

UNIVERSITY OF BAHRAIN
College of Information Technology



Deep Clustering for IoT Anomaly Detection and Device Identification

A Thesis Submitted in Partial Fulfilment of the Requirements for the
M.Sc. Degree in Machine Learning and Computational Intelligence

Submitted by
Noor Hussain Abdulla Sarhan
ID Number: 20062375

Supervised by
Dr. Hesham Al-Ammal
Assistant Professor
University of Bahrain

Co-Supervised by
Dr. Hessa Al-Junaid
Assistant Professor
University of Bahrain

Kingdom of Bahrain
April 2023

Defense Committee's Approval Page

University of Bahrain
Deanship of Graduate Studies
And scientific Research



جامعة البحرين
صناديق الدراسات العليا والبحث العلمي

Thesis Modifications Approval Form

Student Name: Noor Hussain Abdulla Sarhan	Student ID: 20062375
Department: Computer Science	College: Information Technology
Program: M.Sc. in Machine Learning and Computational Intelligence	
Thesis Title: Deep Clustering for IoT Anomaly Detection and Device Identification	
Supervisor Name: Dr. Hesham Al-Ammal and Dr. Hessa Al-Junaid	Student GPA: 3.78

We would like to report that the student has carried out all the required modifications made by the discussion committee. Upon this, we recommend awarding the student a grade **Pass** Choose an item. In the thesis course number **ITML 607** Click here to enter text. based on the Discussion Committee recommendation. Choose an item.

Endorsement:

	Name	Academic Title	Signature	Date
1.	Dr. Hesham Al-Ammal and Dr. Hessa Al-Junaid	Advisor		3-9-2023
2.	Dr. Wael Elmedany	Internal Examiner		
3.	Dr. Nabil Hewahi	Department Coordinator		3/9/2023
4.	Dr. Ahmad Fahad	Department Chairman		19/9/2023
5.	Dr. Hessa Al-Junaid	College Dean		

7 Approved at the University Council 6th meeting, 2013
In accordance with Decision No. 736/2013 dated 14/3/2013
Modified by University Council Decision No. 2043/2013 dated 20/11/2013.

Abstract

IoT devices are small physical objects that can communicate through a specific network. IoT employment has demonstrated substantial growth in the past years specifically during Covid-19 pandemic. A key pursuit for the IoT market is to support vital sectors with new sophisticated technology such as health and education specially in smart cities such as Shanghai, Seoul, and New York. As the use of IoT devices continues to grow, the demand for effective anomaly detection and device identification solutions is likely to increase. IoT networks and devices can be vulnerable against cyber-attacks because of IoT device's characteristics such as limited computational power, memory, and battery life. Because of such features, it is challenging to implement advanced identification schemes

This thesis aims to implement automated IoT anomaly detection and device identification scheme using deep clustering algorithms. The significance of the research comes from the urgent need for powerful identification models that can cope with IoT changing devices and environment and to be able to detect known and unknown anomalies and devices. The scope of this thesis's experiments is to investigate both classification and clustering algorithms to detect anomalies and identify IoT devices. Several machine learning and deep learning algorithms were selected based on performance and effectiveness. All algorithms were trained and tested using two public IoT datasets, namely CIC-IoT-2022 and IoT23. For classification purposes, the selected algorithms were Convolutional Neural Network (CNN), Artificial Neural Network (ANN), Random Forest (RF), Decision Tree (DT), and K-nearest Neighbor (KNN). For clustering purposes, the selected algorithms were K-Means (KM), Deep Clustering Network (DCN), Deep Embedding Clustering (DEC), and Dynamic Auto-Encoder (DynAE). The performance of the applied algorithms was evaluated using several evaluation measures such as confusion matrix and Normalized Mutual Information (NMI). It was concluded that the DynAE algorithm is the most effective deep clustering algorithm with 81% accuracy on IoT23 malware dataset.

Dedication

This work is dedicated to people who believed in me all the way... To my mother, father, and husband. This dedication is for all the patience I have depleted, the time I have employed, and the support I have benefited from to achieve all my goals. They have been the source of courage and inspiration and this work will not be done without their help and support.

Lastly, I dedicated my work to the light of my life, my kids, Yusuf and Hussain, for giving me the courage to follow and achieve my dreams.

Acknowledgment

I would like to express my deepest appreciation to Dr. Hesham Al-Ammal and Dr. Hessa Al-Junaid for their endless support throughout my master's program and thesis stage.

Also, I would like to convey my gratitude to my leaders at Khawla Secondary Girls School, Mrs. Manal Senan, and Mrs. Fatima Al-Hussaini for their continuous support and help.

Table of Contents

Defense Committee's Approval Page.....	B
Abstract.....	C
Dedication	D
Acknowledgment	E
List of Tables.....	J
List of Figures.....	K
Abbreviations	M
Chapter One Introduction to the Study	1
1.1 Background.....	1
1.2 Problem Definition.....	2
1.3 Research Objectives	2
1.4 Research Aim.....	3
1.5 Research Significance.....	4
1.6 Thesis Organization.....	5
Chapter Two Literature Review	6
2.1 Background.....	6
2.1.1 State-of-the-art Algorithms	8
2.1.2 IoT Devices Profiling and Identification	10
2.1.3 IoT Anomaly Detection	10
2.2 Machine Learning Algorithms	13
2.2.1 IoT Devices Identification Using ML	14

2.2.2	IoT Anomaly Detection Using ML.....	17
2.3	Deep Learning Algorithms.....	22
2.3.1	IoT Devices Identification Using DL.....	22
2.3.2	IoT Anomaly Detection Using Deep Learning	25
2.4	Summary	28
Chapter Three Research Methodology.....		30
3.1	Projected Methodology.....	30
3.2	Datasets	31
3.2.1	CIC-IoT-2022 Dataset Description.....	32
3.2.2	IoT23 Dataset Description	37
3.3	Data preprocessing.....	40
3.3.1	Prepare data for Loading.....	40
3.3.2	Features Selections	42
3.3.3	Features Engineering	44
3.4	Selected Algorithms	45
3.4.1	Machine Learning Algorithms.....	46
i.	K Nearest Neighbor (KNN)	46
i.	Random Forest (RF).....	47
iii.	Artificial Neural Network (ANN).....	48
iv.	Density-Based Clustering Algorithm (DBSCAN)	49
v.	K-Means Clustering (KM).....	49
3.4.2	Deep Learning Algorithms.....	50

i. Convolutional Neural Network (CNN).....	50
ii. Dynamic Auto-Encoder (DynAE)	50
iii. Deep Embedded Clustering (DEC).....	51
iv. Deep Clustering Network (DCN)	51
3.5 Evaluation Matrices.....	52
3.6 Summary	54
Chapter Four Analysis and Discussion	55
4.1 Data Analysis Results	55
4.1.1 Analysis of IoT23 dataset.....	55
4.1.2 Analysis of CIC-IoT-2022 dataset.....	57
4.2 Experiments Results	58
4.2.1 Classification Algorithms Results.....	59
i. ANN	60
ii. CNN	62
iii. DT, RF and KNN	64
4.2.2 Clustering Algorithms Results.....	64
i. ML algorithms: KM and DBSCAN.....	65
ii. Deep Learning Algorithms: DCN, DEC and DynAE.....	67
4.3 Discussion	70
4.4 Summary	72
Chapter Five Conclusion and Recommendations	73
5.1 Conclusion.....	73

5.2 Future Work and Research Limitations.....	74
References	77

List of Tables

Table (3.1): CIC-IoT-2022 Dataset Features.....	33
Table(3.2): Labels Distribution in CIC-IoT-2022 dataset.....	
Table(3.3): Labels Distribution in IoT23 dataset.....	39
Table (3.4): Comparison between Standardization and Normalization	45
Table (3.5): Classifiers Evaluation Measures	52
Table (4.1): ML Algorithms Performance Comparison	64
Table (4.2): Results of Deep Clustering Algorithms VS ML clustering Algorithms.....	70
Table (4.3): Comparison of ML classifiers with work of CIC-IoT-2022 developers	71

List of Figures

Figure (2.1): Growth of IoT devices from 2010 to 2025 (Duarte, 2023)	7
Figure (2.2): Demonstration of anomaly datapoints detected using ML (Davis David ,2019).....	11
Figure (2.3): Categories of Anomalies in IoT Networks (Chatterjee & Bestoun, 2022).....	12
Figure (2.4): Types of ML algorithms (Abdul Wahid, 2017)	13
Figure (2.5): Components of DEFT Model (Vijayanand Thangavelu, 2019)	17
Figure (2.6): Results of ML classifiers on MQTT-IoT-IDS2020 dataset	18
Figure (2.7): F1-score for multiclass classification (Vitorino, Andrade, Praça, Sousa, & Maia, 2022).....	19
Figure (2.8):F1-score for proposed method compared to other classifiers. (Kozik, Pawlicki, & Choraś, 2021)	25
Figure (2.9): Accuracy scores comparison between 3 models	27
Figure (3.1): Label distribution among global category feature in CIC dataset	37
Figure (3.2): IoT23 Features (Dutta, Choraś, Pawlicki, & Kozik, 2020).....	38
Figure (3.3): IoT23 dataset employment from 2020 - 2022.....	39
Figure (3.4): Convert stored captures to data frame in IoT23 dataset.....	41
Figure (3.5): re-label labels to be unified in IoT23 dataset.....	41
Figure (3.6): Partial view of correlation coefficient for CIC-IoT-2022 dataset	42
Figure (3.7): How FR operates (Khan, et al., 2021).....	47
Figure (3.8): Random Forest Vs. Decision Tree (Saini, 2021)	48
Figure 4.1): Counts of records for each attack.....	56

Figure (4.2): Unique Values of Specific Features in IoT23 dataset.....	56
Figure (4.3): Samples Count of IoT Devices	57
Figure (4.4): Global Category of Deices in the Dataset	57
Figure (4.5): Devices classification based on Port class.....	58
Figure (4.6): Packet average size per devices categories.....	58
Figure (4.7): Accuracy Vs Loss for ANN	60
Figure (4.8): Confusion Matrix for IoT devices classification based on global category.....	61
Figure (4.9): Device type-based identification using ANN	61
Figure (4.10): CNN Structure Summary	62
Figure (4.11): Confusion Matrix for CNN model.....	63
Figure (4.12): Classification Report for CNN Model	63
Figure (4.13): Elbow graph to select K in CIC-IoT-2022 dataset	66
Figure (4.14): Clusters created by KM on IoT23 dataset	66
Figure (4.15): Clusters created by DBSCAN algorithm.....	67
Figure (4.16): Deep Clustering in 1st epoch VS last epoch using DCN	68
Figure (4.17): Loss Value Comparison the epochs	68

Abbreviations

AD	Anomaly Detection
AI	Artificial Intelligence
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
DBSCAN	Density-Based Clustering Algorithm
DCN	Deep Clustering Network
DL	Deep Learning
DT	Decision Tree
DNN	Deep Neural Network
DynAE	Dynamic Auto-Encoder
GAN	Generative Adversarial Network
IDS	Intrusion Detection System
IoT	Internet of Things
IP	Internet Protocol
KNN	K-nearest Neighbor
LoRaWAN	Long Range Wide Area Network
LSTM	Long-short Term Memory
MAC	Media Access Control
ML	Machine Learning
NB	Naïve Bayes

OCSVM	One-Class Support Vector Machine
OD	Outlier Detection
PCA	Principal Component Analysis
PCAP	Packet Capture
Pyod	Python Library for Outlier Detection
RF	Random Forest
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
XGBoost	eXtreme Gradient Boosting

Chapter One

Introduction to the Study

In this chapter, the thesis context is outlined to provide an overview of IoT concepts and brief definition. The idea behind the conducted research is presented along with the aims and objectives, the research questions, and the research significance. The last section of the chapter summarizes the thesis by providing a brief outline of each chapter's details.

1.1 Background

The concept of Internet of Things (IoT) has been used since 1999 (Sharma, Shamkuwar, & Singh, 2019) due to the massive increase devices' number and objects depending on the Internet to establish communication and function effectively. IoT is an AI concept that has a major impact in our daily life. It can significantly improve education, finance, and security sectors. With this increased usage of IoT devices, comes the importance of security and privacy aspect of all connected devices.

Anomaly detection is an important task of any powerful IoT security system. It has been discussed in the literature for a long time. The importance of anomaly detection lies in the need to identify any abnormal behavior or connection in IoT network. Also, profiling IoT devices enhances network security by identifying infected devices (with unexpected behavior) that can be a source of security threats or information leaks.

1.2 Problem Definition

The continued need for stable cybersecurity techniques is significant to enhance the security level of IoT networks. IoT devices have special features and characteristics that can affect their ability to secure user's data. IoT devices have limited power and memory which can result in producing insecure cryptographic keys (Safi, et al., 2022). According to that, IoT devices are exposed to an extensive range of attacks such as spoofing and broken authentication.

IoT AD and device identification can be difficult for several reasons. The first reason is that IoT networks usually contain different types of devices which lead to multiple protocols and control sequences. Also, the number and type of IoT connected devices is increasing, which means that using a simplified general technique is not sufficient.

Furthermore, most of the literature attempts to perform classification and supervised learning for identification of IoT devices and AD, which is inadequate if the attack is a new (i.e., zero-day) attack due to unavailability of labeled training data.

1.3 Research Objectives

This research is attempted to analyze network communication patterns and investigate different unsupervised clustering algorithms to identify the type of IoT devices and create communication profiles that can help identify anomalous communications in the future. The next points present the research objectives and contributions:

1. Perform automated IoT device identification based on IoT network activity's specific features using different ML algorithms utilizing new IoT devices datasets.
2. Investigate the use of unsupervised deep clustering algorithms to enhance IoT devices identification process.
3. Investigate the use of deep clustering algorithms for anomaly detection. By accomplishing this objective, the research will provide a clear picture of the effectiveness and the efficiency of deploying DL clustering methods and algorithms.

1.4 Research Aim

The rapid growth of the IoT has led to an increase in the number of connected devices, making it challenging to identify and manage these devices. Anomaly detection is also crucial for detecting security breaches and other irregularities in IoT devices that may pose a threat to the network. Deep learning algorithms have shown significant promise in addressing these challenges by enabling the automatic identification of IoT devices and the detection of anomalous behavior. The study seeks to answer several research questions, including whether deep clustering algorithms can effectively detect anomalies in IoT devices and if these algorithms can perform well by creating generalized application across different IoT networks. Additionally, the research intends to explore whether deep clustering algorithms can provide additional benefits, such as improved accuracy and efficiency, compared to traditional ML classification algorithms. By answering these research questions, the study aims to contribute to the development of effective and efficient methods for securing IoT networks and devices. The main aim can be stated as to explore the effectiveness of state-of-the-art deep clustering algorithms in identifying

IoT devices and detecting anomalies, with the goal of improving the security and management of IoT networks

1.5 Research Significance

The popularity of IoT is rapidly increasing to become common devices in our daily lives. Different IoT devices are being deployed in various sectors such as the environment, smart homes, and health care (F. Meneghello, 2019). IoT devices have special characteristics such as small memory, low power, and controlled computation (Xiao Liang, 2018). Those features can affect their ability to perform effectively under potential or complex security events, especially with dense data streams. As their use grows, concerns about cybersecurity and attacks through impersonation, phishing, or other forms of security breaches become more evident.

Recent statistics shows that the number of IoT devices attacks are increasing every year. According to (Nuspire, 2022), there was 300% increase in IoT device attacks compared to the 2021 and the global monthly number of IoT cyber-attacks was 32.7 million. Studies discuss the affected of IoT security attacks on health sectors as 78% of healthcare organizations experienced an IoT-related security incident. These statistics along with other explains the real need behind building powerful and automated IoT anomaly detection and devices identification model.

This research will contribute to investigate the utilization of unsupervised learning methods in IoT anomaly detection. Unsupervised deep learning techniques are particularly appealing to intrusion detection systems since they detect known and unknown types of attacks as well as zero-day attacks. Also,

deep learning helps in avoiding the dimensionality reduction step, as autoencoders and other techniques can perform this step automatically.

1.6 Thesis Organization

The thesis involves five main chapters, organized as follows:

Chapter One: is the introduction chapter and contains an overview and needed background on IoT devices and anomaly detections in IoT networks. The chapter contains a description of the research problem, objectives, and significance.

Chapter Two: contains the literature review of the problem and consists of two parts. The first part discusses anomaly detection in IoT devices using supervised Machine Learning (ML) algorithms. The second part presents different applications of Deep Learning (DL) algorithms in the field of anomaly detection for IoT devices.

Chapter three: outlines the methodology and shows a full discussion of the accomplishment of the research objectives. The chapter is divided into two main sections. The first section explains different supervised ML algorithms used for IoT devices identification and profiling. The second part discusses different unsupervised DL algorithms that were investigated for anomaly detection.

Chapter four: is the results and discussion chapter, and it contains descriptions of the datasets and results of various algorithms that were tested on these datasets.

Chapter five: contains conclusions, recommendations, and possible future wo

Chapter Two

Literature Review

This chapter incorporates a broad review of the literature related to IoT device identification and anomaly detection. This chapter is divided into two sections. The First section presents the related work on using Machine Learning (ML) algorithms in IoT device profiling as well as anomaly detection. The second part is an extensive review of the usage of Deep Learning (DL) models in the same field of study. Both supervised and unsupervised algorithms are included in the review.

Studying different research papers is a critical step before starting to work on the research. Reporting other research's work helps in identifying the trends and patterns in the field of IoT anomaly detection and device identification. It helps to understanding of the most common approaches and methods being used, as well as the strengths and weaknesses of these approaches, which help to identify areas where further research is needed and can also help to guide the development of new approaches that build on the strengths of existing methods. After performing literature review search, the significance of the conducted research was easily identified, and the main objectives were clearly stated.

2.1 Background

Recent technologies intensely affect and change our daily lives, as they are employed in various vital sectors such as education, health, economy, and others. Numerous works are now done at the click of a button via the Internet. One of those important technologies is the Internet of Things (IoT).

IoT is defined as a technology that deploys the Internet to create a network to connect and control multiple devices and equipment (Patel, 2016). IoT network functions based on specific protocols and sensors to collect information from the devices that provided the opportunity to build a smart environment. According to (Vermesan Ovidiu, 2014), the main aim of IoT technology is to enable devices (or things) to communicate with people and other devices without considering the physical or time limitations.

By utilizing IoT based services in schools, companies' offices, or health-care centers, IoT can deliver new opportunities and support for different end users and fulfill their various needs (Simar Preet Singh, 2021). According to IoT Analytics (2023), the size of connected IoT devices has extended. Based on recent studies (Howarth, 2023), a major shift from non-IoT devices has been taking place over the last ten years and it's forecasted that more than 70% of the devices will be converted to IoT by the end of 2023. As shown in Figure 2.1 (Duarte, 2023), the number of IoT and non-IoT devices used in 2019 was around 10 billion devices while IoT devices usage has been growing dramatically to reach 20 billion active connections in the current year.

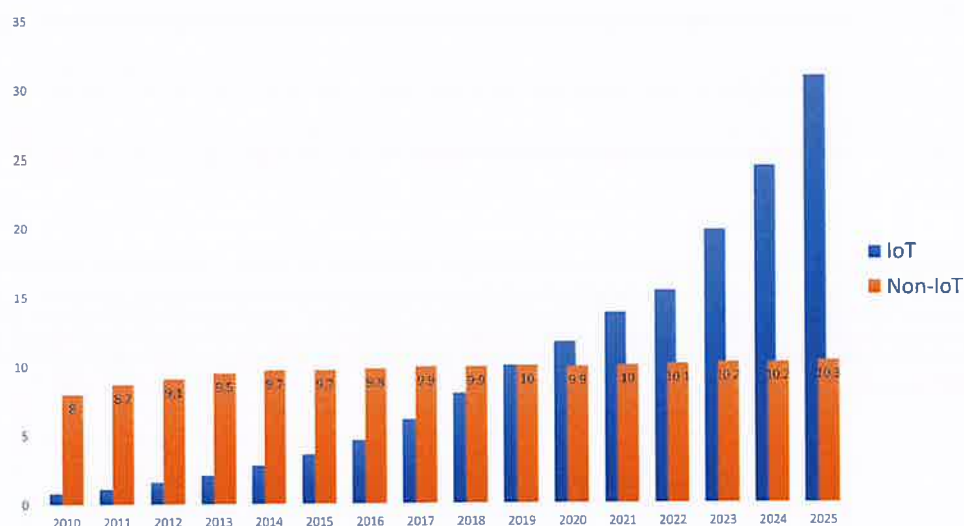


Figure (2.1): Growth of IoT devices from 2010 to 2025 (Duarte, 2023)

(Helmi Ben Hmida, 2015) explained the main characteristics of any IoT system. To build a stable IoT system, several attributes should be taken into consideration. Those attributes are as follows:

- *Intelligence*: Where the IoT system can gather and process raw data into knowledge by deploying intelligent reasoning and context understanding framework.
- *Architecture*: IoT architecture usually is combination of two architectures: time-based and event-based architectures. This mixture allows IoT systems to handle devices communication effectively and efficiently.
- *Complexity*: The communication process in IoT system is enhanced and developed based on how powerful the device is and its added value to the network. Considering the complexity of each device, the system should be able to handle better communication processes for improved data handling and sophisticated cognitive capabilities.
- *Size*: a powerful IoT system should be able to handle the growth of connected IoT devices. It is forecasted that more than 15 billion IoT devices are connected to the Internet in 2023 and it may be doubled by the year 2030 (Duarte, 2023). The IoT framework is required to take this density into consideration and assure the stability of the communication between all devices.

