani, P., ... Zorzi, M. (2020). Replay-attack countermeasures for underwater acoustic networks. In *Global oceans 2020: Singapore–us gulf coast* (pp. 1–9).

Canzanese, R., Kam, M., & Mancoridis, S. (2013). Toward an automatic, online behavioral malware classification system. In *2013 ieee 7th international conference on self-adaptive and self-organizing systems* (pp. 111–120).

Cassel, M., & Lima, F. (2006). Evaluating one-hot encoding finite state machines for seu reliability in sram-based fpgas. In *12th ieee international on-line testing symposium (iolts'06)* (pp. 6–pp).

Catarinucci, L., De Donno, D., Mainetti, L., Palano, L., Patrono, L., Stefanizzi, M. L., & Tarricone, L. (2015). An iot-aware architecture for smart healthcare systems. *IEEE internet of things journal*, *2*(6), 515–526.

Cervantes, C., Poplade, D., Nogueira, M., & Santos, A. (2015). Detection of sinkhole attacks for supporting secure routing on 6lowpan for internet of things. In *2015*

ifip/ieee international symposium on integrated network management (im) (pp. 606–611).

Chandra, S., Paira, S., Alam, S. S., & Sanyal, G. (2014). A comparative survey of symmetric and asymmetric key cryptography. In *2014 international conference on electronics, communication and computational engineering (icecce)* (pp. 83–93).

Chang, Y., Li, W., & Yang, Z. (2017). Network intrusion detection based on random forest and support vector machine. In *2017 ieee international conference on computational science and engineering (cse) and ieee international conference on embedded and ubiquitous computing (euc)* (Vol. 1, pp. 635–638).

Chaudhari, M. P., & Patel, S. R. (2014). A survey on cryptography algorithms. *International Journal of Advance Research in Computer Science and Management Studies*, *2*(3).

Chawla, N. V. (2009). Data mining for imbalanced datasets: An overview. *Data mining and knowledge discovery handbook*, 875–886.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321–357.

Chawla, N. V., Japkowicz, N., & Kotcz, A. (2004). Special issue on learning from imbalanced data sets. *ACM SIGKDD explorations newsletter*, *6*(1), 1–6.

Chen, S., Pang, Z., Wen, H., Yu, K., Zhang, T., & Lu, Y. (2020). Automated labeling and learning for physical layer authentication against clone node and sybil attacks in industrial wireless edge networks. *IEEE Transactions on Industrial Informatics*, *17*(3), 2041–2051.

Cheng, R., Wu, K., Su, Y., Li, W., Cui, W., & Tong, J. (2021). An efficient ecc-based cp-abe scheme for power iot. *Processes*, *9*(7), 1176.

Cheng, Z., & Chai, S. (2020). A cyber intrusion detection method based on focal loss neural network. In *2020 39th chinese control conference (ccc)* (pp. 7379–7383).

Choi, B. G., Cho, E. J., Kim, J. H., Hong, C. S., & Kim, J. H. (2009). A sinkhole attack detection mechanism for lqi based mesh routing in wsn. In *2009 international conference on information networking* (pp. 1–5).

Choi, Y., Choi, Y., Kim, D., & Park, J. (2017). Scheme to guarantee ip continuity for nfc-based iot networking. In *2017 19th international conference on advanced communication technology (icact)* (pp. 695–698).

Chuang, Y.-H., Lo, N.-W., Yang, C.-Y., & Tang, S.-W. (2018). A lightweight continuous authentication protocol for the internet of things. *Sensors*, *18*(4), 1104.

Chugh, K., Aboubaker, L., & Loo, J. (2012). Case study of a black hole attack on lowpan-rpl. In *Proc. of the sixth international conference on emerging security information, systems and technologies (securware), rome, italy (august 2012)* (pp. 157–162).

Chung, C.-C., Chen, W.-T., & Chang, Y.-C. (2020). Using quantization-aware training technique with post-training fine-tuning quantization to implement a mobilenet hardware accelerator. In *2020 indo–taiwan 2nd international conference on computing, analytics and networks (indo-taiwan ican)* (pp. 28–32).

Čolaković, A., & Hadžialić, M. (2018). Internet of things (iot): A review of enabling technologies, challenges, and open research issues. *Computer Networks*, *144*, 17–39.

Colding, J., & Barthel, S. (2017). An urban ecology critique on the "smart city" model. *Journal of Cleaner Production*, *164*, 95–101.

Conti, M., Dragoni, N., & Lesyk, V. (2016). A survey of man in the middle attacks. *IEEE Communications Surveys & Tutorials*, *18*(3), 2027–2051.

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. (2018). Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, *35*(1), 53–65.

Dai, H.-N., Wang, Q., Li, D., & Wong, R. C.-W. (2013). On eavesdropping attacks in wireless sensor networks with directional antennas. *International Journal of Distributed Sensor Networks*, *9*(8), 760834.

De Meulenaer, G., Gosset, F., Standaert, F.-X., & Pereira, O. (2008). On the energy cost of communication and cryptography in wireless sensor networks. In *2008 ieee international conference on wireless and mobile computing, networking and communications* (pp. 580–585).

Deogirikar, J., & Vidhate, A. (2017). Security attacks in iot: A survey. In *2017 international conference on i-smac (iot in social, mobile, analytics and cloud)(i-smac)* (pp. 32–37).

Dhanabal, L., & Shantharajah, S. (2015). A study on nsl-kdd dataset for intrusion detection system based on classification algorithms. *International journal of advanced research in computer and communication engineering*, *4*(6), 446–452.

Dhindsa, A., Bhatia, S., Agrawal, S., & Sohi, B. S. (2021). An improvised machine learning model based on mutual information feature selection approach for microbes classification. *Entropy*, *23*(2), 257.

Ding, H., Chen, L., Dong, L., Fu, Z., & Cui, X. (2022). Imbalanced data classification: A knn and generative adversarial networks-based hybrid approach for intrusion detection. *Future Generation Computer Systems*, *131*, 240–254.

Doersch, C. (2016). Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*.

Domingo, M. C. (2012). An overview of the internet of things for people with disabilities. *journal of Network and Computer Applications*, *35*(2), 584–596.

Dong, G., Liao, G., Liu, H., & Kuang, G. (2018). A review of the autoencoder and its variants: A comparative perspective from target recognition in synthetic-aperture radar images. *IEEE Geoscience and Remote Sensing Magazine*, *6*(3), 44–68.

Donta, P. K., Srirama, S. N., Amgoth, T., & Annavarapu, C. S. R. (2021). Survey on recent advances in iot application layer protocols and machine learning scope for research directions. *Digital Communications and Networks*.

Dragomir, D., Gheorghe, L., Costea, S., & Radovici, A. (2016). A survey on secure communication protocols for iot systems. In *2016 international workshop on secure internet of things (siot)* (pp. 47–62).

Du, W., Deng, J., Han, Y. S., & Varshney, P. K. (2006). A key predistribution scheme for sensor networks using deployment knowledge. *IEEE Transactions on dependable and secure computing*, *3*(1), 62–77.

Du, W., Wang, R., & Ning, P. (2005). An efficient scheme for authenticating public keys in sensor networks. In *Proceedings of the 6th acm international symposium on*

mobile ad hoc networking and computing (pp. 58–67).

Dua, M., et al. (2019). Machine learning approach to ids: A comprehensive review. In *2019 3rd international conference on electronics, communication and aerospace technology (iceca)* (pp. 117–121).

Dubey, A., Cammarota, R., & Aysu, A. (2020). Bomanet: Boolean masking of an entire neural network. In *2020 ieee/acm international conference on computer aided design (iccad)* (pp. 1–9).

Eisenbarth, T., Kumar, S., Paar, C., Poschmann, A., & Uhsadel, L. (2007). A survey of lightweight-cryptography implementations. *IEEE Design & Test of Computers*, *24*(6), 522–533.

Elhadi, S., Marzak, A., Sael, N., & Merzouk, S. (2018). Comparative study of iot protocols. *Smart Application and Data Analysis for Smart Cities (SADASC'18)*.

