T.C.

AKSARAY UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE

DEPARTMENT OF ELECTRICAL-ELECTRONIC AND COMPUTER ENGINEERING

ROBUST IDSS DESIGN FOR IoT BASED ON SMART ALGORITHMS

Ph. D. THESIS

Tamara Saad Mohamed AL-JANABI

SUPERVISOR

Prof. Dr. Sezgin AYDIN

AKSARAY, 2022

T.C.

AKSARAY UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE

DEPARTMENT OF ELECTRICAL-ELECTRONIC AND COMPUTER ENGINEERING

ROBUST IDSS DESIGN FOR IoT BASED ON SMART ALGORITHMS

Ph. D. THESIS

Tamara Saad Mohamed Al-janabi

SUPERVISOR

Prof. Dr. Sezgin AYDIN

AKSARAY, 2022

PLAGIARISM PAGE

I hereby declare that all information in this thesis has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work; otherwise I accept all legal responsibility.

Tamara Saad Mohamed AL-JANABI

ACKNOWLEDGEMENTS

First of all, I would like to express my deepest appreciation to my esteemed supervisor, Prof. Dr. Sezgin AYDIN, who gave me this great research opportunity. I am heartily thankful to him for his tireless guidance, extraordinary patience and constant encouragement during my academic journey. Without his persistent help, this thesis would not have been possible. I would also like to thank Dr.Mustafa Sabah Taha and Dr.Yazen Al-jerody for their technical support on my study and their invaluable advices. In particular, I would love to express my gratitude to my parents, for their love and support throughout my life, sister for her support and kindness and my brother, without their tremendous understanding and encouragement in the past few years, it would be impossible for me to complete my study. All of their love, support, and sacrifices over these past years gave me the strength and determination to make my dreams come true.

Tamara Saad Mohamed AL-JANABI

AKSARAY, 2022

TABLE OF CONTENTS

DOKTORA TEZİ

AKILLI ALGORITMALARA DAYALI NESNELERIN INTERNETI IÇIN DAYANIKLI IDS TASARIMI

Tamara Saad Mohamed AL-JANABI

Aksaray Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Sezgin AYDIN

ÖZET

Genel olarak Nesnelerin İnterneti'nin (IoT'ler) ve özellikle Hareketli Nesnelerin İnterneti'nin (IoMT'ler) güvenlik açıkları, araştırmacıları davetsiz misafirlere ve saldırılara karşı güvenlik sistemleriyle donatmaya motive eder. IoMT'ler için anormallik tespitinin izinsiz giriş tespiti ile entegrasyonu yeterince ele alınmamıştır. Bu çalışma, anormallik tespiti için bir Kalman ve Cauchy kümelemesi oluşturarak ve bunu Extreme Learning Machine (ELM) sınıflandırıcısını kullanarak IoMT'ler içindeki kimlik doğrulama düğümleri için kullanarak bu sorunu ele almaktadır. Algoritma, çeşitli bileşenlerden oluşur. Bunlardan ilki, WiFi'yi IMU verileriyle birleştirmeye dayalı bir kapalı ortam içindeki yayaların yörüngesini tahmin etmek için Kalman filtresi tabanlı model, ikincisi Kalman filtresini kullanarak tahmini yörüngeye dayalı olarak IoMT'deki anormallik davranışını tespit etmek için güvenilirlik değerlendirmesi, üçüncüsü de, bir Online Sequential Extreme öğrenme makinesi (OSELM) kullanarak saldırıların tanımlanması için anormallik algılamayı çevrimiçi öğrenme ile entegre ederek IoMT sistemleri için IDS modelidir. OSELM algoritması, WiFi parmak izi için TamperU veri seti ve izinsiz giriş tespiti için KDD99 kullanılarak uygulanmış ve değerlendirilmiştir. Ayrıca, izinsiz giriş tespiti ve anormallik tespiti için karşılaştırma ölçütleri ile yapılan bir karşılaştırma, önerilen tüm sınıflandırma ölçütleri açısından önerilen yaklaşımın üstünlüğünü kanıtlamaktadır. Geliştirilen algoritma, anormallik tespiti için mevcut iki modelle, yani gelişen veri akışı için bir çok yoğunluklu kümeleme algoritması (MUDI) ve gelişen veri akışlarının rastgele şekillendirilmiş kümeler halinde tamamen çevrimiçi kümelenmesi (CEDAS) ile karşılaştırıldı. Sonuçlar, bu çalışmada geliştirilen algoritmanın, kullanılan anormalliklerin farklı yüzdeleri, farklı sayıda yaya sayısı ve farklı ortalama yaya hızlarını içeren üç farklı senaryo altında anormallik ve izinsiz giriş tespiti açısından üstünlüğünü kanıtlamıştır.

Anahtar Kelimeler: Kalman filtresi, Anomali Tespiti, Saldırı Tespit Sistemleri, Aşırı Öğrenme Makinesi, Hareketli nesnelerin interneti, Wi-Fi sensörleri, Kalman Cauchy.