2.1.1 State-of-the-art Algorithms

State-of-the-art algorithms are continually advancing as new research and development leads to the creation of more advanced and effective techniques. They refer to the most advanced and effective algorithms that are currently

available for a specific problem or task (Mothe, Mkhitarian, & Haroutunian, 2017). These algorithms are typically developed by researchers and practitioners in a particular field, and represent the best practices based on the most recent research and development. For example, in the field of computer vision, convolutional neural networks have become the state-of-the-art algorithm for a wide range of image recognition tasks, but there are ongoing efforts to develop even more advanced architectures and techniques that can improve performance even further.

One important aspect of state-of-the-art algorithms is their ability to generalize well to new data and tasks (Neu, Lahann, & Fettke, 2021). This is particularly important in fields such as machine learning, where the ability to learn from diverse datasets and apply that knowledge to new situations is critical. State-of-the-art algorithms are designed to be robust and adaptable, with the ability to handle a wide range of inputs and outputs.

There are several state-of-the-art algorithms that have been used by researchers to test their effectiveness in the field of IoT anomaly detection and device identification. For instance, autoencoders as deep learning models that can learn to reconstruct inputs from a compressed representation. By training an autoencoder on normal data, anomalies can be detected as inputs that cannot be accurately reconstructed (Chatterjee & Bestoun, 2022). Other example is One-Class classification based on SVM. In this type of algorithms, the model is trained on only normal data, then the algorithm can identify inputs that are significantly different from what it has learned (Razzak, Zafar, Imran, & Xu, 2020).

In addition to the mentioned ML and DL algorithms, IoT devices can be identified using device fingerprinting. *Device Fingerprinting* (Ferman & Tawfeeq, 2021) is the process of identifying IoT devices based on their unique characteristics, such as MAC address, IP address, and network traffic patterns. This can be useful for network management and security purposes. There are several state-of-the-art algorithms for device fingerprinting, including machine learning-based approaches such as Random Forests, Decision Trees, and K-Nearest Neighbors.

2.1.2 IoT Devices Profiling and Identification

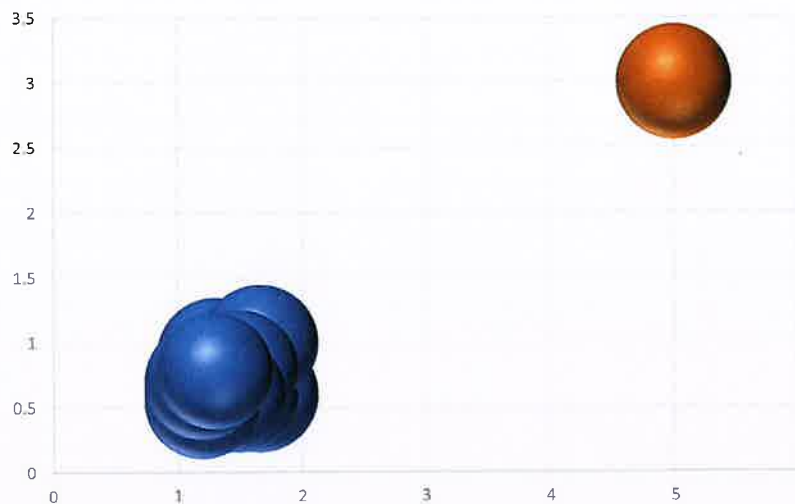
The usage of IoT devices is increasing rapidly. As the number of IoT networks and connected devices are growing, the process of identifying and controlling interconnected devices becomes gradually more challenging and it raises several security issues. To enhance the security of IoT networks, devices profiling, and identification tools became a necessity for the IoT ecosystem.

IoT profiling tools are defined by (Safi, et al., 2022) as smart tools used to identify IoT devices by analyzing network traffic and features. The usage of such tools is significant as attackers can use IoT devices as access points to the network to infect and disrupt communication. IoT identification methods are fundamental to profiling and observing associated devices' behavior in a specific network (Safi, et al., 2022). Deploying effective identification techniques can inhibit malicious activities through detecting and suspected and vulnerable devices.

2.1.3 IoT Anomaly Detection

An abnormal point, or an *anomaly*, is a data point in a modeled system that is not connected to the expected behavior. As defined by (Shen, Lin, & Zhang,

2020), anomalies are uncommon occurrences or observations that dramatically depart from expected behavior or patterns and are seen in a single data point. Figure 2.2 (Davis David ,2019) represents anomaly definition where red datapoints are anomaly standing out of the data set.



Figure(2.2): Demonstration of anomaly datapoints detected using ML (Davis David ,2019)

Anomaly Detection (AD) is an essential Artificial Intelligent (AI) technique that has been used with several applications to enhance security. AD methods and algorithms object to detect data points that depart from most of other datapoints (Songqiao Han, 2022). The usage of AD algorithms is developing when it comes to securing IoT networks. It is mainly used to identify abnormal behavior produced by affected devices.

AD in IoT context is defined by (Cook, 2019) as the process or technique used to detect significant and unexpected behavior that changes the device's normal state. A survey was conducted by (Chatterjee & Bestoun, 2022), indicated that IoT anomaly detection methods can be grouped into 4 main groups. They are grouped by method type, by usage, by anomaly type and by latency. An overview is described in Figure 2.3 (Chatterjee & Bestoun, 2022). The first category explains which method is to be used to detect anomalies

such as ML methods or statistical methods. The second category groups abnormal objects by their application whether used as malicious anomalies or for data cleaning purposes. There are three types of anomalies that can be used to categories them which are contextual, collective and point anomalies. The last category is to group IoT anomalies by latency. Latency is whether the detection approach is executed online or offline.

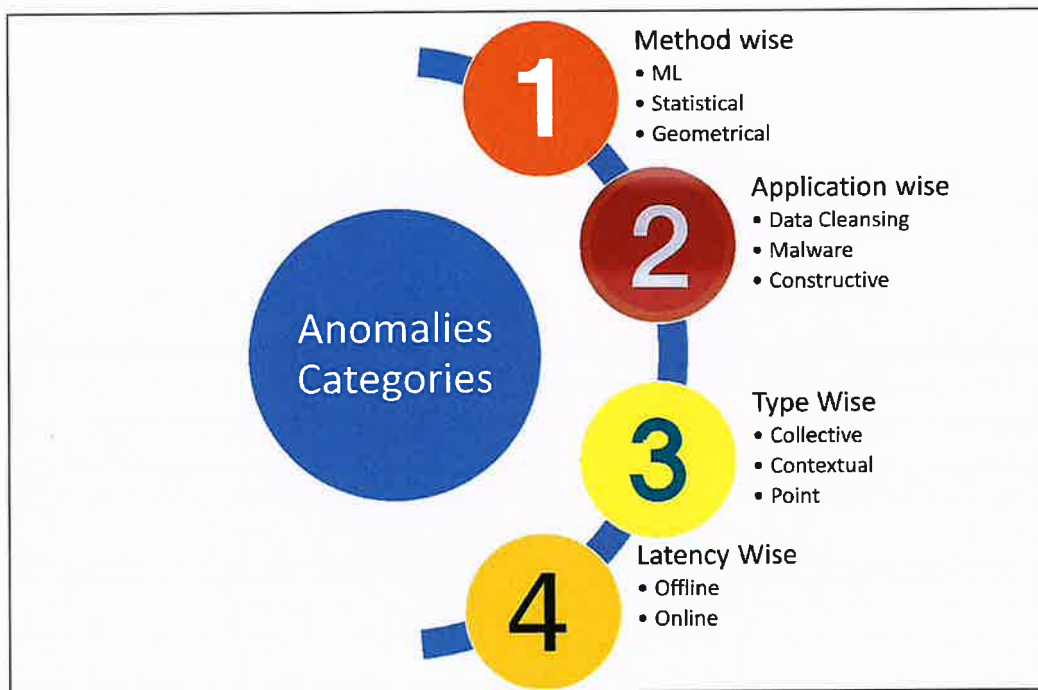


Figure 2.3): Categories of Anomalies in IoT Networks (Chatterjee & Bestoun, 2022)

The importance of AD in IoT networks comes from the urgent need for detecting unusual behavior between the devices. For instance, a failing device can be detected through capturing anomalies in data sensors, while abnormal behavior in network traffic could imply a cyber-attack (Eleonora Achiluzzi, 2022). Fraud detection and unnormal transaction can be detected by strong AD models.

2.2 Machine Learning Algorithms

Machine learning (ML) algorithms have proved their impact and effectiveness in solving intricate problems. ML uses can vary from simple task such as searching on websites and can reach very complex tasks such face recognition (Rebala, Ravi, & Churiwala, 2019) .ML refers to specific techniques used to enable computer systems or devices to perform certain tasks without the need for explicit programming. As explained in Figure 2.4, ML algorithms and models are divided into three groups based on their framework and the way of functioning. The first category is supervised learning, where the objects are labeled, and the algorithm should be able to “learn” to classify objects based on features and labels. The second type is unsupervised algorithms where the model is fed with data that contains no label and the model should be able to understand the relationship and group the data based on that understating. The last type is called semi-supervised learning in which the model is trained using a small part of the labels to perform classification or detection.

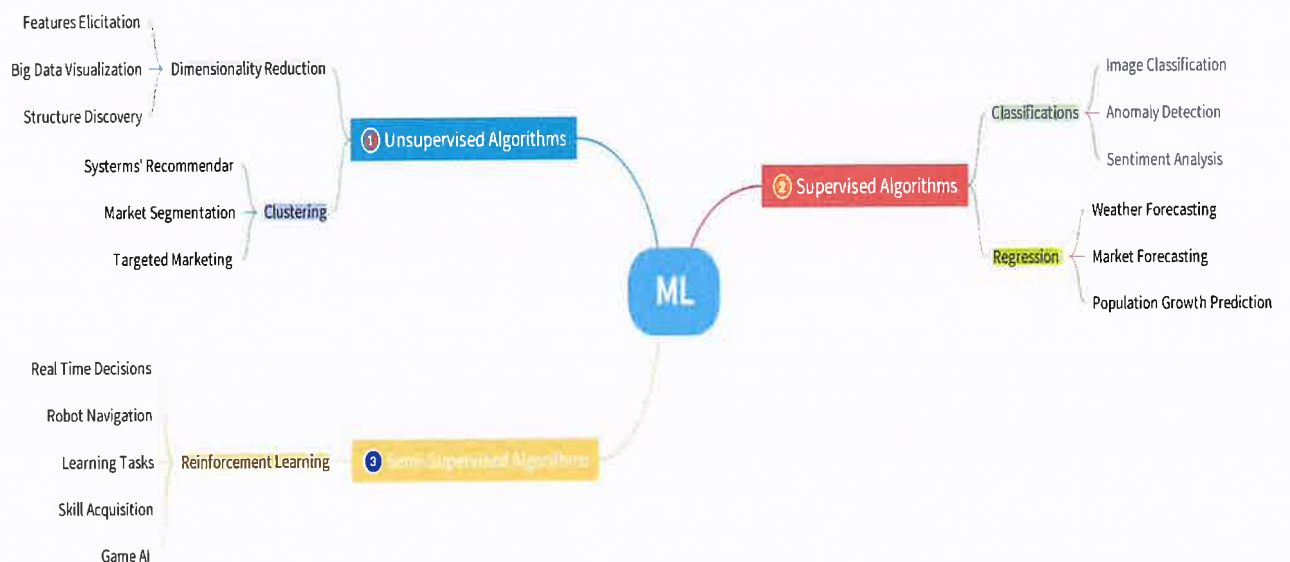


Figure 2.4): Types of ML algorithms (Abdul Wahid, 2017)

2.2.1 IoT Devices Identification Using ML

IoT devices identification and profiling is an emerging study area. Researchers deployed ML algorithms to identify IoT devices in three main ways: first method through network traffic analysis, second method is through device type identification and the last method focused on unseen devices identification. This section contains a review of ML algorithms used to identify IoT devices. The review is done for the significant publications that are related to this research.

Several research focused on network traffic analysis to identify IoT devices by investigating the effectiveness of different ML algorithms. Researchers (Nesrine A, 2020) presented a near-real time classification model to build IoT autonomous devices identification. They directed their effort in traffic flows and packets payloads analysis. Various ML algorithms were investigated, and results indicated that Decision Tree (DT) based algorithms had the best performance with 97% accuracy. Same for research conducted by (Yair Meidan, 2017), network traffic was analyzed to build profiles for IoT devices using ML algorithms. The contribution of this research is shown in the fact that they build an authentic dataset. To build the dataset, data was collected and labelled from 13 different IoT devices, computers, and smart mobile phones. The captures were labelled to apply supervised ML learning. Four ML algorithms were trained over two stages of training, and XGBoost classifier successfully identified IoT devices from non-IoT devices scoring 99% accuracy. Usage of Packets' information analysis to identify IoT devices was the focus of (Yousefnezhad, 2021) research. The research discussed the weakness of MAC and IP address as identifiers from IoT devices. That is why they depended on usage of Packets information and made the experiment

under normal and abnormal situations. Among the four examined ML algorithms, SVM had the superior accuracy and F1-score with 92%. According to (Linna Fan, 2020), supervised learning algorithms can be time-consuming due to the need for a large amount of label data. Based on that discussion, they trained semi-supervised ML model to identify IoT devices through network traffic analysis. A strong identification model was built and successfully classified 99% of IoT devices using only 5% of the labels.

In the past years, researchers have taken into consideration the need to detect IoT devices based on device type; not only if it is an IoT device or not. For any model to build clear device profiles, it should treat IoT devices differently based on their type and characteristics. For instance, home automation devices should be handled differently than audio devices. Authors in (Sivanathan A. S., 2017) proposed a ML classification model to identify IoT devices based on devices specific feature, network traffic and device normal mode. For three weeks, they collected data by tracing more than 20 different smart devices. Depending on statistical attributes, the used algorithm was able to recognize IoT devices with 95% accuracy. In the same context, (Bruhadeshwar Bezawada, 2018) conducted research in classification and identification of IoT devices based on device type. The research aimed to present an identification approach to perform IoT devices fingerprinting based on device behavior. Different ML algorithms were trained using public datasets to distinguish IoT devices from each other and from non-IoT devices as well. The challenge of the proposed model was to identify devices type even if they have similar behavior. They reported classification rate of 99% accuracy using gradient boosting algorithm. (Gupta K. , 2019) focused on the concept of re-authentication of IoT devices. Re-authentication mechanism is used

continuously to identify any IoT device as soon as it connects to a certain network. This technique is important to enhance the security of IoT networks. A ML model was built to classify IoT devices into two ways: first IoT or non-IoT devices and second based on device type. The model was able to classify and identify different types of IoT devices with an accuracy of 95%.

One major aspect of IoT devices identification is identifying unknown or unseen devices. Detecting unknown devices is an essential task to enhance IoT network privacy and security. As the number of IoT devices grows, it is important to use intelligent techniques to prevent unauthorized devices from compromising network safety and protect user information. (Meidan, 2017) examined RF classifier to classify IoT devices based on device type. This was done by creating a whitelist that contained all connected IoT devices. The model was trained against the whitelist and if a device was not listed, then it is classified automatically as an unknown device. The whitelist was created by extracting devices features from network traffic. The used dataset contained 17 different IoT devices categorized in 9 categories. The model was able to detect 90% of unknown devices and classify 99% of known devices correctly.

In the same field of study, (Vijayanand Thangavelu, 2019) proposed an advance identification system based on unsupervised ML algorithm. A model called Distributed Fingerprinting Technique (DEFT) was developed to utilize the power of clustering to automatically identify new devices. DEFT is a fingerprinting technique that is mainly used to detect the presence of common and unknown IoT devices. DEFT is a distributed technique which was developed to work in static and dynamic environments. As described in Figure 2.5, DEFT model consisted of two parts. The first part is the gateway where

feature extraction is performed while the second part is the controller that handles the classification process. Based on applied experiments, DEFT achieved high F1-score, and accuracy.

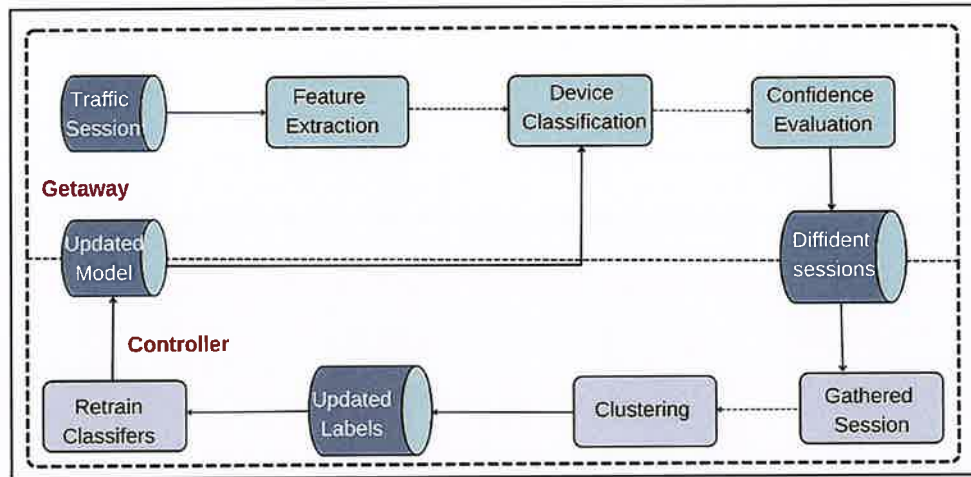


Figure (2.5): Components of DEFT Model (Vijayanand Thangavelu, 2019)

An automated IoT devices identification was proposed by (Hamad, Zhang Sheng, & Nepal, 2019). The suggested method was developed based on ML concepts to automatically detect connected IoT devices and their instances. A security system architecture was provided to implement rules to prevent unknown IoT devices connections. Unknown or unseen devices were located and blocked after applying deep monitoring session. Authors proved the effectiveness of the proposed model through measuring its accuracy and using other evaluation matrices. The model was able to identify 90.3% of device types which is higher by 9.3% compared to other proposed state-of-the-art techniques.

2.2.2 IoT Anomaly Detection Using ML

IoT anomaly detection is a challenging task due to IoT devices architecture, nature and expanding usage. To overcome these challenges, several researchers investigated the effectiveness of applying ML algorithms for IoT anomaly detection. In this section, ML anomaly detection techniques are

categorized into two types: classification-based and clustering-bases methods. For each type, some of the latest studies are introduced.

Most of the scientific research that has been conducted in the field of IoT anomaly detection aimed to apply ML techniques to classify anomalies in network communication or device behavior. A recent study conducted by (Saran & Kesswani, 2023) investigated six ML algorithms to build multiclass Intrusion Detection System (IDS) using multiple models. All models were trained and tested using a public dataset called MQTT-IoT-IDS2020. Using a simulated MQTT network architecture, the dataset was produced. It contained many features and multiple attacks classes. Figure 2.6 presents a comparison between the results produced by each classifier. Both RF and DT classifiers scored the highest in all used evaluation matrices.