Elngar, A. A. (2018). Iot-based efficient tamper detection mechanism for healthcare application. *Int. J. Netw. Secur.*, *20*(3), 489–495.

Elngar, A. A., & Bhatt, S. (2018). Iot-based efficient tamper detection mechanism for healthcare application. *Int. J. Netw. Secur.*, *20*(3), 489–495.

Elsaeidy, A. A., Jagannath, N., Sanchis, A. G., Jamalipour, A., & Munasinghe, K. S. (2020). Replay attack detection in smart cities using deep learning. *IEEE Access*, *8*, 137825–137837.

Erdin, E., Zachor, C., & Gunes, M. H. (2015). How to find hidden users: A survey of attacks on anonymity networks. *IEEE Communications Surveys & Tutorials*, *17*(4), 2296–2316.

Erpek, T., Sagduyu, Y. E., & Shi, Y. (2018). Deep learning for launching and mitigating wireless jamming attacks. *IEEE Transactions on Cognitive Communications and Networking*, *5*(1), 2–14.

Fang, W., Tan, X., & Wilbur, D. (2020). Application of intrusion detection technology in network safety based on machine learning. *Safety Science*, *124*, 104604.

Feng, D., Jiang, C., Lim, G., Cimini, L. J., Feng, G., & Li, G. Y. (2012). A survey of energy-efficient wireless communications. *IEEE Communications Surveys & Tutorials*, *15*(1), 167–178.

Feng, Q., He, D., Zeadally, S., & Wang, H. (2018). Anonymous biometrics-based authentication scheme with key distribution for mobile multi-server environment. *Future Generation Computer Systems*, *84*, 239–251.

Ferdowsi, A., & Saad, W. (2018a). Deep learning-based dynamic watermarking for secure signal authentication in the internet of things. In *2018 ieee international conference on communications (icc)* (pp. 1–6).

Ferdowsi, A., & Saad, W. (2018b). Deep learning for signal authentication and security in massive internet-of-things systems. *IEEE Transactions on Communications*, *67*(2), 1371–1387.

Fernando, K. R. M., & Tsokos, C. P. (2021). Dynamically weighted balanced loss: class imbalanced learning and confidence calibration of deep neural networks. *IEEE Transactions on Neural Networks and Learning Systems*.

Fiesler, E., Choudry, A., & Caulfield, H. J. (1990). Weight discretization paradigm for optical neural networks. In *Optical interconnections and networks* (Vol. 1281, pp. 164–173).

Finnerty, K., Fullick, S., Motha, H., Shah, J. N., Button, M., & Wang, V. (2019). Cyber security breaches survey 2019.

Finnerty, K., Motha, H., Shah, J., White, Y., Button, M., & Wang, V. (2018). Cyber security breaches survey 2018: Statistical release.

Forsberg, D., Ohba, Y., Patil, B., Tschofenig, H., & Yegin, A. (2008). *Protocol for carrying authentication for network access (pana)* (Tech. Rep.). RFC 5191, May.

Frangopoulos, E. D., Eloff, M. M., & Venter, L. M. (2013). Psychosocial risks: Can their effects on the security of information systems really be ignored? *Information Management & Computer Security*.

Ganta, S. R., Kasiviswanathan, S. P., & Smith, A. (2008). Composition attacks and auxiliary information in data privacy. In *Proceedings of the 14th acm sigkdd*

international conference on knowledge discovery and data mining (pp. 265–273).

Gao, J., Liu, K., Wang, B., Wang, D., & Hong, Q. (2021). An improved deep forest for alleviating the data imbalance problem. *Soft Computing*, *25*(3), 2085–2101.

Gao, N., Gao, L., Gao, Q., & Wang, H. (2014). An intrusion detection model based on deep belief networks. In *2014 second international conference on advanced cloud and big data* (pp. 247–252).

Gao, S., Huang, F., Cai, W., & Huang, H. (2021). Network pruning via performance maximization. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 9270–9280).

Garms, L., & Lehmann, A. (2019). Group signatures with selective linkability. In *Iacr international workshop on public key cryptography* (pp. 190–220).

Gaubatz, G., Kaps, J.-P., Ozturk, E., & Sunar, B. (2005). State of the art in ultra-low power public key cryptography for wireless sensor networks. In *Third ieee international conference on pervasive computing and communications workshops* (pp. 146–150).

George, R., & Roy, B. (2022). Handling class imbalance in fraud detection using machine learning techniques. In *Icdsmla 2020* (pp. 803–813). Springer.

George, T. K., Jacob, K. P., & James, R. K. (2018). Token based detection and neural network based reconstruction framework against code injection vulnerabilities. *Journal of Information Security and Applications*, *41*, 75–91.

George, T. K., Jacob, K. P., & James, R. K. (2019). A proposed framework against code injection vulnerabilities in online applications. *Journal of Internet Technology*, *20*(1), 83–96.

Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M. W., & Keutzer, K. (2021). A survey of quantization methods for efficient neural network inference. *arXiv preprint arXiv:2103.13630*.

Ghosal, A., Halder, S., & DasBit, S. (2012). A dynamic tdma based scheme for securing query processing in wsn. *Wireless Networks*, *18*(2), 165–184.

Girs, S., Sentilles, S., Asadollah, S. A., Ashjaei, M., & Mubeen, S. (2020). A systematic literature study on definition and modeling of service-level agreements for cloud services in iot. *IEEE Access*, *8*, 134498–134513.

Gnad, D. R., Krautter, J., & Tahoori, M. B. (2019). Leaky noise: New side-channel attack vectors in mixed-signal iot devices. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 305–339.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, *27*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139–144.

Grover, J., Laxmi, V., & Gaur, M. S. (2013). Attack models and infrastructure supported detection mechanisms for position forging attacks in vehicular ad hoc networks. *CSI transactions on ICT*, *1*(3), 261–279.

Grover, K., & Lim, A. (2015). A survey of broadcast authentication schemes for wireless networks. *Ad Hoc Networks*, *24*, 288–316.

Gu, K., Zhang, W., Lim, S.-J., Sharma, P. K., Al-Makhadmeh, Z., & Tolba, A. (2020). Reusable mesh signature scheme for protecting identity privacy of iot devices. *Sensors*, *20*(3), 758.

Gulzar, M., & Abbas, G. (2019). Internet of things security: a survey and taxonomy. In *2019 international conference on engineering and emerging technologies (iceet)* (pp. 1–6).

Gupta, H., Srikant, R., & Ying, L. (2019). Finite-time performance bounds and adaptive learning rate selection for two time-scale reinforcement learning. *Advances in Neural Information Processing Systems*, *32*.

Gyamfi, E., & Jurcut, A. (2022). Intrusion detection in internet of things systems: A review on design approaches leveraging multi-access edge computing, machine learning, and datasets. *Sensors*, *22*(10), 3744.

Gyamfi, N. K., & Owusu, E. (2018). Survey of mobile malware analysis, detection techniques and tool. In *2018 ieee 9th annual information technology, electronics and mobile communication conference (iemcon)* (pp. 1101–1107).

Habib, M. A., Ahmad, M., Jabbar, S., Ahmed, S. H., & Rodrigues, J. J. (2018). Speeding up the internet of things: Leaiot: A lightweight encryption algorithm toward low-latency communication for the internet of things. *IEEE Consumer Electronics Magazine*, *7*(6), 31–37.

HaddadPajouh, H., Dehghantanha, A., Parizi, R. M., Aledhari, M., & Karimipour, H. (2021). A survey on internet of things security: Requirements, challenges, and solutions. *Internet of Things*, *14*, 100129.

Hague, T. (2018). *Top dutch banks, revenue service hit by cyber attacks.* Retrieved 2021-04-01, from `https://www.securityweek.com/top-dutch-banks-hit-cyber-attacks`

Hahm, O., Baccelli, E., Petersen, H., & Tsiftes, N. (2015). Operating systems for low-end devices in the internet of things: a survey. *IEEE Internet of Things Journal*, *3*(5), 720–734.