Ekim , 2022; 138 Sayfa

Ph.D. THESIS

ROBUST IDSS DESIGN FOR IOT BASED ON SMART ALGORITHMS

Tamara Saad Mohamed AL-JANABI

Aksaray University
Graduate School of Natural and Applied Sciences
Department of Electrical Electronic and Computer Engineering

Supervisor: Prof. Dr. Sezgin AYDIN

ABSTRACT

The vulnerabilities of the Internet of Things (IoTs) in general and the Internet of Mobility Things (IoMTs) in particular motivate researchers to equip them with security systems against intruders and attacks. The integration of anomaly detection with intrusion detection for IoMTs has not been addressed adequately. This study tackles this issue through building a Kalman and Cauchy clustering for anomaly detection and using it for authentication nodes within IoMTs using the Extreme Learning Machine (ELM) classifier. The algorithm is composed of various components; firstly, the Kalman filter-based model for estimating the trajectory of pedestrians within an indoor environment based on fusing WiFi with IMU data. Secondly, trustworthiness assessment for detecting anomaly behaviour in IoMT based on the estimated trajectory using the Kalman filter. Thirdly, the trust IDS model for IoMT systems by integrating anomaly detection with online learning for attacks identification using an Online Sequential Extreme learning machine(OSELM). The OSELM algorithm has been implemented and evaluated using TamperU dataset for WiFi fingerprinting and KDD99 for intrusion detection. Furthermore, a comparison with benchmarks for intrusion detection and anomaly detection proves the superiority of the proposed approach in terms of all the considered classification metrics. The developed algorithm was compared with two existing models for anomaly detection, namely, a multi-density clustering algorithm for evolving data stream (MUDI) and fully online clustering of evolving data streams into arbitrarily shaped clusters (CEDAS). The results proved the superiority of the developed algorithm in this work in terms of anomaly and intrusion detection under three different scenarios that include different percentages of added anomalies, different numbers of pedestrians, and different average speeds of pedestrians.

Keywords : Kalman filter, Anomaly Detection, Intrusion Detection Systems, Extreme Learning Machine, Internet of mobility things, Wi-Fi sensors, Kalman Cauchy.

October , 2022; 138 Pages

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

Aps	Access points
CD	Contrastive divergence
CEDAS	Clustering of evolving data stream
CNN	Convolutional neural network
DBNs	Deep belief networks
DBSCAN	Density-based spatial clustering of applications with noise
DoS	Denial of service
DdoS	Distributed denial of service
E2E	End to end
ELM	Extreme learning machine
FA-OSELM	Feature adaptive online sequential extreme learning machine
IDSs	Intrusion detection systems
IMU	Inertial measurement unit
IoMT	Internet of mobile things
IoT	Internet of things
KC	Kalman cauchy
KC DBN	Kalamn-cauchy deep belief networks
KC ELM	Kalman-cauchy extreme learning machine
KDD99	Knowledge and descovery data minin
M-DBSCAN	Multi- density-based spatial clustering of applications with noise
ML	Machine learning
MUDI	Multi-density
NN	Neural network
OSELM	Online sequential extreme learning machine
OSI	Open systems ınterconnection
OWT	Outlier wight threshold
PDR	Packet delay ratio
PCD	Persistent Contrastive Divergence
QoS	Quality of service
RBMs	Restricted boltzmann machines
RSSI	Received signal strength ındex
UML	Unified modeling language
Wi-Fi	Wireless fidelity

1. INTRODUCTION

The constant development of the Internet of Things (IoT) technology has effectively aided society's progress in recent years. The IoT, which embeds wireless network connectivity, multi-sensors, and technology into traditionally non-smart everyday objects, is gradually making the concept of “smart objects” a reality. Thinking about the IoT, we usually think of stationary objects like appliances, home automation systems, or stationery products like the Good Night Lamp. Moving things, such as self-driving cars and robots, and those that walk beside us (mobile devices and wearables), are anticipated to be key components of the IoT. The IoMT (Internet of Moving Things) takes it a step further, connecting moving objects such as cars, phones, robots, and mobile devices (Althobaiti, 2020). Moving things (such as automobiles, buses, lorries, trains, people, wearable gadgets, mobile phones, and tablets) and others that can be tracked, exchanged, or interacted with bits of data over mobile network or Wi-Fi Internet connections are included in the IoMT. Because moving things are so important in modern life, it's only logical that they'd be linked to networks (Junfeng Tian et al., 2019). Considering the vital role that moving items play in our modern lives, such as delivering people and goods, it becomes only natural that vehicles like cars, trucks, and trains would become networked. Both companies and people are interested in learning more about location, fuel efficiency, and the relationship to other vehicles, and being able to track data across multiple environments provides a more complete picture of our lives, which helps companies design and create products that make more sense for people.

The application of anomaly detection in general has been addressed in different fields such as identifying abnormal behavior using Internet of Things (IoT), predictive maintenance in industrial environment, and detecting abnormal activities in networks (Gaddam et al., 2020), etc. However, one special type of anomaly detection has not received adequate attention from researchers is the anomaly identification in internet of moving things (IoMT). Considering the vulnerability of this networks in the aspect of falling under intrusion from outsiders, providing them with smart anomaly detection algorithm to assist the functionalities of intrusion detection is promising for increasing the security of them and alleviating the accuracy of attacks detection (Junfeng Tian et al., 2019) .