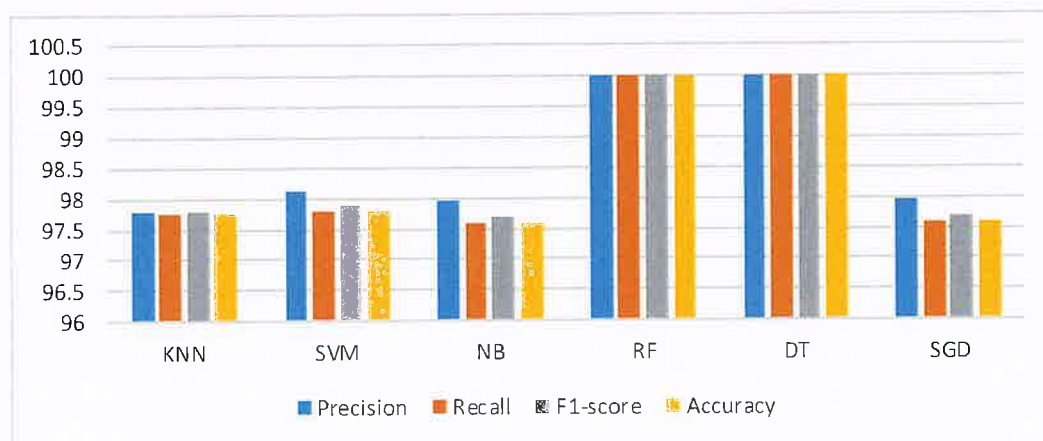


Figure (2.6): Results of ML classifiers on MQTT-IoT-IDS2020 dataset

Malware classification dataset named IoT-23 dataset was investigated by (Vitorino, Andrade, Praça, Sousa, & Maia, 2022). Nine malware captures were subjected to a deep analysis of supervised, unsupervised, and reinforcement ML algorithms. Also, binary, and multi-class classification scenarios were considered. The algorithms used were SVM, XGBoost and LightGBM for supervised learning, iForest and Local Outlier Factor (LOF)

algorithm for unsupervised learning. Figure 2.7 presented a comparison of F1-score for all six classifiers in both binary and multiclass evaluation. As the figure indicated, the best performance was for supervised ML algorithms and the lowest was from unsupervised learning algorithms. Authors concluded that more research should be conducted in utilizing unsupervised learning algorithms. Also, the DRL model, which is reinforcement learning model, did perform well which indicates that if the model was fed with more data that will enhance the performance to reach supervised learning algorithms' scores.

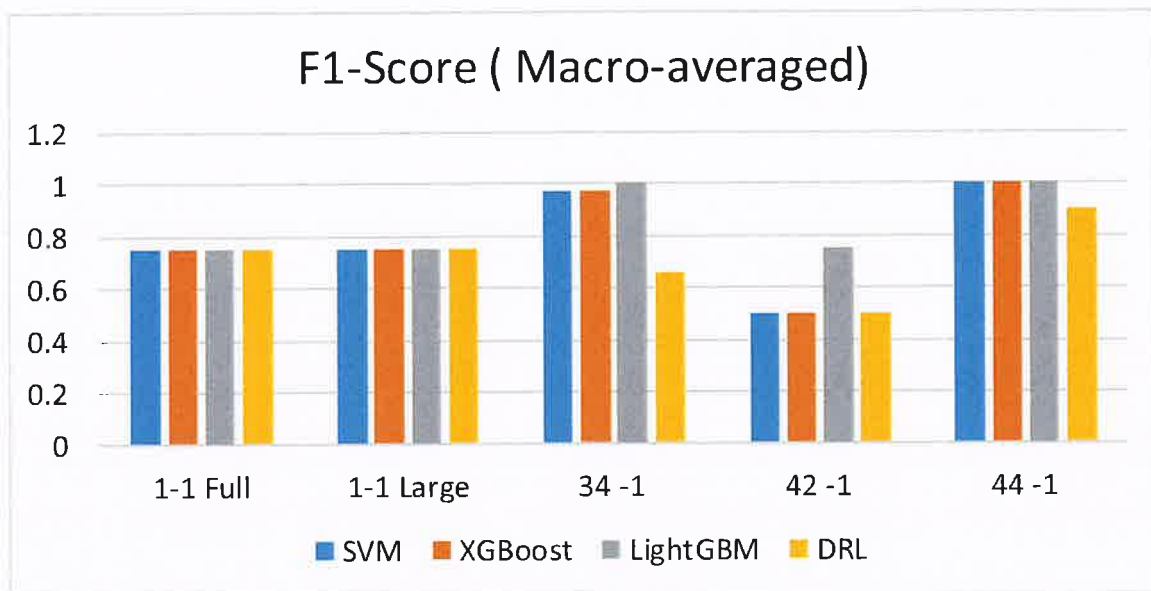


Figure 2.7): F1-score for multiclass classification (Vitorino, Andrade, Praça, Sousa, & Maia, 2022)

Several articles suggested using clustering methods and algorithms to overcome the weakness of classification algorithms and address security issues in IoT networks. In the work of (Valrorta, Martino, Cuomo, & Garlisi, 2019), authors proposed a clustering model that utilizes K-means to identify anomalies in IoT network. The focus of the research was to capture packets produced by LoRaWAN protocol to perform the clustering process. Devices were group based on network and radio behavior. The model was tested on a real network that produced more than 200,000 packets for more than 700

devices. The proposed model was able to provide remarkable performance based on used evaluation matrices.

An interesting investigation was proposed by (Fang & Li, 2018) regarding applying the K- Anonymity clustering approach. The aim of the proposed model was to enhance security and protect user privacy in IoT wearable devices. Based on K- Anonymity concept, the used dataset was split into several equal subsets with certain range of size to work in privacy-preserving manner. To evaluate model effectiveness, Euclidean distance was used to calculate how near two records were to each other. All results were plotted to determine the appropriate number of subsets clusters. Based on the result, the proposed model provided a promising future of using K- Anonymity clustering algorithm for IoT anomaly detection.

In context of applying clustering method for anomaly detection, (Pahl & Aubet, 2018) discussed the use of ML clustering algorithms to develop multi-dimensional detection model for micro-services IoT devices. The model observed the devices behavior by monitoring the internal communication. The model was able to learn through a combination of sub-models results for each micro-service IoT device in the network. The model structure depended on Virtual State Layer (VSL) that isolated traffic between IoT devices and convert them into distinct communication flows. The results of using VSL were clearly explained in Figure 2 in the research paper itself. As shown in the figure, VSL separated each communication signal into different categories based on communication source. Utilization of VSL made the clustering process more effective and efficient.

Other researchers took into consideration the need to compare the performance of classification and clustering models on specific datasets. This comparison allows better understanding of IoT devices' nature, network features and best practices for IoT anomaly detection. One example was the research conducted by (Best, Foo, & Tian, 2022). Authors proposed a hybrid model that used Naïve Bayes (NB) and K-Means (KM) models. The initial step to cluster datapoints into seven different clusters that represented types of attacks. Based on clustering result, NB classifier was trained to be able to classify new datapoints into the correct cluster or attack type. This functionality combined the accuracy of NB as supervised learning algorithm and the speed of KM as unsupervised learning algorithm. The model was trained and tested using multiple IoT datasets and had more than 90% identification accuracy in all of them.

A novel AD model was proposed by (Bobrovnikova, Lysenko, Gaj, Martynyuk, & Denysiuk, 2020). The target of the study was to present a new anomaly detection technology by analyzing DNS traffic. The model had a proactive mechanism developed based on obtaining cyberattack features from networks and building feature vectors. After feature extraction, several ML algorithms were used to perform clustering and classification. Semi-supervised fuzzy c-means clustering, SVM, and Artificial Immune System (AIS) classification techniques were tested against certain IoT datasets. According to the Authors, using the SVM produced superior outcomes than the other two approaches. (Karanfilovska, et al., 2022) compared several classification and clustering models to determine the best algorithm for anomaly detection. The proposed IDS used network traffic to classify or cluster attacks. It was trained on an IoT dataset named NF-ToN-IoT-v2. Using

supervised learning and Azure automated ML platform, XGBoost classifier reached the best results, with an F-Score of 98.8%. Followed by RF with an F-Score of 98.6%. To perform clustering, PCA was used and accomplished using PyCart automated ML platform. The clustering algorithm scored a silhouette score of 0.533.

2.3 Deep Learning Algorithms

Deep Learning (DL) (Deng & Yu, 2014) concept has been introduced since 2006 as a new field of study in ML and AI. IBM (IBM, 2020) defined DL a subgroup of ML that reflects concepts of Artificial Neural Network (ANN) with using more layers. Simulating the human brain with multiple hidden layers allows DL models to learn from data in an effective way and build the ability to handle challenging tasks. Most DL algorithms far exceed standard ML algorithms in several contexts.

As discussed by (Safi, et al., 2022), DL models are powerful tools to extract IoT devices features, identify them, and detect any anomalies. The strength of DL algorithms lies in their ability to automatically extract important features from the dataset to perform classification or clustering operations, which increases the accuracy and effectiveness of the model.

In the following sections, several studies and research papers are discussed to highlight the utilization of DL algorithms in IoT devices identification and anomaly detection.

2.3.1 IoT Devices Identification Using DL.

DL algorithms can be effectively used to identify IoT devices based on analyzing network data. (Liu, Han, & Du, 2022) proposed a model to identify

IoT devices by deploying 1D-CNN algorithm. The aim of the study was to overcome weakness of currently used passive IoT devices methods. The approach was built to analyze packets length and deal with the data in sequences. The model worked in two steps. The first step was to generate fingerprints for all devices connected to the network. The second step was to use CNN to extract deep characteristics that would identify devices effectively. The discussed model was able to detect devices type with more than 99% accuracy.

(Dupont, et al., 2021) discussed a semantic similarity-based clustering strategy to solve limits of device standard ML device identification models. Using data from 20,000 devices, the method was assessed and compared to other classification models. The results suggested that the model successfully classified significant proportion devices with 99.5% reliability.

(Kotak & Elovici, 2020) provided a clear discussion regarding the necessity of developing powerful IoT devices identification model. The need became very urgent because of the growth of cyber-attacks that target IoT devices. Additionally, they presented the weakness of current identification models that are incapable of detecting devices outside the whitelist. According to that, a DL model was proposed. To perform the classification, the model started with studying communication behavior of each device and presented the results in small images that contain network traffic details. The images were fed to multiclass classification model. The trained model was able to identify 10 different IoT devices along with smartphone and other devices with greater than 99% accuracy. The same results were achieved when the model was trained to detect unknown IoT devices.

Sivanathan et.al. (Sivanathan, Gharakheili, & Sivaraman, 2019) investigated the use of unsupervised deep clustering to identify IoT device types. The main target was to build a powerful clustering model to adopt any network changes to be able to identify existing devices or new types. A major step was to recognize essential traffic features to understand devices' characteristics, then a one-class clustering model was developed to classify each device by its network behavior. The model performance was enhanced by using noise filtering and conflict resolution. The noise filtering step which described as stage C enhanced the model's performance significantly until it reached 94% accuracy.

Generative Adversarial Network (GAN) was investigated by (Nukavarapu & Nadeem, 2021) and (Hou, Wang, Lu, Liu, & Sagduyu, 2022). Nukavarapu et. al. discussed the effectiveness of using semi-supervised GANs to identify IoT devices. The model was developed to perform multiclass classification using GANs. The training phase was done using 3% of labeled data. The classifier was able to classify IoT devices into three groups: anomaly devices, IoT devices, and non IoT devices. The classifier was able to detect 90% of new devices and classify them into the correct group. The proposed model was developed to support zero-day malware attacks, which is an important task in anomaly detection field.

Using GAN, (Hou, Wang, Lu, Liu, & Sagduyu, 2022) proposed an innovative model to test the strength of ML IoT identification models. As explained by the authors, most ML identification tools depend on data availability to be able to function. This dependency may increase the model vulnerability. That is why, the Authors developed an attack model called "IoTGAN" that worked as manipulative model to change network traffic information and fed to ML

identification models. IoTGAN worked as replacement model that fit the classification algorithm using black-box rules. The result of this experiment showed that most of the tested ML identification models were disrupted by IoTGAN which indicated that there is a necessary need to develop more security classification models that utilize DL algorithms.

2.3.2 IoT Anomaly Detection Using Deep Learning

For the past few years, DL algorithms have been used heavily for anomaly detection in IoT networks to protect the devices and end users' privacy and security. In this context, research conducted by (Kozik, Pawlicki, & Choraś, 2021) studied the effectiveness of using Feed Forward Neural Network algorithm as a classifier for IoT anomaly detection. The model was built using core anomaly detection unit, embedding vectors, encoder, and transformers. The IoT23 dataset was used to train and test the model. The proposed model was compared with other classifiers in several scenarios. F1-score was used as evaluation matrix and all results were plotted in one graph as shown in Figure 2.8. As results indicated, the suggested method had the highest F1-score in all scenarios which prove the effectiveness of DL algorithms for anomaly detection in IoT networks or environments.

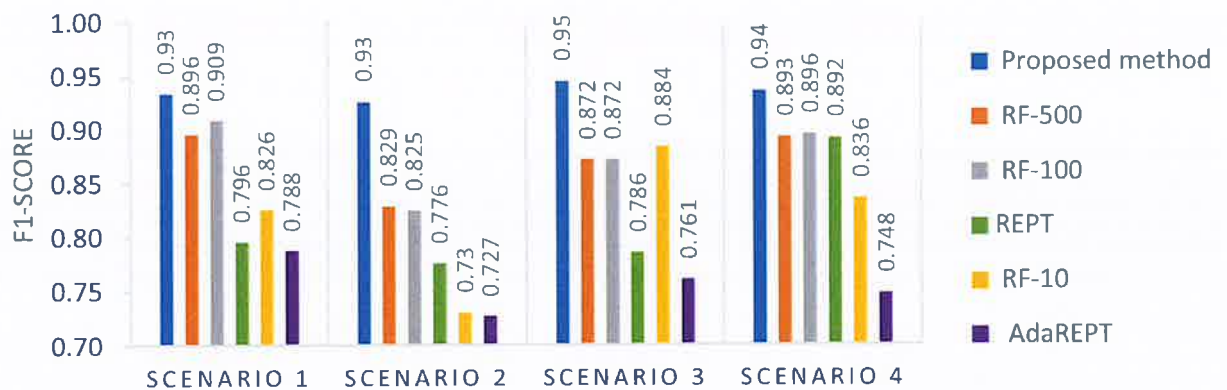


Figure 2.8):F1-score for proposed method compared to other classifiers. (Kozik, Pawlicki, & Choraś, 2021))

Several research studies were conducted to combine the power of CNN and LSTM algorithms to detect anomalies in IoT network. An anomaly detection model was suggested by (Sahu, Sharma, Tanveer, & Raja, 2021) to detect malware from 20 Raspberry Pi hacked IoT devices. The proposed model consists of two layers. The first layer represented CNN model that was used for feature extraction. LSTM algorithm worked in the second layer to perform classification and detection of abnormal datapoints. The combination of CNN and LSTM allow the model to work as a novel detection technique with 96% accuracy. Another hybrid model was proposed by (Chen, Xiao, & Jin, 2021). The presented model employed both CNN and LSTM algorithms to detect anomalies from system logs files. As illustrated by the authors, the model included two sections: log vectorization section and anomaly detection model which used both CNN and LSTM to perform binary classification task.

LSTM algorithm had been used in hybrid detection models with other algorithms than CNN. (Dutta, Chora's, Pawlicki, & Kozik, 2020) presented a joint model that managed two DL models: DNN and LSTM along with Logistic Regression (LR) ML detection algorithm. All three models were combined using stacked generalization concept. A two-step procedure was applied by the model to detect network anomalies. The first step focused on preprocessing the data to prepare it for training. The preprocessing phase was effectively done using a deep sparse autoencoder. Secondly, classification phase was done by training both models on heterogeneous datasets. The datasets were used to train each algorithm separately than combined to address the effectiveness of the proposed model. As shown in Figure 2.12, the stacked algorithm was able to detect 99% of data samples.

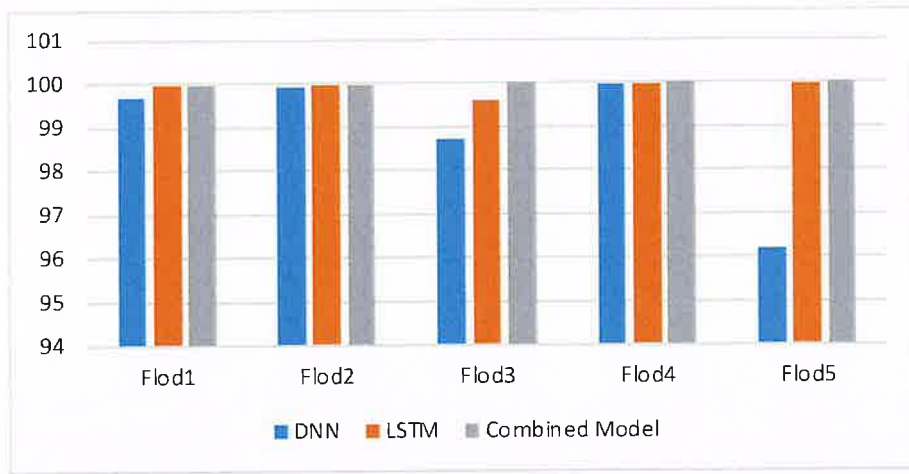


Figure (2.9): Accuracy scores comparison between 3 models

(Weng et al. ,2022) presented a new idea of utilizing Graph Neural Network (GNN) to develop IoT anomaly detection system. Authors proposed a model called E- GraphSAGE which defined as a GNN model that admits feature capturing and presenting network dataset as graphs. Based on their claim, the proposed model was the first successful GNN model for anomaly detection. E- GraphSAGE was trained and tested using advanced IDS datasets. The results showed that E- GraphSAGE performed more effectively. Authors recommend studying GNN as a new field of ML anomaly detection.

IoT anomaly detection and device identification has been present in cyber-security practice for a significant time. Most of the literature used machine learning classification algorithms to perform IoT devices profiling and detected anomalies. Other researchers studied using deep learning algorithms to classify IoT devices based on device types. Also, clustering algorithms, such as K-Means, were used mainly for anomaly detection by analyzing IoT network traffic.

Despite all studies carried out, there is still room for improvement of implemented techniques. Classification algorithms need labeled data and that is not practical for detecting new attacks or devices. This research intends to explore the effectiveness of deep learning clustering algorithms to identify

IoT devices and network anomaly detection. The major contribution of this research is to deploy several state-of-the-art clustering algorithms that have not been examined before for anomaly detection and IoT devices profiling. Through this research, experiments results are compared against the ground truth. The usage of labeled datasets gives a clear picture of algorithms performance and accuracy.

2.4 Summary

This chapter stated recent studies that discussed the usage of using ML and DL for IoT anomaly detection and device identification. Most of the studies in the literature focused on supervised classification algorithms, with the topmost used algorithms being Random Forest (RF), Decision Tree (DT), Artificial Neural Networks (ANN), and Support Vector Machines (SVM).

Many of the studies use data collected from IoT devices, such as sensor data or network traffic data, to train machine learning models for anomaly detection and device identification. Deep learning techniques, such as Convolutional Neural Networks (CNNs), Graph Neural Network (GNN) and Recurrent Neural Networks (RNNs) have also been used in some studies, with promising results.

Some studies have also explored the use of unsupervised learning algorithms, such as k-means clustering and principal component analysis (PCA), for anomaly detection and device identification in IoT networks. However, these techniques have limitations in terms of accuracy and scalability.

Overall, the literature suggests that machine learning and deep learning techniques have great potential for improving IoT security through anomaly detection and device identification. However, there are still challenges to be

addressed, such as the need for large amounts of labeled data, the interpretability of models, and the scalability of algorithms to handle the large volumes of data generated by IoT devices.

Chapter Three

Research Methodology

The conducted review of the existing research done in the field of IoT devices identification and anomaly detection has evidently identified the lack of utilizing unsupervised deep clustering to enhance the security of any IoT network and specify the needed future work ideas. The objective of this chapter is to explain in detail the methodology, including the used datasets, to address the effective solutions for research problems and accomplish research objectives.

3.1 Projected Methodology

Several steps were followed as a research methodology to achieve research objective. The initial step was to select the appropriate datasets. Selecting IoT dataset is crucial to achieve accurate results. The dataset should contain a diverse range of IoT devices and should be large enough to support the analysis of network communication patterns. The second step involves performing data exploration and feature selection. This process can help to identify the most important features for clustering algorithms, and can reduce the dimensionality of the dataset, which can improve the efficiency of the clustering algorithms.