Ham, H.-S., Kim, H.-H., Kim, M.-S., & Choi, M.-J. (2014). Linear svm-based android malware detection for reliable iot services. *Journal of Applied Mathematics*, *2014*.

Hamad, R. A., Kimura, M., & Lundström, J. (2020). Efficacy of imbalanced data handling methods on deep learning for smart homes environments. *SN Computer Science*, *1*(4), 1–10.

Harrington, P. (2012). *Machine learning in action*. Simon and Schuster.

Hasan, M., Islam, M. M., Zarif, M. I. I., & Hashem, M. (2019). Attack and anomaly detection in iot sensors in iot sites using machine learning approaches. *Internet of Things*, *7*, 100059.

Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P., & Sikdar, B. (2019). A survey on iot security: application areas, security threats, and solution architectures. *IEEE Access*, *7*, 82721–82743.

Hayaty, M., Muthmainah, S., & Ghufran, S. M. (2020). Random and synthetic over-

sampling approach to resolve data imbalance in classification. *International Journal of Artificial Intelligence Research*, *4*(2), 86–94.

He, D., Chen, C., Chan, S., Bu, J., & Vasilakos, A. V. (2012). Retrust: Attack-resistant and lightweight trust management for medical sensor networks. *IEEE transactions on information technology in biomedicine*, *16*(4), 623–632.

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 ieee international joint conference on neural networks (ieee world congress on computational intelligence)* (pp. 1322–1328).

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, *21*(9), 1263–1284.

He, H., & Ma, Y. (2013). Imbalanced learning: foundations, algorithms, and applications. *Wiley-IEEE Press*.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hoffstein, J., Howgrave-Graham, N., Pipher, J., & Whyte, W. (2009). Practical lattice-based cryptography: Ntruencrypt and ntrusign. In *The lll algorithm* (pp. 349–390). Springer.

Holzer, R., & de Meer, H. (2011). Methods for approximations of quantitative measures in self-organizing systems. In *International workshop on self-organizing systems* (pp. 1–15).

Horrow, S., & Sardana, A. (2012). Identity management framework for cloud based internet of things. In *Proceedings of the first international conference on security of internet of things* (pp. 200–203).

Hossain, M. M., Fotouhi, M., & Hasan, R. (2015). Towards an analysis of security issues, challenges, and open problems in the internet of things. In *2015 ieee world congress on services* (pp. 21–28).

Huang, X., Craig, P., Lin, H., & Yan, Z. (2016). Seciot: a security framework for the internet of things. *Security and communication networks*, *9*(16), 3083–3094.

Hussain, F., Hussain, R., Hassan, S. A., & Hossain, E. (2020). Machine learning in iot security: Current solutions and future challenges. *IEEE Communications Surveys & Tutorials*, *22*(3), 1686–1721.

Hussein, A. A., Leow, C. Y., & Rahman, T. A. (2017). Robust multiple frequency multiple power localization schemes in the presence of multiple jamming attacks. *PloS one*, *12*(5), e0177326.

Hussen, H. R., Tizazu, G. A., Ting, M., Lee, T., Choi, Y., & Kim, K.-H. (2013). Sakes: Secure authentication and key establishment scheme for m2m communication in the ip-based wireless sensor network (6l0wpan). In *2013 fifth international conference on ubiquitous and future networks (icufn)* (pp. 246–251).

Ito, T., Ohta, H., Matsuda, N., & Yoneda, T. (2007). A key predistribution scheme for deployable sensor networks using the node deployment probability density function. *Electronics and Communications in Japan (Part II: Electronics)*, *90*(10), 73–83.

Jahan, S., Riley, I., Walter, C., Gamble, R. F., Pasco, M., McKinley, P. K., & Cheng, B. H. (2020). Mape-k/mape-sac: An interaction framework for adaptive systems with security assurance cases. *Future Generation Computer Systems*, *109*, 197–209.

Janarthanan, T., & Zargari, S. (2017). Feature selection in unsw-nb15 and kddcup'99 datasets. In *2017 ieee 26th international symposium on industrial electronics (isie)* (pp. 1881–1886).

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent data analysis*, *6*(5), 429–449.

Jara, A. J., Ladid, L., & Gómez-Skarmeta, A. F. (2013). The internet of everything through ipv6: An analysis of challenges, solutions and opportunities. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, *4*(3), 97–118.

Jayasinghe, S., Rambukkanage, L., Silva, A., de Silva, N., & Perera, A. S. (2021). Critical sentence identification in legal cases using multi-class classification. In *2021 ieee 16th international conference on industrial and information systems (iciis)* (pp. 146–151).

Jeba, A., Paramasivan, B., & Usha, D. (2011). Security threats and its countermeasures in wireless sensor networks: An overview. *International Journal of Computer Applications*, *29*(6), 15–22.

Jiang, K., Wang, W., Wang, A., & Wu, H. (2020). Network intrusion detection combined hybrid sampling with deep hierarchical network. *IEEE Access*, *8*, 32464–32476.

Jianhong, H. (2015). Network intrusion detection algorithm based on improved support vector machine. In *2015 international conference on intelligent transportation, big data and smart city* (pp. 523–526).

Jiao, Z., Zhang, B., Gong, W., & Mouftah, H. (2015). A virtual queue-based back-pressure scheduling algorithm for wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*, *2015*(1), 1–9.

Jing, Q., Vasilakos, A. V., Wan, J., Lu, J., & Qiu, D. (2014). Security of the internet of things: perspectives and challenges. *Wireless Networks*, *20*(8), 2481–2501.

Johns, E. (2020). Cyber security breaches survey 2020. *London: Department for Digital, Culture, Media & Sport*.

Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, *6*(1), 1–54.

Kalloniatis, C., Kavakli, E., & Gritzalis, S. (2008). Addressing privacy requirements in system design: the pris method. *Requirements Engineering*, *13*(3), 241–255.

Kanda, M., & Chasko, S. (2012). Pana applicability in constrained environments. In *Smart object security wksp.*

Kang, J. J., Fahd, K., Venkatraman, S., Trujillo-Rasua, R., & Haskell-Dowland, P. (2019). Hybrid routing for man-in-the-middle (mitm) attack detection in iot networks. In *2019 29th international telecommunication networks and applications conference (itnac)* (pp. 1–6).

Kang, T., Li, X., Yu, C., & Kim, J. (2013). A survey of security mechanisms with direct sequence spread spectrum signals. *Journal of Computing Science and Engineering*, *7*(3), 187–197.

Karim, R., Rumi, L. S., Islam, M. A., Kobita, A. A., Tabassum, T., & Hossen, M. S. (2021). Digital signature authentication for a bank using asymmetric key cryptography algorithm and token based encryption. In *Evolutionary computing and mobile sustainable networks* (pp. 853–859). Springer.

Kaur, G., Rani, M. S., & Aseri, T. C. (2015). Improved aodv routing protocol for mitigating effects of grayhole attack in vanet using genetic algorithm. *Int. J. Comput. Sci. Eng. Technol.*, *5*(7).

Kaur, P., & Sharma, S. (2015). Spyware detection in android using hybridization of description analysis, permission mapping and interface analysis. *Procedia Computer Science*, *46*, 794–803.

Kavin, B. P., & Ganapathy, S. (2020). Ec (dh) 2: an effective secured data storage mechanism for cloud based iot applications using elliptic curve and diffie-hellman. *International Journal of Internet Technology and Secured Transactions*, *10*(5), 601–617.

Kavitha, R., & Caroline, B. E. (2015). Hybrid cryptographic technique for heterogeneous wireless sensor networks. In *2015 international conference on communications and signal processing (iccsp)* (pp. 1016–1020).

Khan, M. A., Quasim, M. T., Alghamdi, N. S., & Khan, M. Y. (2020). A secure framework for authentication and encryption using improved ecc for iot-based medical sensor data. *IEEE Access*, *8*, 52018–52027.

Khan, R., Khan, S. U., Zaheer, R., & Khan, S. (2012). Future internet: the internet of things architecture, possible applications and key challenges. In *2012 10th international conference on frontiers of information technology* (pp. 257–260).