The third step involves building deep clustering models to identify IoT devices. In this case, the deep clustering models are trained on the selected features from the dataset, with the aim of identifying clusters of IoT devices that share similar communication patterns.

The fourth step involves investigating different architectures and methods to create IoT device profiles from the identified clusters. This implies analyzing the communication patterns of the devices within each cluster and creating profiles that capture the unique characteristics of each cluster. This can help to improve the accuracy of device identification and can make it easier to detect anomalous communications in the future.

Finally, the fifth step involves evaluating the proposed model using the selected dataset. This involves measuring the accuracy and generalization capabilities of the model and comparing it to other clustering algorithms and models. This step is crucial to ensure that the proposed model is effective and can be applied to other datasets and scenarios.

Overall, the methodology followed to conduct the proposed research involves a series of well-defined steps that are designed to ensure the accuracy and effectiveness of the proposed model. By following this methodology, the proposed model can help to improve the identification of IoT devices and detect anomalous communications in a more effective manner, which can have significant implications for the security and reliability of IoT networks.

3.2 Datasets

The proposed models were trained and tested using two public IoT datasets. The first dataset was used to train ML algorithms to detect and identify devices. The second dataset was used to train deep clustering algorithms for anomaly detection. A full description of both datasets is provided in the following subsections.

There are several reasons behind selecting the used datasets. The two datasets selected are the CIC-IoT-2022 and IoT-23 datasets. The first reason for

selecting the CIC-IoT-2022 dataset is that it was specifically developed to create an automated IoT identification model, which is the goal of the project. This means that the dataset is well-suited for the problem at hand and is likely to provide useful insights into how to develop an effective identification model. The second reason for selecting the CIC-IoT-2022 dataset is that it is a very recent dataset that has not been used in the literature yet. This indicates that the project has the potential to contribute new knowledge to the field. Additionally, the dataset is updated on a regular basis, which means that the project can take advantage of new data as it becomes available.

The second dataset selected is the IoT-23 dataset, which is a well-known dataset for malware detection in an IoT network. While this dataset has been used by other researchers for classification projects, it has not been used for clustering. This means that there is potential for the to provide insights into how clustering can be used to detect malware in an IoT network. Additionally, both datasets are a labeled dataset with enough samples and labels to perform reliable training and testing. This implies that the project can take advantage of the labeled data to improve the accuracy and generalization capabilities of the algorithms being used. Overall, the selection of these two datasets for the project is likely to provide useful insights into how to develop effective deep clustering algorithms for IoT identification and malware detection.

3.2.1 CIC-IoT-2022 Dataset Description

CIC-IoT-2022 dataset was created by (Dadkhah, et al., 2022) at the Canadian Institute for Cybersecurity (CIC). The dataset was developed in December 2022 to be used as a realistic reference for IoT devices identification and profiling. The data was collected from real IoT environment in CIC lab through analyzing network communication of 60 different IoT devices. Each

device had been examined under four different situations: idle, active, interaction and powered on. Moreover, network activities were captured using multiple scenarios. All collected data was modified, collected, and grouped to represent 28 types of devices under 3 main categories: Home Automation, Camera, and Audio devices. Table 3.1 presents features extracted for IoT devices. The experiment results in extracting 48 features representing network traffic that distinguish each device. Features were selected to describe IP packet information and provide statistical information about the packets. Feature were specified based on their efficiency to distinguish IoT device types.

Table (3.1): CIC-IoT-2022 Dataset Features

No	Feature	Data type	Description
1	TCP Protocol	int64	Used control protocol to connect devices
2	UDP Protocol	int64	Used datagram protocol to start connectionless communication
3	HTTP Protocol	int64	Used Hypertext transfer protocol
4	HTTPS Protocol	int64	Used Secure hypertext transfer protocol
5	Padding of IP	int64	Modified packet size length as multiplication of 32
6	Alert for Router	int64	Warning for user to study sent packets of IP
7	Class of Source Port	int64	4 port classes: 0 for none -1 for known ports – 2 for registered ports = 3 for dynamic ports
8	Class of Destination Source	int64	
9	Packet size	int64	Total size of sent packets (in bytes)

10	Transformed data	int64	Raw data without processing
11	Destination Address	int64	IP address of destination device
12	Frame Size	int64	Ethernet frame type
13	Frame Type	int64	Ethernet frame size
14	Time to Live	int64	Packet's lifetime
15	Final Packet Size	int64	Size of packets
16	Port of Source	int64	Protocols used by different layers
17	Port of Destination	int64	Device's destination port
18	Count of DNS	int64	DNS's number of requests
19	Count of NTP	int64	NTP's number of requests
20	Count of ARP	int64	ARP's number of requests
21	Count of Protocols	int64	Total number of used protocols
22	Count of Destination IPs	int64	Total number of IP address accessed by destination device
23	Frequent Destination IPs	int64	Most frequently visited IP address
24	Frequent Destination Protocol	int64	Most frequently used protocols
25	Frequent Source IPs	int64	Most frequently used ports
26	Frequent Destination port	int64	Most frequently accessed port
27	Total Packet Size	int64	Data aggregation result
28	Min Packet	float64	Minimum size of 20 packets
29	Max Packet	float64	Maximum size of 20 packets
30	Med Packet	float64	Median size of 20 packets
31	Avg Packet	float64	Average size of 20 packets

32	Variance Packet	float64	Spread measurement between 20 packets.
33	3 rd Quartile packet	float64	Calculated center point between the median and maximum
34	1 st Quartile Packet	float64	Calculated center point between the median and minimum
35	Inner Quartile Packet	float64	Spread amount for 50% of packets
36	Frame total	int64	Recording timestamp
37	Ethernet min.	int64	Minimum size of 20 frame
38	Ethernet max.	int64	Maximum size of 20 frame
39	Frame med.	float64	Median size of 20 frame
40	Frame Avg.	float64	Average size of 20 frame
41	Frame variance	float64	Spread measurement between 20 frames.
42	3 rd Quartile Frame	float64	Calculated center point between the median and maximum
43	1 st Quartile Ethernet	float64	Calculated center point between the median and minimum
44	Inner Quartile Frame	float64	Spread amount for 50% of packets
45	Timestamp of epoch	float64	Recording timestamp
46	Time of arrival	float64	Time difference between previous and current packet
47	time frame from last frame	float64	Time difference between first and current packet
48	Gabel category	Object	Classification label (0 = Camera / 1 = Home Automation / 2 = Audio)

Devices in CIC-IoT-2022 dataset are labelled using two categories: by device type or by global category. Number of instances for each device in the dataset is shown in Table 3.2. As shown, there is a small difference between label distribution which is not an issue because dataset does not need to be very symmetrical. As shown in Figure 3.1, most of devices fell in Camera category with a big difference comparing to other datapoints.

Table (3.2): Labels Distribution in CIC-IoT-2022 dataset

No	Label	Datapoint Counts
1	Amazon plug	1168
2	Amcrest cam	36724
3	Arlobasestation cam	41040
4	Arloq cam	42100
5	Atomi coffeemaker	826
6	Borun cam	4404
7	Dlink cam	28198
8	Echo dot1	1958
9	Echo dot2	3134
10	Echo spot	5646
11	Echo studio	6146
12	Eufy homebase	11007
13	Globel amp	908
14	Heim vision cam	13787
15	Heim vision lamp	3813
16	Home eye cam	27619
17	Luohe cam	20076
18	Nest cam	7490
19	Nest mini	3269
20	Netat mocam	10625
21	philipshue	919
22	Room ba	1106
23	Sim cam	34217
24	smartboard	116
25	sonos	6569
26	Sonos one	6569

27	teckin1	469
28	yutron2	234

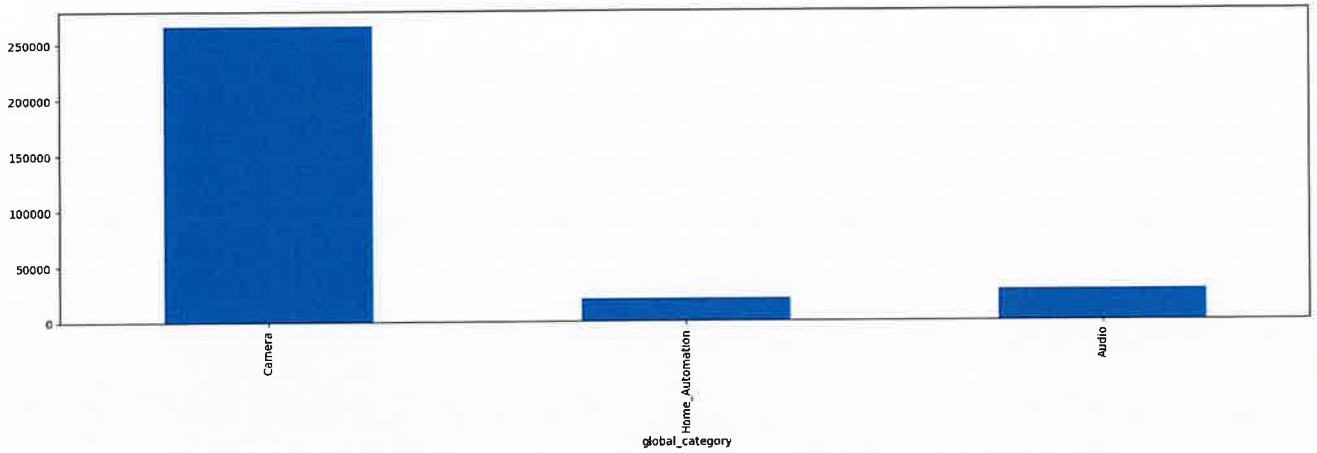


Figure 3.1): Label distribution among global category feature in CIC dataset

CIC-IoT-2022 dataset is a recent IoT fingerprinting dataset in which it updated on a regular basis by the authors and has not been used in any experiments except for a classification model conducted by the CIC-IoT-2022 developer. It is a powerful tool to build strong identification model to identify IoT using network communication data and devices specifications.

3.2.2 IoT23 Dataset Description

IoT23 (Garcia, Aposemat IoT-23, 2020) is well known labeled dataset for malware and anomaly detection in IoT network. It was built by (Garcia, Parmisano, & Erquiaga, 2020). IoT23 was captured from 2018 to 2019 and published in 2020. The dataset reflects large IoT attacks and benign traffic real-time data to be used in IoT anomaly detection research and projects. The dataset contains 3 benign IoT traffic and 20 malware captures from different devices. Captured scenarios are divided into several PCAP files from attacked IoT devices and saved to reflect the attack name. The same process was done

for benign captures. Raspberry Pi device was used to execute several actions to initiate attacks under different protocols.

IoT23 dataset is a labeled dataset that contains 21 features. As shown in Figure 3.1, features reflect network traffic information that differentiate each capture and attack type. All features have numeric values except for the attack label feature.

Attribute Number	Features	Description	Attribute Number	Features	Description
1	fields-ts	Flow start time	10	orig-bytes	Source2destination transaction bytes
2	uid	Unique ID	11	resp-bytes	Destination2source transaction bytes
3	id.orig-h	Source IP address	12	conn-state	Connection state
4	id.orig-p	Source port	13	local-orig	Source local address
5	id.resp-h	Destination IP address	14	local-resp	Destination local address
6	id.resp-p	Destination port	15	missed-bytes	Missing bytes during transaction
7	proto	Transaction protocol	16	history orig-pkts	History of source packets
8	service	http, ftp, smtp, ssh, dns, etc.	17	orig-ip-bytes	Flow of source bytes
9	duration	Record total duration	18	resp-pkts	Destination packets
			19	resp-ip-bytes	Flow of destination bytes
			20	tunnel-parents	Traffic tunnel
			21	label	Attack label

Figure (3.2): IoT23 Features (Dutta, Choraś, Pawlicki, & Kozik, 2020)

Labels distribution in IoT23 dataset is shown on Table 3.3. The dataset has unbalanced data since there are some attacks with very low instances. In this research, the issue of unbalanced data is solved by performing dawn-sampling. Dawn-sampling (Agarwal, 2019) is a data preprocessing technique or task which is performed by removing subsets with low records number comparing to labels' majority.

Table(3.3): Labels Distribution in IoT23 dataset

No	Label	Datapoint Counts
1	PartOfAHorizontalPortScan	676033
2	Okiru	237732
3	Benign	199641
4	DDoS	113818
5	C&C	15063
6	Attack	3915
7	C&C-HeartBeat	337
8	C&C-FileDownload	43
9	C&C-Torii	30
10	FileDownload	10
11	C&C-Mirai	1

IoT23 dataset have been used heavily in several research in IoT anomaly detection field. Figure 3.3 presents statistical information about different research conducted using IoT23 dataset. Since 2020, the dataset has been trained and tested in more than 20 research papers and 5 ML theses. This heavy utilization of the dataset indicates how effective and efficient the dataset is and to which extent it can help improving AD for IoT devices models and algorithms.

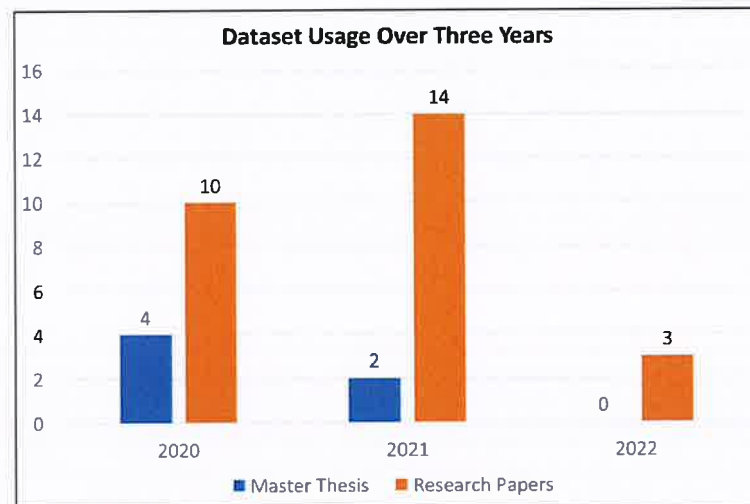


Figure 3.3): IoT23 dataset employment from 2020 - 2022

3.3 Data preprocessing

IoT datasets have a special nature and most of them are saved as PCAP files. Several steps should be followed to prepare data to be employed in ML and DL models. This section describes methodologies used for data preprocessing through this research.

3.3.1 Prepare data for Loading

Most of the data extracted from IoT network captures are saved as PCAP files which should be extracted and stored as CSV files to be loaded, manipulated, and fed to ML and DL models.

Captures in CIC-IoT-2022 dataset are stored as PCAP for each device separately. For each device, there are 5 files reflecting each scenario followed when captures were taken. All files related to certain device were saved in a separated folder and named after the device type followed by scenario name. In the research, PCAP files are converted using network software called Wireshark. Wireshark (Sharpe, Warnicke, & Lamping, 2022) is a user-friendly software to process packet files captured from any network. It is utilized to convert PCAP files related to the dataset to CSV file. Generated CSV files are organized and named in the same method as original PCAP files. Converted CSV files are loaded as Pandas data frame and the feature (label) is added to reflect the name of the device. Also, “global category” label is added to represent whether the device belongs to home automation, audio, or camera categories. The labeling is done based on information provided by dataset developers (Dadkhah, et al., 2022) in the released research paper. The final step is combining all data frames together in one CSV file to ease the process of training and testing the models and algorithms.

Captures in IoT23 dataset are read from PCAP files into conn.log and saved in sperate folders named after each label as an attack or benign action. To process conn.log files, specific python codes lines are used depending on Pandas library. Those commands are used to convert captured data and combine them into one CSV file. Figure 3.4 shows a screenshot of commands used to convert loaded captures into Pandas data frames. After conversion, labels are renamed to have unified style. For instance, some benign captures use (- Benign-) as a label and others labelled as (Benign) which affected the performance of any used classifier. Figure 3.5 shows a screenshot of the code used to unify all labels in IoT23 data frame.

```
In [26]: #convert the stored captures to pandas dataframe
data43 = pd.read_table(filepath_or_buffer=cap_43, skiprows=10, nrows=100000)
data43.columns=['ts','uid','id.orig_h','id.orig_p','id.resp_h','id.resp_p','proto','service','duration','orig_bytes',
'resp_bytes','conn_state','local_orig','local_resp','missed_bytes','history','orig_pkts','orig_ip_bytes',
'resp_pkts','resp_ip_bytes','label']
data43.drop(data43.tail(1).index,inplace=True)

In [27]: #convert the stored captures to pandas dataframe
data44 = pd.read_table(filepath_or_buffer=cap_44, skiprows=10, nrows=100000)
data44.columns=['ts','uid','id.orig_h','id.orig_p','id.resp_h','id.resp_p','proto','service','duration','orig_bytes',
'resp_bytes','conn_state','local_orig','local_resp','missed_bytes','history','orig_pkts','orig_ip_bytes',
'resp_pkts','resp_ip_bytes','label']
data44.drop(data44.tail(1).index,inplace=True)

In [28]: #convert the stored captures to pandas dataframe
data48 = pd.read_table(filepath_or_buffer=cap_48, skiprows=10, nrows=100000)
data48.columns=['ts','uid','id.orig_h','id.orig_p','id.resp_h','id.resp_p','proto','service','duration','orig_bytes',
'resp_bytes','conn_state','local_orig','local_resp','missed_bytes','history','orig_pkts','orig_ip_bytes',
'resp_pkts','resp_ip_bytes','label']
data48.drop(data48.tail(1).index,inplace=True)
```

Figure (3.4): Convert stored captures to data frame in IoT23 dataset.

```
In [61]: full.loc[(full.label == '- Malicious PartOfAHorizontalPortScan'), 'label'] = 'PartOfAHorizontalPortScan'
full.loc[(full.label == '(empty) Malicious PartOfAHorizontalPortScan'), 'label'] = 'PartOfAHorizontalPortScan'
full.loc[(full.label == '- Malicious Okiru'), 'label'] = 'Okiru'
full.loc[(full.label == '(empty) Malicious Okiru'), 'label'] = 'Okiru'
full.loc[(full.label == '- Benign -'), 'label'] = 'Benign'
full.loc[(full.label == '(empty) Benign -'), 'label'] = 'Benign'
full.loc[(full.label == '- Malicious DDoS'), 'label'] = 'DDoS'
full.loc[(full.label == '- Malicious C&C'), 'label'] = 'C&C'
full.loc[(full.label == '(empty) Malicious C&C'), 'label'] = 'C&C'
full.loc[(full.label == '- Malicious Attack'), 'label'] = 'Attack'
full.loc[(full.label == '(empty) Malicious Attack'), 'label'] = 'Attack'
full.loc[(full.label == '- Malicious C&C-HeartBeat'), 'label'] = 'C&C-HeartBeat'
full.loc[(full.label == '(empty) Malicious C&C-HeartBeat'), 'label'] = 'C&C-HeartBeat'
full.loc[(full.label == '- Malicious C&C-FileDownload'), 'label'] = 'C&C-FileDownload'
full.loc[(full.label == '- Malicious C&C-Torii'), 'label'] = 'C&C-Torii'
full.loc[(full.label == '- Malicious C&C-HeartBeat-FileDownload'), 'label'] = 'C&C-HeartBeat-FileDownload'
full.loc[(full.label == '- Malicious FileDownload'), 'label'] = 'FileDownload'
full.loc[(full.label == '- Malicious C&C-Mirai'), 'label'] = 'C&C-Mirai'
full.loc[(full.label == '- Malicious Okiru-Attack'), 'label'] = 'Okiru-Attack'

In [62]: full.loc[(full.label == '- benign -'), 'label'] = 'Benign'

In [63]: full['label'].value_counts()

Out[63]: PartOfAHorizontalPortScan    676033
Okiru    237732
Benign    199641
DDoS    113818
C&C    15063
Attack    3915
C&C-HeartBeat    337
C&C-FileDownload    43
C&C-Torii    30
C&C-HeartBeat-FileDownload    8
FileDownload    2
C&C-Mirai    1
Name: label, dtype: int64
```

Figure (3.5): re-label labels to be unified in IoT23 dataset.

Both datasets are analyzed deeply to understand the content of each one and features' correlation before applying ML and DL algorithms.

3.3.2 Features Selections

Features Selection task is an important factor in increasing the efficiency and effectiveness of any ML model. Gupta A. (2020) defines features selection as methods used in ML algorithms to recognize the relevant features to be used to train algorithms. Features selection techniques are chosen based on the type of algorithm, either supervised or unsupervised learning model.