Khanam, S., Ahmedy, I. B., Idris, M. Y. I., Jaward, M. H., & Sabri, A. Q. B. M. (2020). A survey of security challenges, attacks taxonomy and advanced countermeasures in the internet of things. *IEEE Access*, *8*, 219709–219743.

Khanam, S., Saleem, H. Y., & Pathan, A.-S. K. (2012). An efficient detection model of selective forwarding attacks in wireless mesh networks. In *International conference on internet and distributed computing systems* (pp. 1–14).

Khushi, M., Shaukat, K., Alam, T. M., Hameed, I. A., Uddin, S., Luo, S., ... Reyes, M. C. (2021). A comparative performance analysis of data resampling methods on imbalance medical data. *IEEE Access*, *9*, 109960–109975.

Kim, A., Oh, J., Ryu, J., & Lee, K. (2020). A review of insider threat detection approaches with iot perspective. *IEEE Access*, *8*, 78847–78867.

Kim, H., Park, J., Lee, C., & Kim, J.-J. (2021). Improving accuracy of binary neural networks using unbalanced activation distribution. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 7862–7871).

Kim, T. (2021). Generalizing mlps with dropouts, batch normalization, and skip connections. *arXiv preprint arXiv:2108.08186*.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P., Mohamed, S., Rezende, D. J., & Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in neural information processing systems* (pp. 3581–3589).

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Klahr, R., et al. (2017). Cyber security breaches survey. *A Survey Detailing Business Action or Cyber Security and the Costs and Impacts of Cyber Breaches and Attacks. United Kingdom: Department for Culture, Media; Sport*, *2017*.

Koh, J. Y., Ming, J. T. C., & Niyato, D. (2013). Rate limiting client puzzle schemes for denial-of-service mitigation. In *2013 ieee wireless communications and networking conference (wcnc)* (pp. 1848–1853).

Koh, J. Y., Nevat, I., Leong, D., & Wong, W.-C. (2016). Geo-spatial location spoofing detection for internet of things. *IEEE Internet of Things Journal*, *3*(6), 971–978.

Koidl, K. (2013). Loss functions in classification tasks. *School of Computer Science and Statistic Trinity College, Dublin*.

Komninos, N., Philippou, E., & Pitsillides, A. (2014). Survey in smart grid and smart home security: Issues, challenges and countermeasures. *IEEE Communications Surveys & Tutorials*, *16*(4), 1933–1954.

Kortesniemi, Y., Lagutin, D., Elo, T., & Fotiou, N. (2019). Improving the privacy of iot with decentralised identifiers (dids). *Journal of Computer Networks and Communications*, *2019*.

Kotsev, A., Schade, S., Craglia, M., Gerboles, M., Spinelle, L., & Signorini, M. (2016). Next generation air quality platform: Openness and interoperability for the internet of things. *Sensors*, *16*(3), 403.

Kotsiantis, S., Kanellopoulos, D., Pintelas, P., et al. (2006). Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, *30*(1), 25–36.

Kugler, P., Nordhus, P., & Eskofier, B. (2013). Shimmer, cooja and contiki: A new toolset for the simulation of on-node signal processing algorithms. In *2013 ieee international conference on body sensor networks* (pp. 1–6).

Lai, B., Kim, S., & Verbauwhede, I. (2002). Scalable session key construction protocol for wireless sensor networks. In *Ieee workshop on large scale realtime and embedded systems (lartes)* (Vol. 7).

La Manna, M., Perazzo, P., & Dini, G. (2021). Sea-brew: A scalable attribute-based encryption revocable scheme for low-bitrate iot wireless networks. *Journal of Information Security and Applications*, *58*, 102692.

Lansky, J., Ali, S., Mohammadi, M., Majeed, M. K., Karim, S. H. T., Rashidi, S., ... Rahmani, A. M. (2021). Deep learning-based intrusion detection systems: a systematic review. *IEEE Access*, *9*, 101574–101599.

Laurie, A. (2007). Practical attacks against rfid. *Network Security*, *2007*(9), 4–7.

Le, A., Loo, J., Chai, K. K., & Aiash, M. (2016). A specification-based ids for detecting attacks on rpl-based network topology. *Information*, *7*(2), 25.

Lee, J., & Park, K. (2021). Gan-based imbalanced data intrusion detection system. *Personal and Ubiquitous Computing*, *25*(1), 121–128.

Lee, Y., Lee, W., Shin, G., & Kim, K. (2017). Assessing the impact of dos attacks on iot gateway. In *Advanced multimedia and ubiquitous engineering* (pp. 252–257). Springer.

Leevy, J. L., Khoshgoftaar, T. M., & Peterson, J. M. (2021). Mitigating class imbalance for iot network intrusion detection: a survey. In *2021 ieee seventh international conference on big data computing service and applications (bigdataservice)* (pp.

143–148).

Lei, M., Li, X., Cai, B., Li, Y., Liu, L., & Kong, W. (2020). P-dnn: An effective intrusion detection method based on pruning deep neural network. In *2020 international joint conference on neural networks (ijcnn)* (pp. 1–9).

Li, F., Han, Y., & Jin, C. (2016). Practical access control for sensor networks in the context of the internet of things. *Computer Communications*, *89*, 154–164.

Li, J., Zhang, Y., Ning, J., Huang, X., Poh, G. S., & Wang, D. (2020). Attribute based encryption with privacy protection and accountability for cloudiot. *IEEE Transactions on Cloud Computing*.

Li, L., Wang, Z., & Li, N. (2020). Efficient attribute-based encryption outsourcing scheme with user and attribute revocation for fog-enabled iot. *IEEE Access*, *8*, 176738–176749.

Li, S., Tryfonas, T., & Li, H. (2016). The internet of things: a security point of view. *Internet Research*.

Li, S., Zhang, B., Fei, P., Shakeel, P. M., & Samuel, R. D. J. (2020). Computational efficient wearable sensor network health monitoring system for sports athletics using iot. *Aggression and Violent Behavior*, 101541.

Li, X., Yu, L., Chang, D., Ma, Z., & Cao, J. (2019). Dual cross-entropy loss for small-sample fine-grained vehicle classification. *IEEE Transactions on Vehicular Technology*, *68*(5), 4204–4212.

Li, Z., Rios, A. L. G., Xu, G., & Trajković, L. (2019). Machine learning techniques for classifying network anomalies and intrusions. In *2019 ieee international symposium on circuits and systems (iscas)* (pp. 1–5).

Liang, T., Glossner, J., Wang, L., Shi, S., & Zhang, X. (2021). Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, *461*, 370–403.

Liao, B., Ali, Y., Nazir, S., He, L., & Khan, H. U. (2020). Security analysis of iot devices by using mobile computing: a systematic literature review. *IEEE Access*, *8*, 120331–120350.

Lin, L., Yang. (2018). *Transmission apparatus and transmission method thereof.* Retrieved from https://patents.google.com/patent/US9954982B2/

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the ieee international conference on computer vision* (pp. 2980–2988).

Lin, X. (2013). Lsr: Mitigating zero-day sybil vulnerability in privacy-preserving vehicular peer-to-peer networks. *IEEE Journal on Selected Areas in Communications*, *31*(9), 237–246.

Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. *applied sciences*, *9*(20), 4396.

Liu, J., Kantarci, B., & Adams, C. (2020). Machine learning-driven intrusion detection for contiki-ng-based iot networks exposed to nsl-kdd dataset. In *Proceedings of the 2nd acm workshop on wireless security and machine learning* (pp. 25–30).

Liu, W., Keranidis, S., Mehari, M., Vanhie-Van Gerwen, J., Bouckaert, S., Yaron, O., & Moerman, I. (2013). Various detection techniques and platforms for monitoring interference condition in a wireless testbed. In *Measurement methodology and tools* (pp. 43–60). Springer.

Liu, X.-Y., Wu, J., & Zhou, Z.-H. (2008). Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *39*(2), 539–550.

Liu, Y., Ma, M., Liu, X., Xiong, N. N., Liu, A., & Zhu, Y. (2018). Design and analysis of probing route to defense sink-hole attacks for internet of things security. *IEEE Transactions on Network Science and Engineering*, *7*(1), 356–372.

Liu, Z., Liu, H., Xu, W., & Chen, Y. (2010). Wireless jamming localization by exploiting nodes' hearing ranges. In *International conference on distributed computing in sensor systems* (pp. 348–361).

Liu, Z., Sun, M., Zhou, T., Huang, G., & Darrell, T. (2018). Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*.

Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., & Lloret, J. (2017). Conditional

variational autoencoder for prediction and feature recovery applied to intrusion detection in iot. *Sensors*, *17*(9), 1967.

Lu, T., Yao, P., Zhao, L., Li, Y., Xie, F., & Xia, Y. (2015). Towards attacks and defenses of anonymous communication systems. *International Journal of Security and Its Applications*, *9*(1), 313–328.

Ma, T., Wang, F., Cheng, J., Yu, Y., & Chen, X. (2016). A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks. *Sensors*, *16*(10), 1701.

Ma, Y., Rao, J., Hu, W., Meng, X., Han, X., Zhang, Y., . . . Liu, C. (2012). An efficient index for massive iot data in cloud environment. In *Proceedings of the 21st acm international conference on information and knowledge management* (pp. 2129–2133).

Mahalle, P. N., Prasad, N. R., & Prasad, R. (2014). Threshold cryptography-based group authentication (tcga) scheme for the internet of things (iot). In *2014 4th international conference on wireless communications, vehicular technology, information theory and aerospace & electronic systems (vitae)* (pp. 1–5).

Mahmoud, R., Yousuf, T., Aloul, F., & Zualkernan, I. (2015). Internet of things (iot) security: Current status, challenges and prospective measures. In *2015 10th international conference for internet technology and secured transactions (icitst)* (pp. 336–341).

Manikandan, G., & Abirami, S. (2018). A survey on feature selection and extraction techniques for high-dimensional microarray datasets. In *Knowledge computing and its applications* (pp. 311–333). Springer.

Manzo, M., Roosta, T., & Sastry, S. (2005). Time synchronization attacks in sensor networks. In *Proceedings of the 3rd acm workshop on security of ad hoc and sensor networks* (pp. 107–116).

Marigowda, C., Thriveni, J., Gowrishankar, S., & Venugopal, K. (2018). An efficient secure algorithms to mitigate dos, replay and jamming attacks in wireless sensor network. In *Proceedings of the world congress on engineering and computer science* (Vol. 1).

Marjani, M., Nasaruddin, F., Gani, A., Karim, A., Hashem, I. A. T., Siddiqa, A., & Yaqoob,

I. (2017). Big iot data analytics: architecture, opportunities, and open research challenges. *ieee access*, *5*, 5247–5261.

Mathur, A., Newe, T., & Rao, M. (2016). Defence against black hole and selective forwarding attacks for medical wsns in the iot. *Sensors*, *16*(1), 118.

Mehmood, A., Mukherjee, M., Ahmed, S. H., Song, H., & Malik, K. M. (2018). Nbc-maids: Naïve bayesian classification technique in multi-agent system-enriched ids for securing iot against ddos attacks. *The Journal of Supercomputing*, *74*(10), 5156–5170.

Meshram, C., Li, C.-T., & Meshram, S. G. (2019). An efficient online/offline id-based short signature procedure using extended chaotic maps. *Soft Computing*, *23*(3), 747–753.

Miller, A., Horne, R., & Potter, C. (2015). Information security breaches survey 2015. *Inf. Secur. Breach Survey*.

Mishra, A. K., Tripathy, A. K., Puthal, D., & Yang, L. T. (2018). Analytical model for sybil attack phases in internet of things. *IEEE Internet of Things Journal*, *6*(1), 379–387.

Mitrokotsa, A., Rieback, M. R., & Tanenbaum, A. S. (2010a). Classification of rfid attacks. *Gen*, *15693*(14443), 14.

Mitrokotsa, A., Rieback, M. R., & Tanenbaum, A. S. (2010b). Classifying rfid attacks and defenses. *Information Systems Frontiers*, *12*(5), 491–505.

Molchanov, P., Mallya, A., Tyree, S., Frosio, I., & Kautz, J. (2019). Importance estimation for neural network pruning. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 11264–11272).

Morales-Molina, C. D., Hernandez-Suarez, A., Sanchez-Perez, G., Toscano-Medina, L. K., Perez-Meana, H., Olivares-Mercado, J., . . . Garcia-Villalba, L. J. (2021). A dense neural network approach for detecting clone id attacks on the rpl protocol of the iot. *Sensors*, *21*(9), 3173.

More, A. (2016). Survey of resampling techniques for improving classification performance in unbalanced datasets. *arXiv preprint arXiv:1608.06048*.

Moustafa, N., & Slay, J. (2015). Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *2015 military communications and information systems conference (milcis)* (pp. 1–6).

Muhammad, F., Anjum, W., & Mazhar, K. S. (2015). A critical analysis on the security concerns of internet of things (iot). *International Journal of Computer Applications*, *111*(7), 1–6.

Muhammad, K., Hamza, R., Ahmad, J., Lloret, J., Wang, H., & Baik, S. W. (2018a). Secure surveillance framework for iot systems using probabilistic image encryption. *IEEE Transactions on Industrial Informatics*, *14*(8), 3679–3689.

Muhammad, K., Hamza, R., Ahmad, J., Lloret, J., Wang, H., & Baik, S. W. (2018b). Secure surveillance framework for iot systems using probabilistic image encryption. *IEEE Transactions on Industrial Informatics*, *14*(8), 3679–3689.

Mulyanto, M., Faisal, M., Prakosa, S. W., & Leu, J.-S. (2021). Effectiveness of focal loss for minority classification in network intrusion detection systems. *Symmetry*, *13*(1), 4.

Mushtaq, M. F., Jamel, S., Disina, A. H., Pindar, Z. A., Shakir, N. S. A., & Deris, M. M. (2017). A survey on the cryptographic encryption algorithms. *International Journal of Advanced Computer Science and Applications*, *8*(11), 333–344.

Nandy, T., Idris, M. Y. I. B., Noor, R. M., Kiah, L. M., Lun, L. S., Juma'at, N. B. A., ... Bhattacharyya, S. (2019). Review on security of internet of things authentication mechanism. *IEEE Access*, *7*, 151054–151089.

Napiah, M. N., Idris, M. Y. I. B., Ramli, R., & Ahmedy, I. (2018). Compression header analyzer intrusion detection system (cha-ids) for 6lowpan communication protocol. *IEEE Access*, *6*, 16623–16638.

Napierala, K., & Stefanowski, J. (2016). Types of minority class examples and their influence on learning classifiers from imbalanced data. *Journal of Intelligent Information Systems*, *46*(3), 563–597.

Narkhede, S. (2018). Understanding auc-roc curve. *Towards Data Science*, *26*(1), 220–227.

Naru, E. R., Saini, H., & Sharma, M. (2017). A recent review on lightweight cryptography in iot. In *2017 international conference on i-smac (iot in social, mobile, analytics and cloud)(i-smac)* (pp. 887–890).

Nasiri, S., Sadoughi, F., Tadayon, M. H., & Dehnad, A. (2019). Security requirements of internet of things-based healthcare system: a survey study. *Acta Informatica Medica*, *27*(4), 253.

Nauman, A., Jamshed, M. A., Ahmad, Y., Ali, R., Zikria, Y. B., & Kim, S. W. (2019). An intelligent deterministic d2d communication in narrow-band internet of things. In *2019 15th international wireless communications & mobile computing conference (iwcmc)* (pp. 2111–2115).

Nguyen, H. M., Cooper, E. W., & Kamei, K. (2012). A comparative study on sampling techniques for handling class imbalance in streaming data. In *The 6th international conference on soft computing and intelligent systems, and the 13th international symposium on advanced intelligence systems* (pp. 1762–1767).

Nguyen, K. T., Laurent, M., & Oualha, N. (2015). Survey on secure communication protocols for the internet of things. *Ad Hoc Networks*, *32*, 17–31.