In this study, specific features selections are used to enhance models performance depending on used datasets nature. Correlation Coefficient is employed to select features for supervised learning algorithms. Correlation (Gupta A. , 2020) is a determine of the relationship between two variables represented by straight line (linear relationship). Figure 3.6 presents of calculate correlation coefficient for CIC-IoT-2022 dataset. By using conditions, only features that have coefficient above 0.8 will be visible in the

	id	L4_tcp	L4_udp	L7_http	L7_https	port_class_src	port_class_dst	pck_size	ip_dst_new	ethernet_frame_size
id	1.000000	nan	nan	nan	nan	nan	nan	nan	nan	nan
L4_tcp	nan	1.000000	-0.989440	nan	0.939036	nan	nan	nan	nan	nan
L4_udp	nan	-0.989440	1.000000	nan	-0.929120	nan	nan	nan	nan	nan
L7_http	nan	nan	nan	1.000000	nan	nan	nan	nan	nan	nan
L7_https	nan	0.939036	-0.929120	nan	1.000000	nan	nan	nan	nan	nan
port_class_src	nan	nan	nan	nan	nan	1.000000	nan	nan	nan	nan
port_class_dst	nan	nan	nan	nan	nan	nan	1.000000	nan	nan	nan
pck_size	nan	nan	nan	nan	nan	nan	nan	1.000000	nan	0.99
ip_dst_new	nan	nan	nan	nan	nan	nan	nan	nan	1.000000	nan
ethernet_frame_size	nan	nan	nan	nan	nan	nan	nan	0.999999	nan	1.00
ttl	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
total_length	nan	nan	nan	nan	nan	nan	nan	0.999999	nan	0.99
protocol	nan	-0.976812	0.997514	nan	-0.917262	nan	nan	nan	nan	nan
source_port	nan	nan	nan	nan	nan	0.879213	nan	nan	nan	nan
dest_port	nan	nan	nan	nan	nan	nan	0.845052	nan	nan	nan
DNS_count	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
NTP_count	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
ARP_count	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
cnt	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
L3_ip_dst_count	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
most_freq_d_ip	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan

Figure (3.6): Partial view of correlation coefficient for CIC-IoT-2022 dataset

matrix, other than that the value will be replaced with nan. Based on correlation matrix, features with low correlation are eliminated.

Different features selection methods are used for unsupervised learning algorithms such as K-Means (KM) algorithm. In this research, PCA tool is used for dimensionality reduction.

Principal Component Analysis (PCA) is a dimensionality reduction used with unsupervised ML algorithms. PCA (Li, 2019) works as linear conversion that applies mathematical equations to select most related features to create new subset or version of the data frame. Data standardization is an entail step that should be performed before using PCA algorithm. PCA mainly is a feature extraction technique, but PCA concept can be utilized to help in performing feature selection in certain situations.

In this research, PCA is used to prepare data for unsupervised clustering algorithms such as KM and GMM. Also, non-numeric features were eliminated since clustering algorithms cannot handle textual data. Along with those methods, several experiments were performed to decided which features to keep and which to drop.

With deep clustering, there is no need to perform either features selection or dimensionality reduction since the encoder accomplishes this task automatically. This is an advantage of using deep learning algorithms.

After performing features selection techniques, 4 features were removed from IoT-23 devices. The remaining 18 features were kept training the models. For CIC-IoT-2022, no features were removed since all of them contain numeric values. PC technique was used to prepare the data for ML clustering

algorithms. PCA model select top 5 features to be used to train the model. This technique was applied for both datasets.

3.3.3 Features Engineering

Feature engineering (Bhandari, 2023) is a vital stage in the development of ML models. There are three main aspects of feature engineering: data scaling, data standardization, and data normalization. All aspects are used to transform data to be more suitable to use with any model changing data to make it more suited for modeling.

Data scaling is known as the technique used to convert data values to be in one scale using certain mathematical equation. It is more effective with distance sensitive algorithms such as KNN and SVM. Data scaling is performed using normalization and standardization tools. The selection between the two techniques depends on several factors such as data size and the nature of the data and the used algorithm itself. Table4 (Bhandari, 2023) contains a comparison between data normalization and data standardization. As discussed in Table 3.4 (Bhandari, 2023), the target of both techniques is to scale data to be less than 1 and datapoints to have the same range of values. Before selecting one of them, the data should be analyzed to understand the data distribution, data relationship importance, and the outliers. This analysis is the main guide to select the suitable technique to the used dataset.

In the context of this research, both standardization and normalization are used in applied ML algorithms. For KM algorithm, standardization was used to obtain the relationship between datapoints especially that the distribution of the data is unknown. For other algorithms like CNN, normalization was

deployed since we need to not change the input shape and the relationship between datapoints is not critical.

Table (3.4): Comparison between Standardization and Normalization

Comparison Aspect	Data Normalization	Data Standardization
Definition	Technique used to change the scale of datapoints values to be between 0 and 1.	Change datapoints to take the mean as centroid and scaled to 1 based on standard deviation
Best use	Best work with non-Gaussian datapoints distribution.	Best work with Gaussian datapoints distribution.
Relation with outliers	More sensitive to outliers	Less sensitive to outliers
Effect on datapoints relationships	May affect the relationship between datapoints	Keep datapoints relationship as its
Output shape	Same shape as the input (x)	Change the shape from input (x)
Mathematical Equation	$X - \text{minimum} / (\text{maximum} - \text{minimum})$	$X - \text{mean} / x \text{ standard deviation}$

3.4 Selected Algorithms

Different algorithms are examined in both CIC-IoT-2022 and IoT23 dataset to accomplish research objectives. The first step is to perform IoT device

identification and classification using CIC-IoT-2022 dataset to train and test the performance of several ML algorithms. The same dataset is used to investigate the effectiveness of unsupervised DL algorithms for IoT device identification. IoT23 dataset is used to perform attacks and IoT anomaly detection using unsupervised DL algorithms. This section reviews the main information and characteristics of the algorithms used, whether they are machine learning or deep learning algorithms.

3.4.1 Machine Learning Algorithms

i. K Nearest Neighbor (KNN)

K-nearest Neighbor (KNN) (Guo, Wang, Bell, Bi, & Greer, 2003) algorithm is a ML classification algorithm. KNN functions by understanding the relationships between datapoints to classify points into specific group of data based on the distance between datapoints. Points in KNN algorithm are classified based on which group contains the closest ‘neighbor’ points. Researchers describe KNN as lazy algorithm or learner. This means that the KNN model will not be built until a certain dataset query is performed.

The letter “K” reflects the number of nearest neighbors to be used when training the model. Selecting K has no clear and direct way to be done and it needs tuning the model until reaching the best number. As explained by (Subramanian, 2019), KNN algorithm is flexible algorithm that can handle multiclass classification. On the other hand, the computational cost of training KNN model is high referring to the need to calculate distance for all datapoints.

i. Random Forest (RF)

Random Forest (RF) is a well-known ML algorithm that can be used in regression and classification scenarios. In classification tasks, RF constructs decision trees from multiple data samples and uses their majority vote which reflect bagging principle (R, 2021).

RF has several positive features. To start with, RF is a good solution for high dimensional dataset since it does not consider all feature while testing or training the data. Moreover, with RF there is no need to split training data from testing data because the algorithm oversees 30% of the data. RF is a very stable algorithm since it works under voting principle. Figure 3.7 presents an illustration of how RF functions.

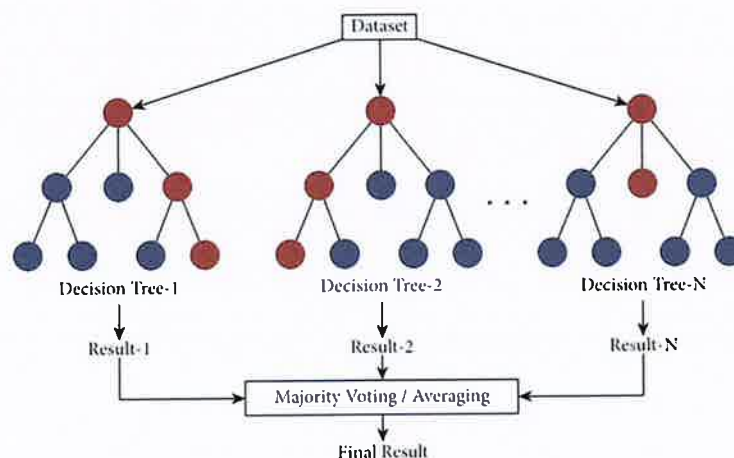


Figure 3.7): How FR operates (Khan, et al., 2021)

ii. Decision Tree (DT)

Decision Tree (DT) is an ML algorithm used to make predictions or classification. DT (Saini, 2021) works using a tree shape model that contains all decisions and possible results. The algorithm functions by separating the data into subgroups based on the most important feature.

Several ML algorithms work using DT concepts. One of them is RF algorithm. The differences between DT and RF are shown in Figure 3.8 (Saini, 2021). As the figure indicates, the main difference between RF and DT algorithms is that RF model produced decisions as a set based on the output. In contrast, DT presents all possible outcomes as tree's branches. This difference makes DT computationally more expensive and slower than RF.

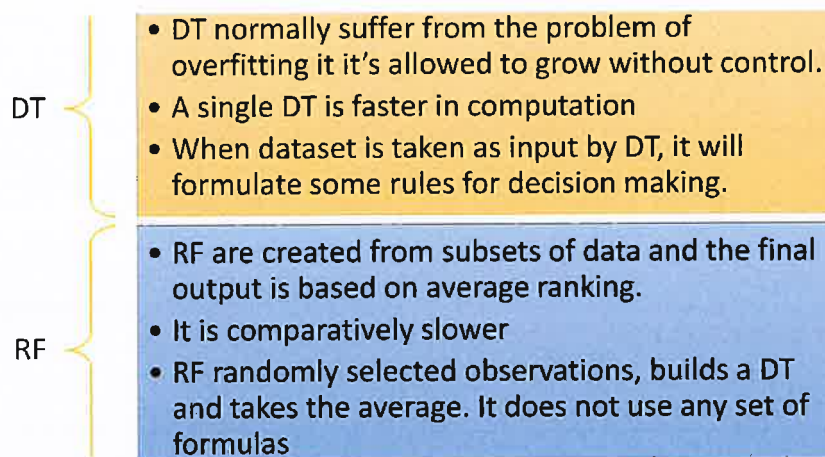


Figure (3.8): Random Forest Vs. Decision Tree (Saini, 2021)

iii. Artificial Neural Network (ANN)

Artificial Neural Networks (ANN) are ML algorithms that simulate how the human brain works (Singh, 2021) . The ANN algorithm can be used for regression and classification. ANN models consist of several layers that contain the nodes and weights. The first layer is the input layer that received information from the dataset. Then the model process received input using the hidden layers. As the number of hidden increases, the complexity of the model increases and is converted to be deep learning model. The last layer is the output layer which either produces one value in case of regression and binary classification or multiple outputs if the network is designed for multiclass classifications. Each layer in ANN model has an activation function that should be specified to allow the layer to perform properly.

Several deep learning models are built using the concept of ANN such as RNN, CNN and LSTM. The strength of ANN lies in its ability to perform complex tasks and handles non-linear data.

iv. Density-Based Clustering Algorithm (DBSCAN)

Density-Based Clustering Algorithm (DBSCAN) (Yıldırım, 2020) is a unsupervised ML clustering algorithm that works based on density level between data groups. DBSCAN identifies clusters as regions of high density or mass. Two important values should be assigned in the initial state of DBSCAN which are the epsilon and the minimum point. The model starts by randomly selecting any datapoint and assigning it to a cluster. Then it calculates number of datapoints located on the distance between the points and X, if the count is less than the initial minimum point, it assigns the other datapoints to the same cluster. DBSCAN is more complicated and computationally expensive than K-means algorithm.

v. K-Means Clustering (KM)

K-Means (KM) is an unsupervised ML algorithm used for clustering purposes. KM (Sharma P. , 2019) is a distance-oriented algorithm where it calculates the distances between a point and the center of a cluster, and a point is assigned to a certain cluster if the distance reached the minimum. The center of a cluster is called centroid and it is calculated by the algorithm itself.

In KM, k refers to number of clusters and it can be selected or defined using several techniques like Silhouette Score. The silhouette score (Sharma P. , 2019) is a value that shows datapoints are assigned to the correct cluster. The best silhouette score is the closest to 1 which indicate that points are assigned to the correct cluster and there is no overlapping between clusters. In other

hand, if silhouette score is closer to 0 it means that there are errors and datapoints are assigned to wrong clusters. If silhouette score has negative value, this means having overlapped clusters with no real boundaries.

3.4.2 Deep Learning Algorithms

i. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a deep learning algorithm that reflects concepts of Deep Neural Networks (DNN). CNN (Mandal, 2021) was developed to overcome difficulties in images classification. The basic idea behind CNN is to have a convolutional layer to reduce input dimensions without data leak or information lost. Using convolutional layer technique creates powerful image classification models.

The main difference between ANN and CNN is having three types of layers that must be used when building CNN model. Other than convolutional layer, any CNN model should have Pooling layer and fully connected layer. The architecture of CNN gives the model the ability to classify both image and non-image data. CNN can perform effectively when trained on audio or network datasets.

ii. Dynamic Auto-Encoder (DynAE)

Dynamic Auto-Encoder (DynAE) is an original deep clustering model that was developed by (Nairouz, Mefraz, Ksantini, & Zied, 2020) in 2020. This model was developed to overcome the weakness of ML clustering algorithms. The DynAE model performs clustering effectively because it focusses on automatic feature selection and dimensionality reduction. DynAE was tested to benchmark datasets such as MINST dataset.

DynAE model was develop based on two main Python libraries which are Pytourtch and Keras (TensorFlow). Pretraining concept is deployed to enhance the performance of the model. After the pertain phase is completed, the model starts the clustering phase. DynAE is used to cluster image datasets and as the base of new model called ConvDynAE proposed by (Mohamed, Ksantini, & Kaabi, 2022) to solve the issue of clustering abundantly semantic image datasets.

iii. Deep Embedded Clustering (DEC)

Deep Embedded Clustering (DEC) was developed in 2015 by (Xie, Girshick, & Farhadi, 2015) as a new deep clustering model. DEC functions based on learning and training it gets from feature representations then cluster points using DNN. The functionality of DEC allows the model to learn in a low space because of dimensionality reduction which enhances the clustering process. DEC has been tested against well-known datasets and reports excellent results.

iv. Deep Clustering Network (DCN)

Deep Clustering Network (DCN) is a novel model develop by (Yang, Fu, Sidiropoulos, & Hong, 2017) . DCN is a deep clustering model built using KM and DNN algorithms. The idea behind DCN is to perform KM clustering using autoencoder and the strong features of DNN. The DCN model has been tested by the developer against images dataset and reported an effective performance compared to ML clustering algorithms. DCN functions based on clusters created by kmeans algorithm then it works around the results to enhance the performance using Deep Neural Network (DNN) and

Autoencoders. Although DCN has been developed since 2017, there is no implementation of DCN to classify network datasets like IoT network.

3.5 Evaluation Matrices

ML and DL classifiers are evaluated based on confusion matrix which formed using four values. Those values can be calculated using certain equations (al, 2021). In this type of classification, dataset has two conditions, either positive or negative. So, outcomes are True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). Table 3.5 presents a brief description about applied equations based on confusion matrix (al, 2021)

Table (3.5): Classifiers Evaluation Measures

Element	Description	Mathematical Equation
Recall (R)	The number of True Positives divided by the total number of actual true classification.	$R = \text{True Positive} / (\text{True Positive} + \text{False Negative})$
Precision (P)	It is also the True Positives, but it should be divided by the total of predicted positives outcome or classifications	$P = \text{True Positive} / (\text{True Positive} + \text{False Positive})$
F1-Score	F-Score is the mean of P and R	$F\text{-Score} = 2PR / (P+R)$
Accuracy (A)	it is the sum of True positive (TN) and True Negative (TN) divided by the number of examples or data points.	$A = (TP+TN) / (TP+TN+FN+FP)$

In this research, both used datasets are labeled. According to that, confusion matrix can be used to evaluate the performance of clustering methods to compare predicted labels with the ground truth. In addition, Normalized Mutual Information (NMI) is calculated. NMI (Rita, 2020) is a mathematical value used to evaluate performance of clustering algorithm by comparing two

splits. The best result is when NMI scores high result reaching close to 1 (or 100%).

The usage of the F1-score, confusion matrix, and normalized mutual information (NMI) for validating the proposed system is justified by their effectiveness in evaluating the performance of clustering algorithms. The F1-score is a measure of the overall accuracy of the clustering algorithm, considering both precision and recall. The confusion matrix provides a visual representation of the performance of the clustering algorithm, showing the number of true positives, false positives, true negatives, and false negatives. Finally, the NMI is a measure of the similarity between the ground truth and the clustering results, reflecting the degree of overlap between the clusters.

Using these metrics to validate the proposed system provides a comprehensive evaluation of its performance, considering both the accuracy and the degree of similarity between the ground truth and the clustering results. Additionally, comparing the performance of different metrics can add significant value to the thesis by providing insights into the strengths and weaknesses of different evaluation measures. For example, while the F1-score is a good overall measure of accuracy, it may not provide insights into the degree of similarity between the ground truth and the clustering results. Similarly, while the NMI provides a good measure of similarity, it may not consider the degree of overlap between the clusters. By using different metrics, it is possible to gain a deeper understanding of the performance of the proposed system.

3.6 Summary

Research methodology is the process of planning, conducting, and evaluating research studies. It involves several steps that are typically followed to ensure that the research is conducted in a systematic and rigorous manner. In this study, the research methodology involved selecting relevant datasets, choosing effective algorithms, and using appropriate evaluation metrics to assess their performance.

The selection of the datasets, CIC-IoT-2022 and IoT-23, was based on their relevance to IoT anomaly detection and their availability. These datasets contain enough IoT traffic samples to be used to train and test selected algorithms for anomaly detection and device identification.

The choice of algorithms was based on their effectiveness and performance. And they are commonly used algorithms for anomaly detection in IoT data and have been shown to be effective in previous studies. New deep clustering algorithms were used such as DynAE, DEC and DCN to investigate the effectiveness of deep clustering for IoT device identification and anomaly detection.

To evaluate the performance of these algorithms, several evaluation metrics were selected, including F1-score, confusion matrix, and NMI.

Chapter Four

Analysis and Discussion

The Results and Discussion chapter is the culmination of the research process, where all findings are presented and reflected on the significance of archived work

In this chapter, a clear and concise presentation of the results, using tables, graphs, and other visual aids are provided. Also, it contains the analysis of the findings, exploring any patterns or relationships that emerge from the data. The last section contains a comparison between our results and related work from other research in the same field using the same datasets or applying the same algorithms.

4.1 Data Analysis Results

In this research, two datasets are used to investigate the effectiveness and efficiency of selected ML and DL algorithms. Data exploration is performed to understand the natures of each data, present important statistical facts and explore features relationship. This step is essential to build strong classification of clustering models. Subsection 4.1.1 presents analysis of IoT23 dataset while subsection 4.1.2 contains analysis results for CIC-IoT-2022 dataset.

4.1.1 Analysis of IoT23 dataset

Several steps are used to perform analysis tasks on the IoT23 dataset. IoT23 dataset is a labeled IoT anomaly detection. The dataset has 22 features and 1346622 records. 50% of the features are numeric and the rest has “object”

datatype. The dataset reflects data about 12 different types of attacks along with benign records.

IoT23 is an unbalanced dataset which is shown clearly in Figure 4.5. As plotted in the Figure, most of records and data are labeled as part of a horizontal port scan attack with 676033 records. Around 14% of the data records are labeled as benign. There are some attacks with very minimal presence in the records such as C&C-Mirai and File download attacks. The issue of unbalanced data is solved by applying under-sampling techniques.