Ojo, M. O., Giordano, S., Procissi, G., & Seitanidis, I. N. (2018). A review of low-end, middle-end, and high-end iot devices. *IEEE Access*, *6*, 70528–70554.

Osada, G., Omote, K., & Nishide, T. (2017). Network intrusion detection based on semi-supervised variational auto-encoder. In *European symposium on research in computer security* (pp. 344–361).

Oualha, N., & Nguyen, K. T. (2016). Lightweight attribute-based encryption for the internet of things. In *2016 25th international conference on computer communication and networks (icccn)* (pp. 1–6).

Pajouh, H. H., Javidan, R., Khayami, R., Dehghantanha, A., & Choo, K.-K. R. (2016). A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in iot backbone networks. *IEEE Transactions on Emerging Topics in Computing*, *7*(2), 314–323.

Palattella, M. R., Dohler, M., Grieco, A., Rizzo, G., Torsner, J., Engel, T., & Ladid, L. (2016). Internet of things in the 5g era: Enablers, architecture, and business models. *IEEE journal on selected areas in communications*, *34*(3), 510–527.

Palmer, D. (2019). *Labour Party Confirms Cyber Attack Was DDoS.* Retrieved 2021-04-01, from https://www.zdnet.com/article/large-scale-cyberattack-hits-labour-party-systems/

Pandarinath, P. (2011). Secure localization with defense against selective forwarding attacks in wireless sensor networks. In *2011 3rd international conference on electronics computer technology* (Vol. 5, pp. 112–117).

Pang, Z. (2013). *Technologies and architectures of the internet-of-things (iot) for health and well-being* (Unpublished doctoral dissertation). KTH Royal Institute of Technology.

Parsaei, M. R., Rostami, S. M., & Javidan, R. (2016). A hybrid data mining approach for intrusion detection on imbalanced nsl-kdd dataset. *International Journal of Advanced Computer Science and Applications*, *7*(6).

Pasupa, K., Vatathanavaro, S., & Tungjitnob, S. (2020). Convolutional neural networks based focal loss for class imbalance problem: A case study of canine red blood cells morphology classification. *Journal of Ambient Intelligence and Humanized Computing*, 1–17.

Patel, A., & Jinwala, D. (2022). A reputation-based rpl protocol to detect selective forwarding attack in internet of things. *International Journal of Communication Systems*, *35*(1), e5007.

Pathak, A. K., Saguna, S., Mitra, K., & Åhlund, C. (2021). Anomaly detection using machine learning to discover sensor tampering in iot systems. In *Icc 2021-ieee international conference on communications* (pp. 1–6).

Pathan, A.-S. K., Abduallah, W. M., Khanam, S., & Saleem, H. Y. (2013). A pay-and-stay model for tackling intruders in hybrid wireless mesh networks. *Simulation*, *89*(5), 616–634.

Patro, S., & Sahu, K. K. (2015). Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*.

Petrov, V., Edelev, S., Komar, M., & Koucheryavy, Y. (2014). Towards the era of wireless keys: How the iot can change authentication paradigm. In *2014 ieee world forum on internet of things (wf-iot)* (pp. 51–56).

Pongle, P., & Chavan, G. (2015). A survey: Attacks on rpl and 6lowpan in iot. In *2015 international conference on pervasive computing (icpc)* (pp. 1–6).

Protić, D. D. (2018). Review of kdd cup'99, nsl-kdd and kyoto 2006+ datasets. *Vojnotehnički glasnik*, *66*(3), 580–596.

Rabbachin, A., Conti, A., & Win, M. Z. (2011). Intentional network interference for denial of wireless eavesdropping. In *2011 ieee global telecommunications conference-globecom 2011* (pp. 1–6).

Rabin, M. O. (1979). *Digitalized signatures and public-key functions as intractable as factorization* (Tech. Rep.). Massachusetts Inst of Tech Cambridge Lab for Computer Science.

Rácz, A., Bajusz, D., & Héberger, K. (2019). Multi-level comparison of machine learning classifiers and their performance metrics. *Molecules*, *24*(15), 2811.

Rao, B. S., & Premchand, P. (2018). Evaluation of differential–linear cryptanalysis combined attack on cryptographic security system. *Int. J. Appl. Eng. Res.*, *13*(23), 16552–16563.

Ravipati, R. D., & Abualkibash, M. (2019). Intrusion detection system classification using different machine learning algorithms on kdd-99 and nsl-kdd datasets-a review paper. *International Journal of Computer Science & Information Technology (IJCSIT) Vol*, *11*.

Ray, P. P. (2018). A survey on internet of things architectures. *Journal of King Saud University-Computer and Information Sciences*, *30*(3), 291–319.

Raza, S., Duquennoy, S., Chung, T., Yazar, D., Voigt, T., & Roedig, U. (2011). Securing communication in 6lowpan with compressed ipsec. In *2011 international conference on distributed computing in sensor systems and workshops (dcoss)* (pp. 1–8).

Raza, S., Wallgren, L., & Voigt, T. (2013). Svelte: Real-time intrusion detection in the internet of things. *Ad hoc networks*, *11*(8), 2661–2674.

Ren, W., Yu, L., Ma, L., & Ren, Y. (2013). How to authenticate a device? formal authentication models for m2m communications defending against ghost compromising attack. *International Journal of Distributed Sensor Networks*, *9*(2), 679450.

Rescorla, E. (1999). *Diffie-hellman key agreement method.* RFC Editor.

Reshan, A., & Saleh, M. (2021). Iot-based application of information security triad. *International Journal of Interactive Mobile Technologies*, *15*(24).

Sahraoui, S., & Bilami, A. (2015). Efficient hip-based approach to ensure lightweight end-to-end security in the internet of things. *Computer Networks*, *91*, 26–45.

Sajjad, M., Khan, S., Hussain, T., Muhammad, K., Sangaiah, A. K., Castiglione, A., ... Baik, S. W. (2019). Cnn-based anti-spoofing two-tier multi-factor authentication system. *Pattern Recognition Letters*, *126*, 123–131.

Sak, H., Senior, A., & Beaufays, F. (2014). Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv:1402.1128*.

Salameh, H. A. B., Almajali, S., Ayyash, M., & Elgala, H. (2018). Spectrum assignment in cognitive radio networks for internet-of-things delay-sensitive applications under jamming attacks. *IEEE Internet of Things Journal*, *5*(3), 1904–1913.

Samuel, S. S. I. (2016). A review of connectivity challenges in iot-smart home. In *2016 3rd mec international conference on big data and smart city (icbdsc)* (pp. 1–4).

Sathish, R., & Kumar, D. R. (2013). Dynamic detection of clone attack in wireless sensor networks. In *2013 international conference on communication systems and network technologies* (pp. 501–505).

Sauka, K., Shin, G.-Y., Kim, D.-W., & Han, M.-M. (2022). Adversarial robust and explainable network intrusion detection systems based on deep learning. *Applied Sciences*, *12*(13), 6451.

Sayakkara, A., Le-Khac, N.-A., & Scanlon, M. (2019). Leveraging electromagnetic side-channel analysis for the investigation of iot devices. *Digital Investigation*, *29*, S94–S103.

Schaffer, P., Farkas, K., Horvath, A., Holczer, T., & Buttyan, L. (2012). Secure and reliable clustering in wireless sensor networks: a critical survey. *Computer Networks*, *56*(11), 2726–2741.

Sethi, P., & Sarangi, S. R. (2017). Internet of things: architectures, protocols, and applications. *Journal of Electrical and Computer Engineering*, *2017*.

Seyhan, K., Nguyen, T. N., Akleylek, S., & Cengiz, K. (2021). Lattice-based cryptosystems for the security of resource-constrained iot devices in post-quantum world: a survey. *Cluster Computing*, 1–20.

Shah, P., Arora, M., & Adhvaryu, K. (2020). Lightweight cryptography algorithms in iot-a study. In *2020 fourth international conference on i-smac (iot in social, mobile, analytics and cloud)(i-smac)* (pp. 332–336).