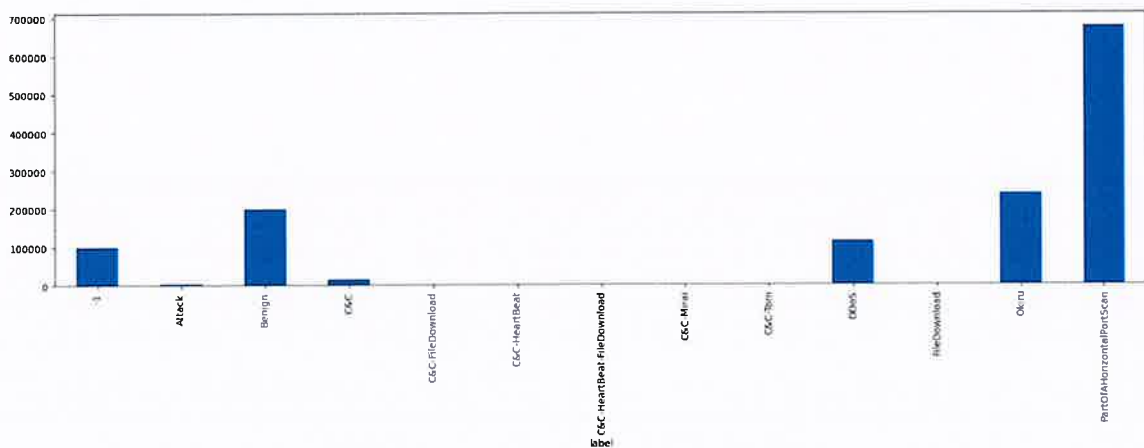


Figure 4.1): Counts of records for each attack.

For further analysis, the data is resampled using the Time feature to understand if there are any time series clustering patterns. The Time is converted to time-date datatype to be used in the analysis. Moreover, specific statistics about some features are extracted to evaluate and select which feature can help enhance clustering algorithm performance. As shown in Figure 4.2, unique values counts are extracted from four features.

```
print('number of unique originater IP address is = ',df['id.orig_h'].nunique())
print('number of unique receiver IP address is = ',df['id.resp_h'].nunique())
print('number of unique originater port is = ',df['id.orig_p'].nunique())
print('number of unique history is = ',df['history'].nunique())

number of unique originater IP address is = 3264
number of unique receiver IP address is = 896750
number of unique originater port is = 62297
number of unique history is = 167
```

Figure 4.2): Unique Values of Specific Features in IoT23 dataset

4.1.2 Analysis of CIC-IoT-2022 dataset

CIC-IoT-2022 dataset is analyzed using specific techniques and libraries in Python. The dataset consists of 50 features and 320137 samples. All features are numeric representing communication and traffic details. Each sample is labeled using device name such as Automi-coffee machine and global category like home automation. Labels are added in data preprocessing phase depending on information provided by dataset's developers.

Analysis of CIC-IoT-2022 dataset shows that 13.15% are records for Arlo camera. As shown in Figure 4.3, there are differences in samples distribution among devices. Figure 4.4 shows that most of devices relate to Camera

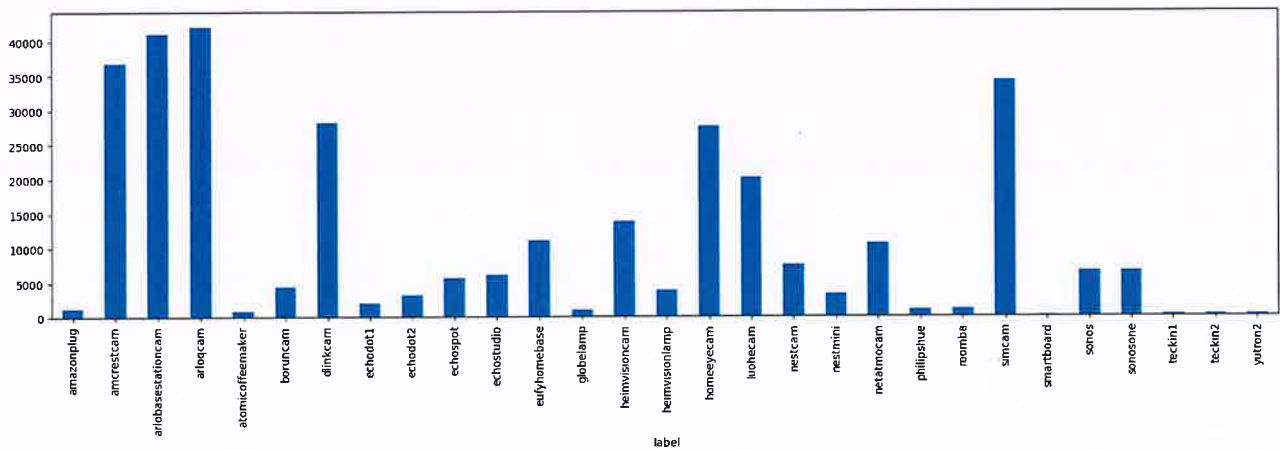


Figure 4.3): Samples Count of IoT Devices

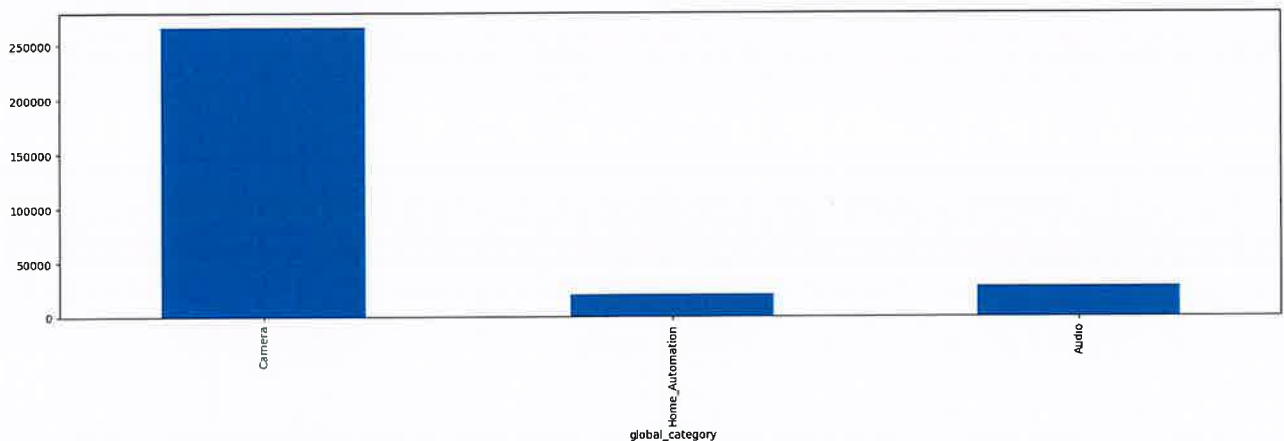


Figure 4.4): Global Category of Deices in the Dataset

One of dataset features is source's port class. Devices are classified using 4 categories of port classes. As illustrated in Figure 4.5, most devices use port class 2 (registered port).

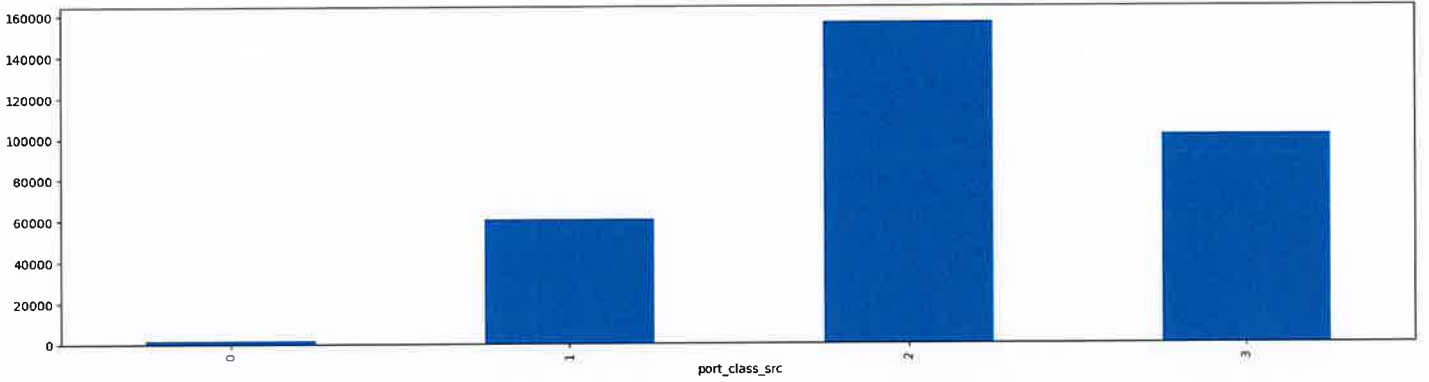


Figure (4.5): Devices classification based on Port class.

For further exploration purposes, data is resampled using the average packet size for each global category. Figure 4.6 plots average packet size for each category.

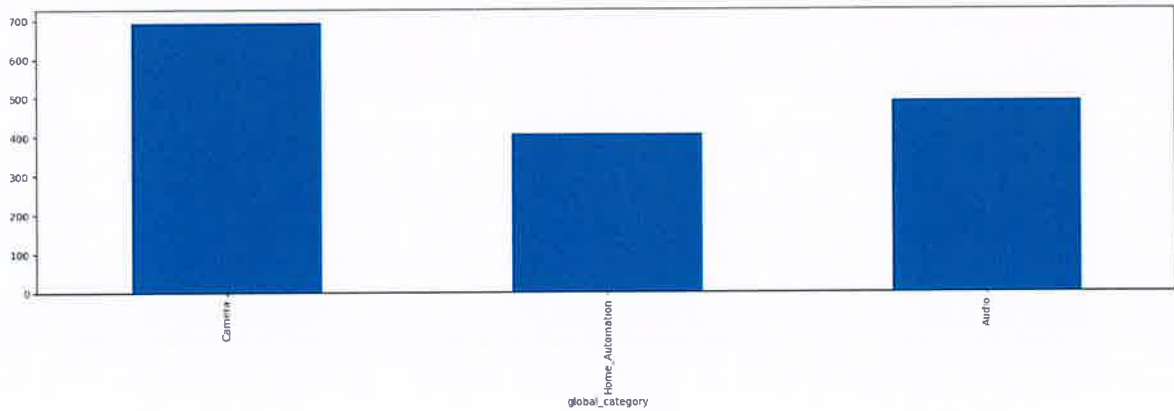


Figure 4.6): Packet average size per devices categories.

4.2 Experiments Results

In this section, the results obtained from applying the various algorithms that were introduced in Chapter 3 is reviewed. The results of the algorithms are

compared according to the purpose of their selection for our study, whether for classification or clustering of IoT device.

Applied algorithms in this research are selected based on several factors such as the algorithm suitability, the nature of IoT databases and its ability in achieving the research's objective.

Data is pre-processed to be fed to the algorithms by exploring data frame for empty values and applying min-max normalization technique. This technique is selected to preserve the relationship between features. Also, the data is divided into training, testing and validation subsets using `training_testing_split` model from `sklearn` library. 60% of the data forms the training set while the remaining 40% is divided equally on validation and testing sets. All labels are converted to numeric values instead of strings.

4.2.1 Classification Algorithms Results

Several ML and DL algorithms are adopted to develop a reliable model for IoT device identification. To perform IoT device detection and profiling, the CIC-IoT-2022 dataset is used. The first approach is to apply ML classifiers in which each one of them is applied individually and compared to each other using discussed evaluation methods. The second approach is to apply DL algorithms to develop an advanced classification model. All algorithms are tested twice to classify IoT devices based on the device type and device category. Classification where multiple classifiers are applied together, and results are combined to be able to obtain benefits from the strengths of each classifier or algorithm.

i. ANN

ANN is built using Keras library. It consists of three layers. The input layer use Relu as activation function with 500 neurons while the hidden layer has 300 neurons. The output layer uses SoftMax activation function because it is a multiclass classification task. ANN hyperparameters are tuned to enhance model's performance. ANN uses batch size of 50 for 10 epochs. Results of using ANN algorithm for IoT device identification are promising since it has 99.4% accuracy and 0.0209 loss on testing set. Figure 4.7 illustrates the change of accuracy and loss for training and testing sets over epochs.

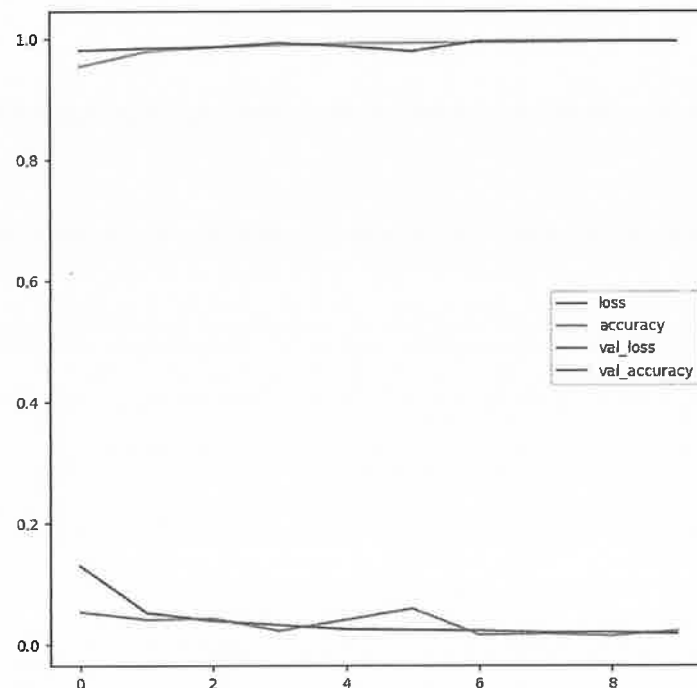


Figure (4.7): Accuracy Vs Loss for ANN

The confusion matrix is plotted to evaluate the model performance along with calculating Recall, Precision and F1-score values. As explained by Figure 4.8, ANN can identify the majority of IoT devices and assign them to the correct global category. Figure 4.9 presents model's effectiveness on IoT device identification based on device type.

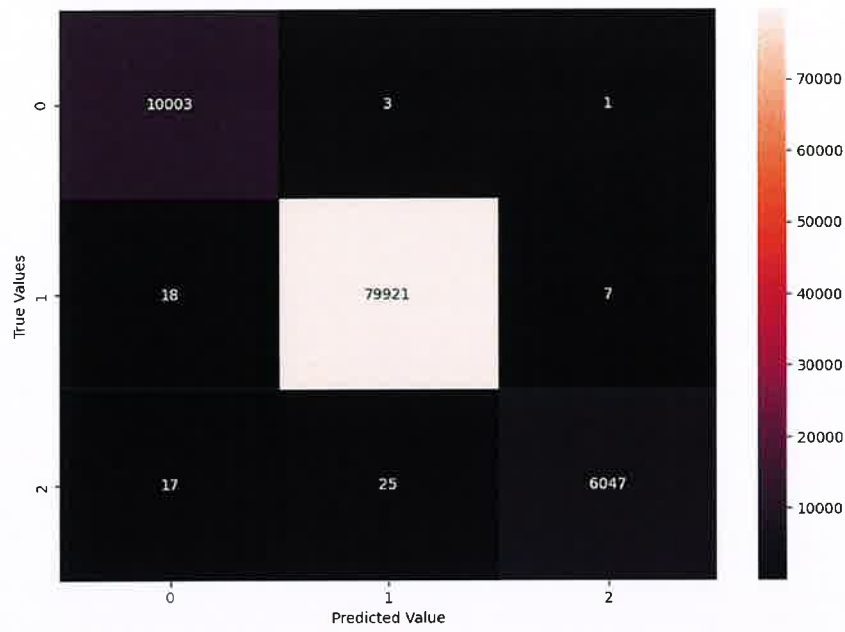


Figure (4.8): Confusion Matrix for IoT devices classification based on global category

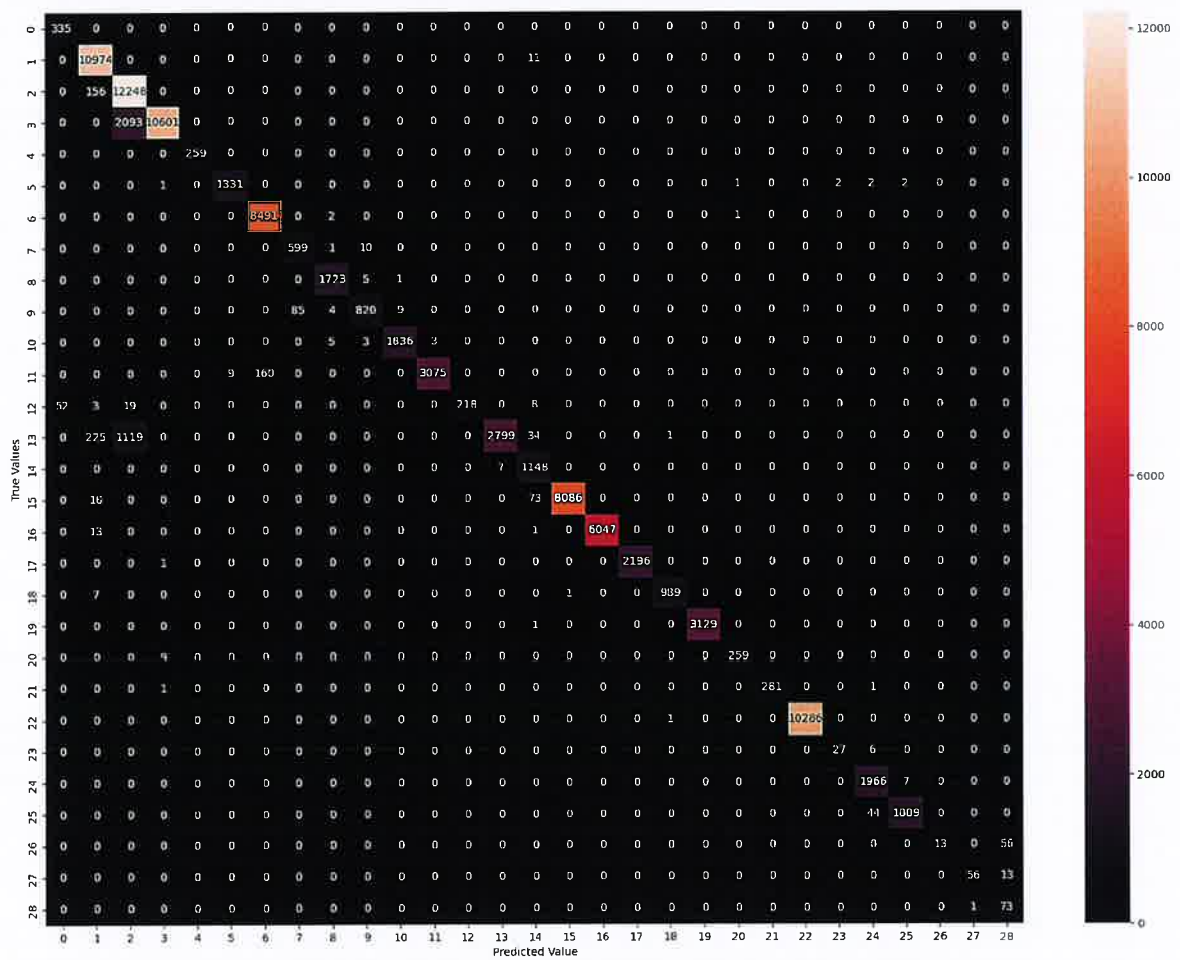


Figure 4.9): Device type-based identification using ANN

ii. CNN

CNN algorithm is used to investigate the effectiveness of DL algorithms for IoT device profiling and identification. CNN model consists of six layers including the output and the input layer. The structure of CNN model is tuned based on the task. For instance, the number of neurons changes between the model used for classification based on global category or based on device types. Figure 4.10 presents summary of CNN for both classification tasks. The left side of Figure 4.10 illustrates CNN structure to classify IoT devices based on global category while the right-side presents how CNN model is built to identify devices based on device type. The resulting model uses Adam as an optimizer and function and categorical cross entropy as loss function.

Model: "sequential"			Model: "sequential"		
Layer (type)	Output Shape	Param #	Layer (type)	Output Shape	Param #
dense (Dense)	(None, 2000)	1000000	dense (Dense)	(None, 2000)	1000000
dense_1 (Dense)	(None, 1500)	3001500	dense_1 (Dense)	(None, 1500)	3001500
dropout (Dropout)	(None, 1500)	0	dropout (Dropout)	(None, 1500)	0
dense_2 (Dense)	(None, 800)	1200800	dense_2 (Dense)	(None, 800)	1200800
dropout_1 (Dropout)	(None, 800)	0	dropout_1 (Dropout)	(None, 800)	0
dense_3 (Dense)	(None, 400)	320400	dense_3 (Dense)	(None, 400)	320400
dropout_2 (Dropout)	(None, 400)	0	dropout_2 (Dropout)	(None, 400)	0
dense_4 (Dense)	(None, 150)	60150	dense_4 (Dense)	(None, 150)	60150
dropout_3 (Dropout)	(None, 150)	0	dropout_3 (Dropout)	(None, 150)	0
dense_5 (Dense)	(None, 3)	453	dense_5 (Dense)	(None, 29)	4379
Total params: 4,683,303			Total params: 4,687,229		
Trainable params: 4,683,303			Trainable params: 4,687,229		
Non-trainable params: 0			Non-trainable params: 0		

Figure 4.10): CNN Structure Summary

The built CNN model effectiveness is explained by the plotted confusion matrix in Figure 4.11. As Figure 4.11 indicates, CNN is an efficient model to classify and identify IoT devices since it has 99% accuracy rate on validation set. This fact is supported by the values of Recall, Precision and F1-score presented in Figure 4.12 for classification IoT devices by device type. The

model was able to classify IoT devices into 28 categories, based on the device name with 99% accuracy and 94% on F1-score.

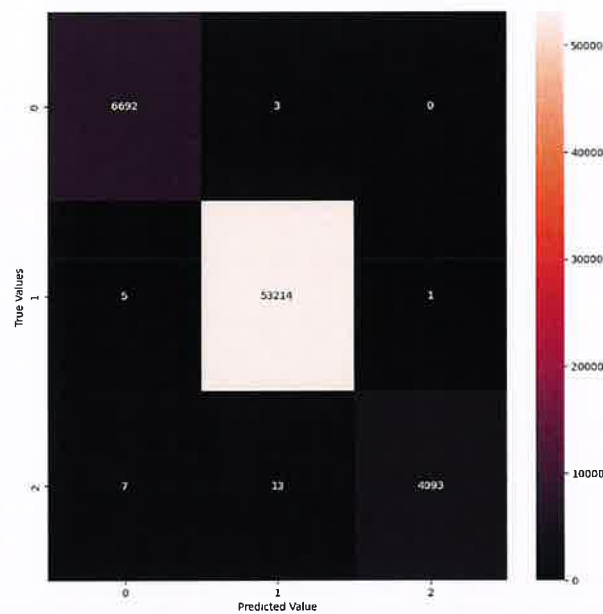


Figure (4.11): Confusion Matrix for CNN model

	precision	recall	f1-score	support
0	1.00	1.00	1.00	250
1	1.00	1.00	1.00	7462
2	1.00	1.00	1.00	8135
3	1.00	0.99	1.00	8647
4	1.00	1.00	1.00	174
5	1.00	0.99	0.99	878
6	1.00	1.00	1.00	5478
7	0.91	0.85	0.88	454
8	1.00	1.00	1.00	1135
9	0.89	0.94	0.91	619
10	1.00	1.00	1.00	1231
11	1.00	1.00	1.00	2199
12	1.00	1.00	1.00	182
13	1.00	1.00	1.00	2758
14	0.99	1.00	1.00	764
15	1.00	1.00	1.00	5525
16	1.00	1.00	1.00	4022
17	0.98	1.00	0.99	1496
18	1.00	1.00	1.00	619
19	1.00	1.00	1.00	2014
20	1.00	0.99	1.00	167
21	1.00	1.00	1.00	210
22	1.00	1.00	1.00	6805
23	1.00	0.83	0.91	30
24	0.84	0.98	0.91	1353
25	0.97	0.82	0.89	1284
26	0.50	0.87	0.63	46
27	1.00	0.67	0.80	49
28	0.54	0.31	0.39	42
accuracy			0.99	64028
macro avg	0.95	0.94	0.94	64028
weighted avg	0.99	0.99	0.99	64028

Figure (4.12): Classification Report for CNN Model

iii. DT, RF and KNN

DT, RF and KNN models are used to identify IoT devices using classical ML algorithms. All three algorithms are built automatically based on Keras library. Table 4.1 presents summary results of using the three algorithms to identify devices based on global category. DT and RF were able to score 100% on every measure value such accuracy, recall and F1 -score. Those results were archived only after splitting the dataset into training, validation, and testing datasets. The full results comparing to other used algorithms is explained in the following section. The results of KNN are normal considering the strength of the model itself as supervised classification algorithm.

Table (4.1): ML Algorithms Performance Comparison

Evaluation Measure	DT	RF	KNN
Recall (R)	100%	100%	99.7
Precision (P)	100%	100%	99.7
F1-Score	100%	100%	99.4
Accuracy (A)	100%	100%	99.7

4.2.2 Clustering Algorithms Results

The main objective of this research is to investigate the usefulness of using deep clustering algorithms for IoT device identification and anomaly detection. To accomplish research objectives, both selected datasets are used to explore the value of using clustering models. All algorithms are used twice, once for each dataset. CIC-IoT-2022 is used to identify IoT devices while the IoT23 dataset is used for anomaly detection. Due to computational expense of clustering algorithms, models are used to identify IoT devices based on the global category only.

The experiment starts by applying ML algorithms as an initial step then applying deep clustering algorithms. This method explains the level of sophistication deep clustering algorithms can bring to IoT anomaly detection and devices identification tasks. Kmeans (KM) and DBSCAN are implemented as ML clustering algorithms. On the other hand, two unsupervised deep clustering algorithms are trained which are DEC and DynAE.

Data from IoT23 is processed to be used in clustering models by under-sampling and removing attacks records with small instances. After this step the data contains 4 types of attacks and benign records. This step is not necessary to perform on CIC-IoT-2022 because the data is balanced and has 3 devices classes.

i. ML algorithms: KM and DBSCAN

Kmeans (KM) clustering algorithm is built using Sklearn library. The first experiment is to train KM model on IoT23 database and CIC-IoT-2022 dataset. Number of clusters is found by using elbow techniques and calculating Silhouette Coefficient (SI) for various number of clustering. Training KM was computationally expensive due to data nature and number of records. Since both datasets are labeled, the performance of KM is compared to the ground truth. The best performance of KM using IoT23 dataset with $Si = 0.68$ on $k = 7$, which is more than the ground truth by 2. The accuracy is calculated by comparing kmeans predicted labels and true labels. The highest accuracy is 50%. KM was not able to perform well using CIC-IoT-2022 dataset and scored 43.2% in accuracy. The best $Si = 0.57$ at $k=3$ which is equal to the real number of global categories. The selection of $k=3$ is clearly shown in Figure 4.13 by using elbow technique. Figure 4.14 shows the

classification reported printed by comparing true classification labels of attacks and predicted KM labels in IoT23 dataset.

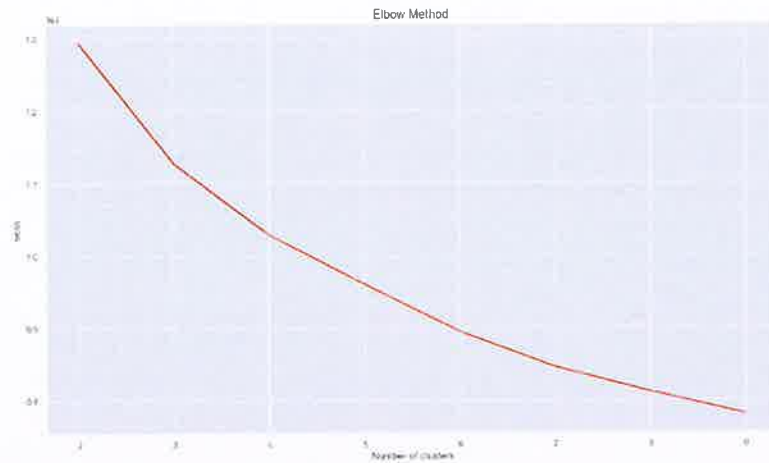


Figure (4.13): Elbow graph to select K in CIC-IoT-2022 dataset

Due to overlapping clusters, it was difficult task to plot KM predicted cluster separated from each other. This result is interpreted in Figure 4.14.

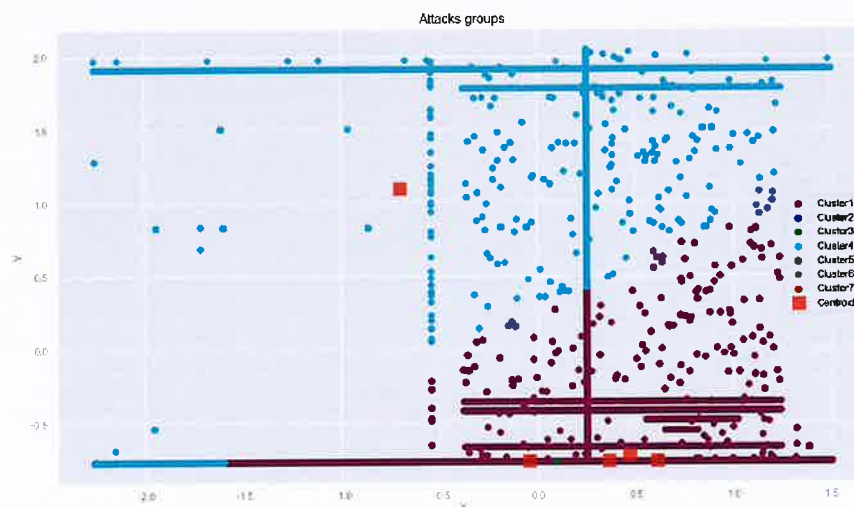


Figure (4.14): Clusters created by KM on IoT23 dataset

The second experiment is done using the DBSCAN algorithm. Same as KM, DBSCAN is trained and tested using both datasets. DBSCAN was able sperate clusters completely from each other, but it assigns datapoint to wrong clusters.

The plotted clusters are shown in Figure 4.15. The model has 34.2% accuracy for $K=4$ for both datasets, which is not equal to real groups of labels.

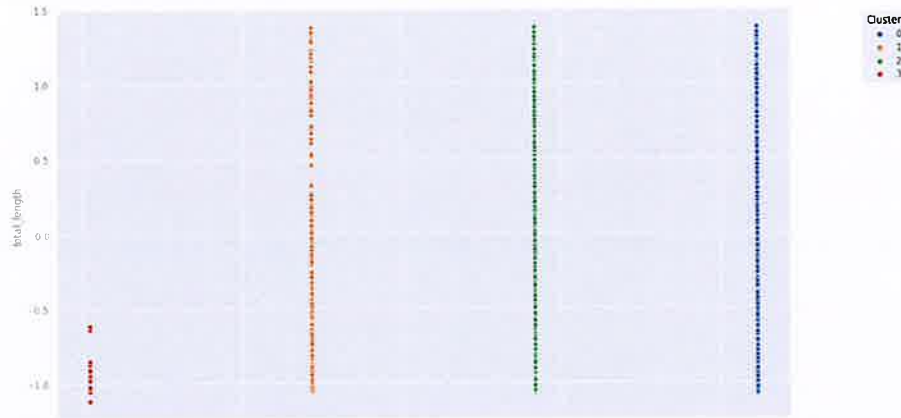


Figure 4.15): Clusters created by DBSCAN algorithm

ii. Deep Learning Algorithms: DCN, DEC and DynAE

Three deep clustering algorithms are used to overcome the weakness of classical clustering algorithms, namely DCN, DEC and DynAE. All three algorithms are built based on DNN and Autoencoders, but DEC and DynAE utilize Pytorch powerful tools and models. The pretraining phase is initial for both DEC and DynAE models.

Figure 4.16 presents the functionality of DCN to relocate datapoints to clusters based on KM results. Graph (a) shows the randomness in datapoints assigning with KM while Graph (c) presents DCN result when clusters having clear distance and points are clustering around cluster's centroid.

Cluster loss is calculated and monitored for the three algorithms. Figure 4.17 presents an example of plotting loss in DynAE models. As shown in the figure, the loss value changes over time and epochs and starts to stable after a certain number.

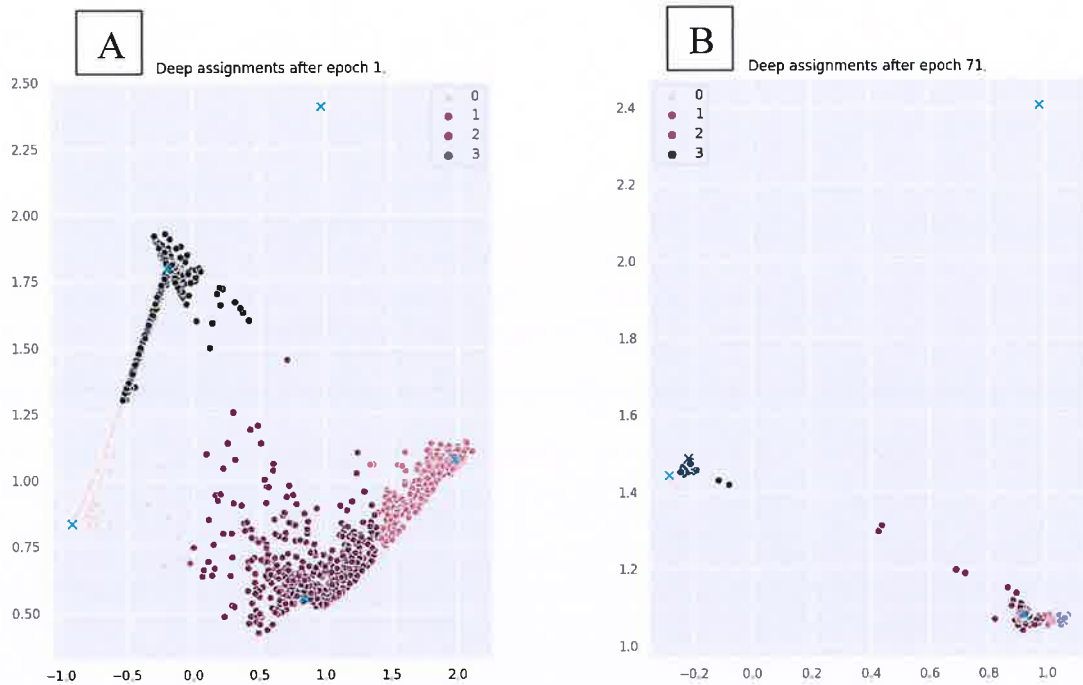


Figure (4.16): Deep Clustering in 1st epoch VS last epoch using DCN

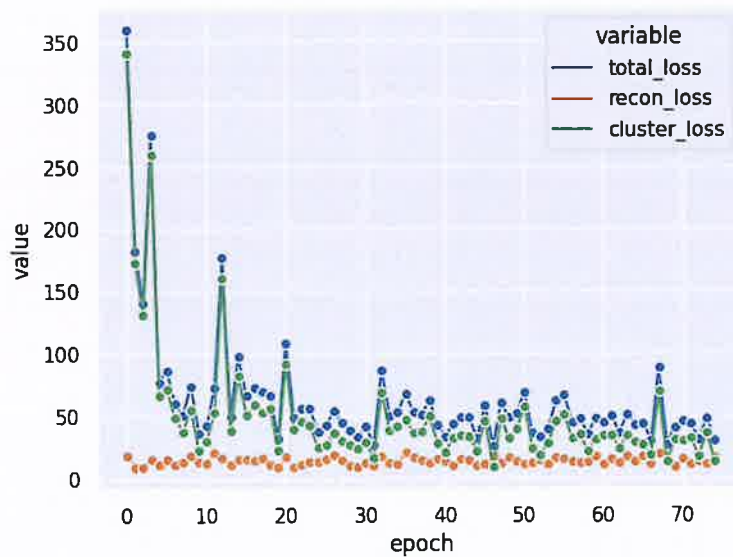


Figure (4.17): Loss Value Comparison the epochs

In general, the performance of deep clustering algorithms is better than ML clustering algorithms. To evaluate the effectiveness of the three models used, Si and NMI values are calculated and compare to the ground truth and results of applying by ML clustering algorithms. Table 4.2 shows comparison

between results of performing clustering using deep clustering models vs ML clustering models. As shown in the table, DynAE has the best performance with $S_i = 0.88$ in $K = 5$ using IoT23 dataset. The model was able to reach 81% accuracy and 0.75 on NMI. The second-best algorithm is DCN with $NMI = 0.7$ and accuracy of 79% using the same dataset. DEC has almost the same performance of DCN, but it requires more computational time. Models' performance on IoT23 for clustering is more efficient than on CIC-IoT-2022 dataset.

Using deep clustering algorithms with IoT dataset is a challenging task because of the special nature of the data extracted from the network. The DynAE model reached the best performance after several modification of code and hypermarkets tuning. The first major step was to prepare the model to load customize dataset other those that the model is trained to load and process. This step was archived after several trial-and-error experiments. After that Google Collab was used to per-train the model and save the pre-train weights but it was not effective nor efficient because of the computational expenses needed by DynAE model. In the stage, Benefit Advanced AI and Computing Lab in UOB was used to run all deep clustering algorithms files, including DEC and DCE. Needless to mention that additional code modifications were required to be able to the run the code since all of them were built using Google Collab notebooks.

Table (4.2): Results of Deep Clustering Algorithms VS ML clustering Algorithms

Evaluation Measures	Classical Clustering Models				Deep Clustering Models					
	KM		DBSCAN		DCN		DynAE		DEC	
	IoT23 True k=5	CIC True k= 3	IoT23 True k=5	CIC True k= 3	IoT23 True k=5	CIC True k= 3	IoT23 True k=5	CIC True k= 3	IoT23 True k=5	CIC True k= 3
K	7	3	5	4	4	4	5	4	4	5
Si	0.54	0.34	0.51	0.33	0.59	0.41	0.88	0.66	0.59	0.41
ACC	50%	43%	34.2%	34.2%	79%	63%	81%	60%	78%	63%
NMI	0.27	0.12	0.20	0.08	0.7	0.55	0.75	0.51	0.7	0.55

4.3 Discussion

In this section, the work done in this research is compared to related work either by using same algorithms or same datasets.

In 2022, developer of CIC-IoT-2022 dataset (Dadkhah, et al., 2022) tested the data using certain ML classifiers. They used the data to train 12 ML algorithms and classify IoT devices based on global category. In this research, the dataset was used to train 5 ML classifiers to detect IoT devices based on device type and global category. Also, they used Accuracy(A) as the evaluation measures while we used confusion matrix elements. Table 8 provides a comparison between the same classifiers used in both studies depending on Accuracy(A). As shown in the table, we reached the optimal scores on all the classifiers on testing and validation sets. This indicates that the built models can identify IoT device types with a large scale of accuracy

even if it is fed with new data and it is visible to be employed in the real world as an IoT devices classification and identifying model. Those results were archived after several experiments specially when applying ANN algorithms. Hypermarkets were tuned multiple times until reaching the best results such as number or hiding layers, number of neurons, epoch size, learning rate and loss function.

Table (4.3): Comparison of ML classifiers with work of CIC-IoT-2022 developers

	DT	KNN	RF	ANN
(Dadkhah, et al., 2022)	98.5	95.5	98.5	50.4
Our method	100%	99.7%	100%	99.9%

As per literature review done in this research, CIC-IoT-2022 dataset has not been employed yet on other experiments neither used with DL classifiers or clustering algorithms. Applying new DL algorithms on this dataset is a significant contribution.

Regarding IoT23 dataset, it has been used in many studies for anomaly detection. Most of the literature used IoT23 with ML and DL classifiers. Few researchers used IoT23 data set for semi-supervised learning and clustering tasks. (Bobrovnikova, Lysenko, Gaj, Martynyuk, & Denysiuk, 2020) applied fuzzy c-means clustering to detect anomalies in IoT network. The proposed model had scored 94.2% in accuracy. In this research, we employed the dataset with new methods of deep clustering models such as DynAE and DEC.

Reasons of low performance of classical ML clustering algorithms refer to the special nature of IoT network data. IoT data in use varies in density and size which led to mislocate datapoints to the correct clusters. This can be solved by using deep clustering algorithms. Also, deep clustering models solved the

issue of features selection and dimensionality reduction automatically by utilizing autoencoders.

4.4 Summary

This chapter presented the results of using deep clustering algorithms for IoT anomaly detection and device identification. Two datasets, CIC-IoT-2022 and IoT-23, were used to train and test several ML and DL algorithms. Results showed that all supervised classifiers achieved high scores in accuracy, F1-score, recall, and precision compared to previous research using the same datasets, particularly the CIC-IoT-2022 dataset. The IoT-23 dataset had not been used to train deep clustering models before, and this was the main contribution of the study.

Among both machine learning and deep learning clustering algorithms, the DynAE algorithm had the best performance. The study's findings suggest that deep clustering algorithms can be effective in IoT anomaly detection and device identification, particularly when dealing with complex datasets. The high performance achieved by the supervised classifiers indicates the importance of using labeled data in training machine learning models.