Shang, B., Liu, S., Lu, S., Yi, Y., Shi, W., & Liu, L. (2020). A cross-layer optimization framework for distributed computing in iot networks. In *2020 ieee/acm symposium on edge computing (sec)* (pp. 440–444).

Sharma, V., You, I., Andersson, K., Palmieri, F., Rehmani, M. H., & Lim, J. (2020). Security, privacy and trust for smart mobile-internet of things (m-iot): A survey. *IEEE Access*, *8*, 167123–167163.

Sharmeen, S., Huda, S., Abawajy, J. H., Ismail, W. N., & Hassan, M. M. (2018). Malware threats and detection for industrial mobile-iot networks. *IEEE access*, *6*, 15941–15957.

Shi, Y., Sagduyu, Y. E., Erpek, T., Davaslioglu, K., Lu, Z., & Li, J. H. (2018). Adversarial deep learning for cognitive radio security: Jamming attack and defense strategies. In *2018 ieee international conference on communications workshops (icc workshops)* (pp. 1–6).

Shila, D. M., & Anjali, T. (2008). Defending selective forwarding attacks in wmns. In *2008 ieee international conference on electro/information technology* (pp. 96–101).

Shim, K.-A. (2019). Universal forgery attacks on remote authentication schemes for wireless body area networks based on internet of things. *IEEE Internet of Things Journal*, *6*(5), 9211–9212.

Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. *IEEE transactions on emerging topics in computational intelligence*, *2*(1), 41–50.

Shoreh, M. H., Hosseinianfar, H., Akhoundi, F., Yazdian, E., Farhang, M., & Salehi, J. A. (2014). Design and implementation of spectrally-encoded spread-time cdma transceiver. *IEEE communications letters*, *18*(5), 741–744.

Singh, K., Kaur, L., & Maini, R. (2021). Comparison of principle component analysis and stacked autoencoder on nsl-kdd dataset. In *Computational methods and data engineering* (pp. 223–241). Springer.

Singh, V. P., Jain, S., & Singhai, J. (2010). Hello flood attack and its countermeasures in wireless sensor networks. *International Journal of Computer Science Issues (IJCSI)*, *7*(3), 23.

Sivaganesan, D. (2021). A data driven trust mechanism based on blockchain in iot sensor networks for detection and mitigation of attacks. *Journal of trends in Computer Science and Smart technology (TCSST)*, *3*(01), 59–69.

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., & Manning, C. D. (2011). Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 conference on empirical methods in natural language processing* (pp. 151–161).

Soe, Y. N., Feng, Y., Santosa, P. I., Hartanto, R., & Sakurai, K. (2019). Implementing lightweight iot-ids on raspberry pi using correlation-based feature selection and its performance evaluation. In *International conference on advanced information networking and applications* (pp. 458–469).

Sohn, K., Lee, H., & Yan, X. (2015). Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, *28*, 3483–3491.

Stefansson, G., & Lumsden, K. (2009). Performance issues of smart transportation management systems. *International Journal of productivity and performance management*.

Stiawan, D., Idris, M., Malik, R. F., Nurmaini, S., Alsharif, N., Budiarto, R., et al. (2019). Investigating brute force attack patterns in iot network. *Journal of Electrical and Computer Engineering*, *2019*.

Su, T., Sun, H., Zhu, J., Wang, S., & Li, Y. (2020). Bat: Deep learning methods on network intrusion detection using nsl-kdd dataset. *IEEE Access*, *8*, 29575–29585.

Sudharsan, B., Breslin, J. G., & Ali, M. I. (2020). Rce-nn: a five-stage pipeline to execute neural networks (cnns) on resource-constrained iot edge devices. In *Proceedings of the 10th international conference on the internet of things* (pp. 1–8).

Sudqi Khater, B., Abdul Wahab, A. W. B., Idris, M. Y. I. B., Abdulla Hussain, M., & Ahmed Ibrahim, A. (2019). A lightweight perceptron-based intrusion detection system for fog computing. *applied sciences*, *9*(1), 178.

Sundmaeker, H., Guillemin, P., Friess, P., Woelfflé, S., et al. (2010). Vision and challenges for realising the internet of things. *Cluster of European research projects on the internet of things, European Commision*, *3*(3), 34–36.

Suo, H., Wan, J., Zou, C., & Liu, J. (2012). Security in the internet of things: a review. In *2012 international conference on computer science and electronics engineering* (Vol. 3, pp. 648–651).

Sutskever, I. (2013). *Training recurrent neural networks*. University of Toronto Toronto, Canada.

Svensson, G. (2013). *Auditing the human factor as a part of setting up an information security management system.*

Sweeney, L. (2002). k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *10*(05), 557–570.

Sze, V., Chen, Y.-H., Yang, T.-J., & Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, *105*(12), 2295–2329.

Taherkordi, A., Eliassen, F., & Horn, G. (2017). From iot big data to iot big services. In *Proceedings of the symposium on applied computing* (pp. 485–491).

Tahir, R. (2018). A study on malware and malware detection techniques. *International Journal of Education and Management Engineering*, *8*(2), 20.

Tandon, A., & Srivastava, P. (2019). Trust-based enhanced secure routing against rank and sybil attacks in iot. In *2019 twelfth international conference on contemporary computing (ic3)* (pp. 1–7).

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009a). A detailed analysis of the kdd cup 99 data set. In *2009 ieee symposium on computational intelligence for security and defense applications* (pp. 1–6).

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009b). Nsl-kdd dataset. Retrieved 2017-09-23, from `https://www.unb.ca/cic/datasets/nsl.html`

Telikani, A., & Gandomi, A. H. (2021). Cost-sensitive stacked auto-encoders for intrusion detection in the internet of things. *Internet of Things*, *14*, 100122.

Tertytchny, G., Nicolaou, N., & Michael, M. K. (2020). Classifying network abnormalities into faults and attacks in iot-based cyber physical systems using machine learning. *Microprocessors and Microsystems*, *77*, 103121.

Thakkar, A., & Lohiya, R. (2021). A review on machine learning and deep learning perspectives of ids for iot: recent updates, security issues, and challenges. *Archives of Computational Methods in Engineering*, *28*(4), 3211–3243.

Thakor, V. A., Razzaque, M. A., & Khandaker, M. R. (2021). Lightweight cryptography algorithms for resource-constrained iot devices: A review, comparison and research opportunities. *IEEE Access*, *9*, 28177–28193.

Thamilarasu, G., & Chawla, S. (2019). Towards deep-learning-driven intrusion detection for the internet of things. *Sensors*, *19*(9), 1977.

Tharwat, A. (2020). Classification assessment methods. *Applied Computing and Informatics*.

Thomas, R., & Pavithran, D. (2018). A survey of intrusion detection models based on nsl-kdd data set. *2018 Fifth HCT Information Technology Trends (ITT)*, 286–291.

Tian, X., Wu, D., Wang, R., & Cao, X. (2018). Focal text: an accurate text detection with focal loss. In *2018 25th ieee international conference on image processing (icip)* (pp. 2984–2988).

Tschannen, M., Bachem, O., & Lucic, M. (2018). Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*.

Tschofenig, H., Arkko, J., Thaler, D., & McPherson, D. (2015). Architectural considerations in smart object networking. *RFC 7452*.

Usman, M., Ahmed, I., Aslam, M. I., Khan, S., & Shah, U. A. (2017). Sit: a lightweight encryption algorithm for secure internet of things. *arXiv preprint arXiv:1704.08688*.

Uthumansa, A., & Shantha, F. (2020). Identifying the impacts of active and passive attacks on network layer in a mobile ad-hoc network: a simulation perspective.

Vaidya, R. (2019). Cyber security breaches survey 2019. *Department for Digital, Culture, Media and Sport*, *66*.

Vaiyapuri, T., & Binbusayyis, A. (2020). Application of deep autoencoder as an one-class classifier for unsupervised network intrusion detection: a comparative evaluation. *PeerJ Computer Science*, *6*, e327.

Vidalis, S., & Angelopoulou, O. (2014). Assessing identity theft in the internet of things. *Journal of IT Convergence Practice*.