Chapter Five

Conclusion and Recommendation

This research is concluded with Chapter 5 by providing a summary of results and findings along with a discussion of research limitations and recommendations for future work.

5.1 Conclusion

IoT networks and devices are an emerging technology that has been used in many aspects of daily life. IoT devices interact with people and other devices and use sensitive information to preserve this communication. Due to the expanded employment of IoT devices and the increasing number of cyber-attacks, it is a necessity to investigate the role of ML and DL algorithms in IoT identification and anomaly detection.

This research was conducted to explore the effectiveness of using unsupervised deep learning algorithms for IoT profiling and detecting anomalies. The idea came from the fact that labeled data are not always available and it is useless in the case of zero-day attack or new devices trying to connect the network.

Two datasets were used to test and train several ML and DL algorithms. For IoT devices profiling, CIC-IoT-2022 dataset was employed to test 5 different models. All tested models were effective and perform classification in high rate of accuracy. This leads to the conclusion that the practitioner can choose any of these classification algorithms, and the main factor in the selection may be the efficiency of the algorithm in terms of time and space used.

For anomaly detection, IoT-23 dataset was selected to explore the optimal way of deep clustering deployment. Three deep clustering algorithms were deployed using both datasets, namely DEC, DCN and DynAE. Research Results showed DynAE to be the most successful deep clustering tool with accuracy of 81%. This success may be due to the algorithm's ability to perform feature selection and utilize the autoencoder in cluster discovery.

The performance of DynAE algorithm provided clear answers to the conducted research questions. The proposed model was able to detect anomalies in IoT devices and can be used to detect known and unknown attacks. The introduced deep clustering algorithms provided additional benefits, such as improved accuracy and efficiency, compared to traditional ML classification algorithms since they do not need to be trained using labelled data, which create a generalized model that can cope with the massive changes in IoT environment. The tested deep clustering models can be trained to identify patterns and similarities in complex data sets. In the context of IoT anomaly detection and device identification, those models can be trained on sensor data collected from various IoT devices. The models can then group the devices into clusters based on their behavior and usage patterns to identify abnormal behavior or unauthorized devices.

5.2 Future Work and Research Limitations

The applications of deep clustering models, such as DynAE and DCN, are not limited to IoT anomaly detection and device identification. In natural language processing, these models can be used to group similar documents together or perform topic modeling. In computer vision, deep clustering

models can be used for image segmentation, object recognition, and facial recognition.

Deep clustering models can learn complex non-linear relationships between data points. This makes them well-suited for tasks where traditional clustering algorithms may not be sufficient. Additionally, these models can be trained on large datasets, which can improve their accuracy and generalization capabilities. As such, the potential applications of deep clustering models are vast, and they are likely to continue to be an active area of research.

There are few limitations in using deep clustering algorithms for IoT devices identification and anomaly detection. First, deep learning algorithms are computationally expensive and time-consuming. It requires many resources to be able to run the algorithm. Also, the IoT network datasets have a specific nature, and it is not clustering-friendly. So, it needs extra effort to understand the data, feature relationships and perform features engineering to prepare the data to be used within clustering algorithm. In addition, the ML classifiers were trained and tested using the same dataset, but it will be best practice to test them using new data to make sure that the model is generalizing well.

Deep learning clustering models are powerful tools to build realistic detection models. To that end, it is recommended to put extra effort in studying the effectiveness of using deep clustering models for IoT anomaly detection systems or IoT identification models. Further investigation is needed to optimize these deep learning models computationally. Also, future studies need to investigate the generalizability and scalability of the proposed deep clustering algorithms and their applicability in real-world IoT systems.

Finally, the research showed that IoT devices and network datasets are mainly collected for classification purposes. It would be a major contribution to build an IoT devices dataset to perform clustering tasks and optimize deep clustering algorithms using a standard realistic IoT dataset.

References

- Agarwal, R. (2019, November 20). *The 5 most useful Techniques to Handle Imbalanced datasets*. Retrieved from Towards Data Science:
<https://towardsdatascience.com/the-5-most-useful-techniques-to-handle-imbalanced-datasets-6cdba096d55a>
- al, S. D. (2021). A predictive Analytics Model for Students Grade Prediction by Supervised Machine Learning . *Materials Science and Engineering*.
- Al-Amri, R., Murugesan, R. K., & Alshari, E. M. (2021). Toward a Full Exploitation of IoT in Smart Cities: A Review of IoT Anomaly Detection Techniques. In *Proceedings of International Conference on Emerging Technologies and Intelligent Systems* (pp. 193 - 214). Bagdad: Springer.
- Best, L., Foo, E., & Tian, H. (2022). A Hybrid Approach: Utilising Kmeans Clustering and Naive Bayes for IoT Anomaly Detection. *arXiv preprint*.
- Bhandari, A. (2023, April 11). *Feature Engineering: Scaling, Normalization, and Standardization*. Retrieved from Analytics vidhya:
<https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/>
- Bobrovnikova, K., Lysenko, S., Gaj, P., Martynyuk, V., & Denysiuk, D. (2020). Technique for IoT Cyberattacks Detection Based on DNS Traffic Analysis. *CEUR Workshop Proceedings*.
- Bruhadeshwar Bezawada, M. B. (2018). IoT Sense :Behavioral Finger printing of IoT Devices. *Attacks and solutions in hardware security* , 41-50.
- Chassagnon, G., Vakalopoulou, M., Paragios, N., & Revel, M. P. (2020). Deep learning: definition and perspectives for thoracic imaging. *European Radiology*, 2021–2030.
- Chatterjee, A., & Bestoun, A. S. (2022). IoT anomaly detection methods and applications: A survey. *Internet of Things*.

- Chen, H., Xiao, R., & Jin, S. (2021). Unsupervised Anomaly Detection Based on System Logs. *The 33rd International Conference on Software Engineering and Knowledge Engineering*, (pp. 92-97). USA: KSI Research Inc.
- Cook, A. A. (2019). Anomaly detection for IoT time-series data: A survey. *IEEE Internet of Things Journal*, 6481-6494.
- Dadkhah, S., Mahdikhani, H., Danso, P. K., Zohourian, A., Truong, K. A., & Ghorbani, A. A. (2022). Towards the Development of a Realistic Multidimensional IoT Profiling Dataset. 1-11.
- Deng, L., & Yu, D. (2014). Deep Learning: Methods and Applications. *Foundations and trends® in signal processing*, 197-387.
- Duarte, F. (2023, February 22). *Number of IoT Devices (2023)*. Retrieved from Exploding Topics: <https://explodingtopics.com/blog/number-of-iot-devices>
- Dupont, G., Leite, C., Santos, D. R., Costante, E., Hartog, J. d., & Etalle, S. (2021). Similarity-Based Clustering For IoT Device Classification. *2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS)*.
- Dutta, V., Chora's, M., Pawlicki, M., & Kozik, R. (2020). A Deep Learning Ensemble for Network Anomaly and Cyber-Attack Detection. *Sensors*.
- Dutta, V., Choraś, M., Pawlicki, M., & Kozik, R. (2020). *A Deep Learning Ensemble for Network Anomaly and Cyber-Attack Detection*. *Sensors*.
- Eleonora Achiluzzi, M. L. (2022). Exploring the Use of Data-Driven Approaches for Anomaly Detection in the Internet of Things (IoT) Environment .
- F. Meneghello, M. C. (2019). IoT: Internet of Threats? A survey of practical security vulnerabilities in real IoT devices. *IEEE Internet of Things Journal*, pp. 8182-8201.
- Fang, L., & Li, T. (2018). A Clustering K-Anonymity Privacy-Preserving Method for Wearable IoT Devices. *Security and Communication Networks*.

- Ferman, V. A., & Tawfeeq, a. M. (2021). A Machine Learning-Based Device Fingerprinting Method for IoT Systems. *Journal of Physics: Conference Series*, 50-62.
- Garcia, S. (2020). *Aposemat IoT-23*. Retrieved from STRATOSPHERE: <https://www.stratosphereips.org/datasets-iot23>
- Garcia, S., Parmisano, A., & Erquiaga, M. J. (2020). *IoT-23: A labeled dataset with malicious and benign IoT network traffic*. Zenodo. doi:<http://doi.org/10.5281/zenodo.4743746>
- Guo, G., Wang, H., Bell, D., Bi, Y., & Greer, K. (2003). KNN Model-Based Approach in Classification. 986-996.
- Gupta, A. (2020). Feature Selection Techniques in Machine Learning. *Analytics Vidhya*.
- Gupta, K. (2019). *Machine Learning-Based Device Type Classification for IoT Device Re- and Continuous Authentication*. [Master Thesis], University of Nebraska-Lincoln.
- Hamad, S. A., Zhang, W. E., Sheng, Q. Z., & Nepal, S. (2019). IoT Device Identification via Network-Flow Based Fingerprinting and Learning. *18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering* (pp. 103-111). IEEE.
- Helmi Ben Hmida, A. B. (2015). Enabling an Internet of Things Framework for Ambient Assisted Living. *AAL kongress 2015 Springer Verlag*.
- Hou, T., Wang, T., Lu, Z., Liu, Y., & Sagduyu, Y. (2022). IoTGAN: GAN Powered Camouflage Against Machine Learning Based IoT Device Identification. *archivePrefix*.
- Howarth, J. (2023, March 16). *Amazing IoT Statistics (2023-2030)*. Retrieved from Exploding topics: <https://explodingtopics.com/blog/iot-stats>

- IBM. (2020). *What is deep learning?* Retrieved from IBM official Website:
<https://www.ibm.com/topics/deep-learning>
- Karanfilovska, M., Kochovska, T., Todorov, Z., Cholakovska, A., akimovski, G., & Efnusheva, D. (2022). Analysis and modelling of a ML-based NIDS for IoT networks . *Procedia Computer Science*, 187–195.
- Khan, M. Y., Abdul, Q., Muhammad, N., Shoaib, S. M., Shaukat, W., & RaaziSyed, K.-U.-R. (2021). Automated Prediction of Good Dictionary EXamples (GDEX): A Comprehensive Experiment with Distant Supervision, Machine Learning, and Word Embedding-Based Deep Learning Techniques.
- Kotak, J., & Elovici, Y. (2020). IoT Device Identification Using Deep Learning. *13th International Conference on Computational Intelligence in Security for Information Systems* (pp. 76–86). Springer.
- Kozik, R., Pawlicki, M., & Choraś, M. (2021). A new method of hybrid time window embedding with transformer-based trafıc data classification in IoT-networked environment. *Pattern Analysis and Applications*, 1441–1449 .
- Li, L. (2019). Principal Component Analysis for Dimensionality Reduction.
- Linna Fan, S. Z. (2020). An IoT Device Identification Method based on Semi-supervised Learning. *In 2020 16th International Conference on Network and Service Management (CNSM)*, 1-7.
- Liu, X., Han, Y., & Du, Y. (2022). IoT Device Identification Using Directional Packet Length Sequences and 1D-CNN.
- Mandal, M. (2021). Introduction to Convolutional Neural Networks (CNN).
- Meidan, Y. B. (2017). Detection of Unauthorized IoT Devices Using Machine Learning Techniques. *arXiv preprint*.
- Mohamed, Z., Ksantini, R., & Kaabi, J. (2022). Convolutional Dynamic Auto-encoder: A Clustering Method for Semantic Images. *Neural Computing and Applications*.

- Mothe, J., Mkhitarian, K., & Haroutunian, M. (2017). Community detection: Comparison of state of the art algorithms. *omputer Science and Information Technologies (CSIT)*, 125-129.
- Nairouz, M., Mefraz, K. a., Ksantini, R., & Zied, L. (2020). Deep clustering with a dynamic autoencoder: From reconstruction towards centroids construction. *Neural Networks*, 206--228.
- Nesrine A, L. N. (2020). Autonomous Identification of IoT Device Types based on a Supervised Classification. *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, (pp. 1-6). Dublin.
- Neu, D. A., Lahann, J., & Fettke, P. (2021). A systematic literature review on state-of-the-art deep learning methods for process prediction. *Artificial Intelligence Review*, 801–827.
- Nukavarapu, S. K., & Nadeem, T. (2021). Securing Edge-based IoT Networks with Semi-Supervised GANs. 579-584.
- Nuspire, T. (2022). *The Current State of IoT Security*. security boulevard.
- Pahl, M. O., & Aubet, F. X. (2018). All Eyes on You: Distributed Multi-Dimensional IoT Microservice Anomaly Detection. *International Conference on Network and Service Management*, 72-80.
- Patel, K. K. (2016). Internet of things-IOT: definition, characteristics, architecture, enabling technologies, application & future challenges. . *International journal of engineering science and computing*,.
- R, S. E. (2021). Understand Random Forest Algorithms With Examples.
- Razzak, I., Zafar, K., Imran, M., & Xu, G. (2020). Randomized nonlinear one-class support vector machines with bounded loss function to detect of outliers for large scale IoT data. *Future Generation Computer Systems*, 715-723.

- Rebala, G., Ravi, A., & Churiwala, S. (2019). *Machine Learning Definition and Basics*. In: *An Introduction to Machine Learning*. Springer, Cham.
doi:https://doi.org/10.1007/978-3-030-15729-6_1
- Rita, L. (2020, May 7). *Normalized Mutual Information*. Retrieved from <https://luisdrita.com/normalized-mutual-information-a10785ba4898>
- Safi, M., Dadkhah, S., Shoeleh, F., Hassan Mahdikhani, Molyneaux, H., & Ghorbani, A. (2022). A Survey on IoT Profiling , Fingerprinting, and Identification. *ACM Transactions on Internet of Things*.
- Sahu, A. K., Sharma, S., Tanveer, M., & Raja, R. (2021). Internet of Things attack detection using hybrid Deep Learning Model. *Computer Communications*, 146–154.
- Saini, A. (2021). Decision Tree Algorithm – A Complete Guide.
- Saran, N., & Kesswani, N. (2023). A comparative study of supervised Machine Learning classifiers for Intrusion Detection in Internet of Things. *Procedia Computer Science*, 2049–2057.
- Sharma, N., Shamkuwar, M., & Singh, I. (2019). The History, Present and Future with IoT. In V. Balas, V. Solanki, R. Kumar, & M. Khari, *Internet of Things and Big Data Analytics for Smart Generation*. Intelligent Systems Reference Library.
- Sharma, P. (2019). The Ultimate Guide to K-Means Clustering: Definition, Methods and Applications.
- Sharpe, R., Warnicke, E., & Lamping, U. (2022). *Wireshark User's Guide*. Wireshark. Retrieved from https://www.wireshark.org/docs/wsug_html_chunked/
- Shen, X. S., Lin, X., & Zhang, K. (2020). *Encyclopedia of Wireless Networks*. Springer, Cham.
- Simar Preet Singh, A. S. (2021). Chapter 5 - Internet of Intelligent Things: Injection of Intelligence into IoT Devices. In P. T. Gurjit Kaur, *Artificial Intelligence to Solve Pervasive Internet of Things Issues* (pp. 85-102). Academic Press.

- Singh, G. (2021). Introduction to Artificial Neural Networks.
- Sivanathan, A. S. (2017). Characterizing and classifying IoT traffic in smart cities and campuses. *IEEE Conference on Computer Communications Workshop*.
- Sivanathan, A., Gharakheili, H. H., & Sivaraman, V. (2019). Inferring IoT Device Types from Network Behavior Using Unsupervised Clustering. 230-233.
- Songqiao Han, X. H. (2022). ADBench: Anomaly Detection Benchmark. *36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmark*.
- Subramanian, D. (2019). A Simple Introduction to K-Nearest Neighbors Algorithm.
- Valrorta, J. M., Martino, A., Cuomo, F., & Garlisi, D. (2019). A Clustering Approach for Profiling LoRaWAN IoT Devices. 58-74.
- Vermesan Ovidiu, F. P. (2014). *Internet of things applications-from research and innovation to market deployment*. Taylor & Francis.
- Vijayanand Thangavelu, D. M. (2019). DEFT: A Distributed IoT Fingerprinting Technique. *IEEE Internet of Things Journal*.
- Vitorino, J., Andrade, R., Praça, I., Sousa, O., & Maia, E. (2022). A Comparative Analysis of Machine Learning Techniques for IoT Intrusion Detection. 191–207.
- Weng, W., Layeghy, S., Sarhan, M., Gallagher, M., & Portmann, M. (2022). E-GraphSAGE: A Graph Neural Network based Intrusion Detection System for IoT. *arXiv*.
- Xiao Liang, W. X. (2018). IoT Security Techniques Based on Machine Learning. *IEEE Signal Processing Magazine*, pp. 41-49.
- Xie, J., Girshick, R., & Farhadi, A. (2015). Unsupervised Deep Embedding for Clustering Analysis.
- Yair Meidan, M. B. (2017). rofillIoT: a machine learning approach for IoT device identification based on network traffic analysis. *Proceedings of the Symposium on Applied Computing* (pp. 506–509). Association for Computing Machinery.

Yang, B., Fu, X., Sidiropoulos, N. D., & Hong, M. (2017). Towards K-means-friendly Spaces: Simultaneous Deep Learning and Clustering.

Yıldırım, S. (2020, 22 4). *DBSCAN Clustering — Explained*. Retrieved from Towards data science: <https://towardsdatascience.com/dbscan-clustering-explained-97556a2ad556>

Yousefnezhad, N. M. (2021). Automated iot device identification based on full packet information using real-time network traffic. . *Sensors*, 2660.

ملخص

تزايد في الآونة الأخيرة توظيف مفهوم انترنت الأشياء في العديد من القطاعات الحيوية مثل التعليم، والصحة، والاقتصاد، وغيرها. يعبر انترنت الأشياء عن التقنية المستخدمة لربط أجهزة باستخدام شبكة الانترنت واستخدام وسائل محددة لإنشاء قنوات التواصل ما بين الأجهزة ذاتها أو ما بين الأجهزة والمستخدمين. تتميز أجهزة انترنت الأشياء بعدة مميزات منها محدودة الذاكرة والقدرة على التعامل مع كم كبير من المعلومات المرسله عبر الشبكة وغيرها من الخصائص. مع الاستخدام المتزايد لأجهزة انترنت الأشياء، تزايد المخاوف حول قدرتها على الحفاظ على معلومات المستخدمين وخصوصيتهم وبالأخص إن قد تكون عرضة للاختراق بسهولة.

أقدم العلماء على توظيف مبادئ تعلم الآلة والتعلم العميق لتصميم خوارزميات خاصة تساعد على زيادة الأمن السيبراني الخاص بأجهزة انترنت الأشياء وحمايتها من الهجمات الإلكترونية. توظيف هذه الخوارزميات بالأخص لتصنيف أجهزة انترنت الأشياء والتعرف عليها بالإضافة إلى رصد حالات الاختراق من خلال رصد أي حالة شذوذ في المعلومات المرسله عبر الشبكة. معظم الباحثين أقدموا على توظيف تقنيات تعلم الآلة بهدف التصنيف، ولكنها محدودة القدرة إذا أخذنا بعين الاعتبار التغير السريع في أجهزة انترنت الأشياء وكذلك تنوع الهجمات الإلكترونية. لذا تتضح الحاجة لتوظيف تقنيات العنقدة العميقة وذلك لتصميم نموذج نكي يمكنه التعرف على الأجهزة وتصنيف حالات الشواذ في الشبكة.

في هذه الدراسة تم توظيف قاعدتين من قواعد البيانات العامة التي تستخدم لدراسة أجهزة انترنت الأشياء وتطبيق العديد من خوارزميات تعلم الآلة والعنقدة العميقة. وقدت أوضحت النتائج تقدم نموذج DynAE وتحقيقه نسبة دقة عالية في رصد الحالة الشاذة في قاعدة البيانات المستخدمة بنسبة ٨١٪.

جامعة البحرين
كلية تقنية المعلومات



توظيف خوارزميات العنقدة العميقة في تصنيف ورصد الحالات الشاذة في أجهزة إنترنت الأشياء

أطروحة مقدمة من ضمن متطلبات الحصول على درجة الماجستير
في تعلم الآلة والحوسبة الذكية

إعداد

نور حسين عبدالله سرحان
رقم جامعي: ٢٠٠٦٢٣٧٥

إشراف

د. هشام العمال
أستاذ مساعد
جامعة البحرين

المشرف المشارك

د. حصة الجنيد
أستاذ مساعد
جامعة البحرين

مملكة البحرين

أبريل ٢٠٢٣ م