Vijayakumar, P., Obaidat, M. S., Azees, M., Islam, S. H., & Kumar, N. (2019). Efficient and secure anonymous authentication with location privacy for iot-based wbans. *IEEE Transactions on Industrial Informatics*, *16*(4), 2603–2611.

Vinayakumar, R., Alazab, M., Soman, K., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. *IEEE Access*, *7*, 41525–41550.

Virat, M. S., Bindu, S., Aishwarya, B., Dhanush, B., & Kounte, M. R. (2018). Security and privacy challenges in internet of things. In *2018 2nd international conference on trends in electronics and informatics (icoei)* (pp. 454–460).

Vu, L., Bui, C. T., & Nguyen, Q. U. (2017). A deep learning based method for handling imbalanced problem in network traffic classification. In *Proceedings of the eighth international symposium on information and communication technology* (pp. 333–339).

Wallgren, L., Raza, S., & Voigt, T. (2013). Routing attacks and countermeasures in the rpl-based internet of things. *International Journal of Distributed Sensor Networks*, *9*(8), 794326.

Wara, M. S., & Yu, Q. (2020). New replay attacks on zigbee devices for internet-of-things (iot) applications. In *2020 ieee international conference on embedded software and systems (icess)* (pp. 1–6).

Wolinsky, D. I., Syta, E., & Ford, B. (2013). Hang with your buddies to resist intersection attacks. In *Proceedings of the 2013 acm sigsac conference on computer & communications security* (pp. 1153–1166).

Wu, C.-K. (2021). Iot network layer security. In *Internet of things security* (pp. 107–123). Springer.

Wu, M., Lu, T.-J., Ling, F.-Y., Sun, J., & Du, H.-Y. (2010). Research on the architecture of internet of things. In *2010 3rd international conference on advanced computer theory and engineering (icacte)* (Vol. 5, pp. V5–484).

Xiang, M., Bai, Q., & Liu, W. (2014). Trust-based adaptive routing for smart grid systems. *Journal of information processing*, *22*(2), 210–218.

Xiao, L., Li, Y., Han, G., Liu, G., & Zhuang, W. (2016). Phy-layer spoofing detection with reinforcement learning in wireless networks. *IEEE Transactions on Vehicular Technology*, *65*(12), 10037–10047.

Xiao, S., Wang, H., & Zhang, J. (2021). New digital signature algorithm based on ecc and its application in bitcoin and iot. *International Journal of High Performance Systems Architecture*, *10*(1), 20–31.

Xiao, Y., Xing, C., Zhang, T., & Zhao, Z. (2019). An intrusion detection model based on feature reduction and convolutional neural networks. *IEEE Access*, *7*, 42210–42219.

Xin, M. (2015). A mixed encryption algorithm used in internet of things security transmission system. In *2015 international conference on cyber-enabled distributed computing and knowledge discovery* (pp. 62–65).

Xu, W., Jang-Jaccard, J., Singh, A., Wei, Y., & Sabrina, F. (2021). Improving performance of autoencoder-based network anomaly detection on nsl-kdd dataset. *IEEE Access*, *9*, 140136–140146.

Xu, X., Li, J., Yang, Y., & Shen, F. (2020). Toward effective intrusion detection using

log-cosh conditional variational autoencoder. *IEEE Internet of Things Journal*, *8*(8), 6187–6196.

Yan, Z., Zhang, P., & Vasilakos, A. V. (2014). A survey on trust management for internet of things. *Journal of network and computer applications*, *42*, 120–134.

Yang, L., Ding, C., & Wu, M. (2013). Establishing authenticated pairwise key using pairing-based cryptography for sensor networks. In *2013 8th international conference on communications and networking in china (chinacom)* (pp. 517–522).

Yang, Y., Wu, L., Yin, G., Li, L., & Zhao, H. (2017). A survey on security and privacy issues in internet-of-things. *IEEE Internet of Things Journal*, *4*(5), 1250–1258.

Yang, Y., Zheng, K., Wu, B., Yang, Y., & Wang, X. (2020). Network intrusion detection based on supervised adversarial variational auto-encoder with regularization. *IEEE Access*, *8*, 42169–42184.

Yang, Y., Zheng, K., Wu, C., & Yang, Y. (2019). Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network. *Sensors*, *19*(11), 2528.

Yang, Y., Zheng, X., & Tang, C. (2017). Lightweight distributed secure data management system for health internet of things. *Journal of Network and Computer Applications*, *89*, 26–37.

Yang, Z., Yue, Y., Yang, Y., Peng, Y., Wang, X., & Liu, W. (2011). Study and application on the architecture and key technologies for iot. In *2011 international conference on multimedia technology* (pp. 747–751).

Yao, X., Chen, Z., & Tian, Y. (2015). A lightweight attribute-based encryption scheme for the internet of things. *Future Generation Computer Systems*, *49*, 104–112.

Yaqoob, I., Hashem, I. A. T., Mehmood, Y., Gani, A., Mokhtar, S., & Guizani, S. (2017). Enabling communication technologies for smart cities. *IEEE Communications Magazine*, *55*(1), 112–120.

Yarotsky, D. (2017). Error bounds for approximations with deep relu networks. *Neural Networks*, *94*, 103–114.

Yavuz, F. Y., Ünal, D., & Gül, E. (2018). Deep learning for detection of routing attacks in the internet of things. *International Journal of Computational Intelligence Systems*, *12*(1), 39–58.

Yehia, L., Khedr, A., Darwish, A., et al. (2015). Hybrid security techniques for internet of things healthcare applications. *Advances in Internet of Things*, *5*(03), 21.

Yi, P., fei Hou, Y., Zhong, Y., Zhang, S., & Dai, Z. (2006). Flooding attack and defence in ad hoc networks. *Journal of Systems Engineering and Electronics*, *17*(2), 410–416.

Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *Ieee Access*, *5*, 21954–21961.

Yu, L., Zhou, R., Chen, R., & Lai, K. K. (2022). Missing data preprocessing in credit classification: One-hot encoding or imputation? *Emerging Markets Finance and Trade*, *58*(2), 472–482.

Yu, Y., Moraitis, M., & Dubrova, E. (2021). Can deep learning break a true random number generator? *IEEE Transactions on Circuits and Systems II: Express Briefs*, *68*(5), 1710–1714.

Yun, P., Tai, L., Wang, Y., Liu, C., & Liu, M. (2019). Focal loss in 3d object detection. *IEEE Robotics and Automation Letters*, *4*(2), 1263–1270.

Zaman, M., & Lung, C.-H. (2018). Evaluation of machine learning techniques for network intrusion detection. In *Noms 2018-2018 ieee/ifip network operations and management symposium* (pp. 1–5).

Zhang, J., & Varadharajan, V. (2010). Wireless sensor network key management survey and taxonomy. *Journal of network and computer applications*, *33*(2), 63–75.

Zhang, J., Wu, Q., Zheng, R., Zhu, J., Zhang, M., & Liu, R. (2018). A security monitoring method based on autonomic computing for the cloud platform. *Journal of Electrical and Computer Engineering*, *2018*.

Zhang, W., & Qu, B. (2013). Security architecture of the internet of things oriented to perceptual layer. *International Journal on Computer, Consumer and Control (IJ3C)*, *2*(2), 37–45.

Zhang, Y., & Pengfei, J. (2014). An efficient and hybrid key management for heterogeneous wireless sensor networks. In *The 26th chinese control and decision conference (2014 ccdc)* (pp. 1881–1885).

Zhou, T., Choudhury, R. R., Ning, P., & Chakrabarty, K. (2011). P2dap—sybil attacks detection in vehicular ad hoc networks. *IEEE journal on selected areas in communications*, *29*(3), 582–594.

Zou, Y., & Wang, G. (2015). Intercept behavior analysis of industrial wireless sensor networks in the presence of eavesdropping attack. *IEEE Transactions on Industrial Informatics*, *12*(2), 780–787.

Zuech, R., Hancock, J., & Khoshgoftaar, T. M. (2021). Detecting web attacks in severely imbalanced network traffic data. In *2021 ieee 22nd international conference on information reuse and integration for data science (iri)* (pp. 267–273).