1.1 IoT Background

Security concerns are impeding the adoption of IoT technology in everyday business and life (Arshad et al., 2020). In comparison to conventional networks, the Internet of Things environment is dynamic (Ghaleb, Maarof, Zainal, Al-Rimy, et al., 2019). As a result, standardizing a clear visualization for its topology is difficult. The vast majority of traditional security measures are incompatible with the Internet of Things paradigm (Moustafa et al., 2018). Furthermore, memory (Arshad et al., 2020) CPU, and power capacity are all limited in many IoT sensors and actuators. Due to these constraints, traditional security measures are computationally prohibitive. Furthermore, a new issue arises when attackers compromise susceptible nodes and exploit them to initiate a series of pipelined attacks from within the internal network. The high proportion of heterogeneous Internet of Things devices presents an additional challenge in this situation.

Despite the fact that IoT devices differ in several ways, They must strive to achieve a shared task, such as , computational capacity, functionality , network connectivity and software specs (Granjal et al., 2015). Regardless, when the participating nodes use different protocols and standards, the heterogeneity of IoT nodes has an impact (Borgia, 2014). As a result of a lack of consistency among many of the protocols, traditional solutions are unable to cooperate with one another (Chen et al., 2018; Qadri et al., 2020). As a result, determining the presence of compromised nodes and/or attacks becomes difficult. Furthermore, the security levels of the various protocols vary. As a result, protocols with weak security pose a threat to the entire IoT system.

In order to deal with IoT security concerns, several solutions have been developed, which can be classified into two categories: detection and prevention (Singh and Kumar, 2020). Prevention is the first line of defence against an attack on the system or its data. To ensure that only those with the proper authorization have access to IoT resources, these solutions employ cryptography and biometrics (Allig et al., 2019). However, if the attacks were launched from a valid (but compromised) node within the network, the preventive measure is ineffective. In this scenario, the attacker has complete control over all resources, including encryption/decryption keys and login credentials. This compromised node appears legitimate to the rest of the network,

allowing the attacker to carry out malicious actions such as data falsification and manipulation (Albahri et al., 2021).

Insurance firms, for example, can track real-time driving behaviours and offer discounts to good drivers (and, presumably, raise the rates of bad drivers). Audi's Traffic Light Detection technology adjusts the vehicle's speed to match the speed of traffic lights, saving time and fuel (de Souza et al., 2015). On the Internet of Things, the Internet of Moving Things has gathered data on communication and the link between moving sensors and servers. Moving items data, such as self-driving cars, people, freight, robots, and drones, are generated swiftly in real life. These moving objects can collect a vast amount of related trajectory data via various sensors and upload it to a server over the network, where we can evaluate and detect anomalies. The Internet of Moving Things poses enormous challenges for city leaders today (Teng et al., 2019), but it also offers significant opportunities to leverage the mobile-technology boom to improve everything, from city services to air quality, as well as provide new insights into public safety and long-term urban planning and air quality forecasting (Wang et al., 2019). When the above moving things move, position information is generated in the form of a trajectory, and multi-sensor data is generated at the same time. Human examination of these moving entities, particularly outlier analysis, has significant practical implications. People normally compare moving objects as a whole, and the complete moving object is the basic unit of outlier detection. We may not be able to detect outlying portions this way and use them for raising the security level of intruders in the network. This motivates developers of systems to integrate anomaly detection of IoMTs with IDSs to increase security and minimize vulnerability. Figure 1.1 presents the usage of IoMTs for detecting anomalies of vehicles driving on the highway.

Figure 1.1. Internet of Moving Things (IoMTs) sensors are useful for detecting anomalies in the roads to mitigates hazards (Wang et al., 2019).

The application of anomaly detection, in general, has been addressed in different fields, such as identifying abnormal behaviors using IoT (Gaddam et al., 2020), predictive maintenance in industrial environments (De Benedetti et al., 2018), and detecting abnormal activities in networks (Salman et al., 2019), etc. Considering the vulnerability of these networks in the aspect of falling under intrusion from outsiders, providing them with a smart anomaly detection algorithm to assist the functionalities of intrusion detection is promising for increasing their security and alleviating the accuracy of attacks detection. The goal of this study is to develop an anomaly detection algorithm and IDS for IoMT. This algorithm exploits the mobility pattern of the individual nodes in the network by using Kalman filtering-based sensor fusion. The utilized sensors are WiFi and inertial measurement units carried by the pedestrians for estimating their trajectories. Next, the predicted trajectories are used as stream data input for the Cauchy-based clustering algorithm for anomaly detection. The anomaly detection is finally used to extend the feature space of the intrusion detection algorithm based on the extreme learning machine model.

Literature evidence suggests that various approaches have been developed for anomaly detection based on stream data clustering, such as those proposed by (Islam et al., 2019; Amini et al., 2016; (Škrjanc et al., 2018); (Weng & Liu, 2019). However, all these proposals were only made for clustering with identifying outliers (Amini et al., 2016); (Islam et al., 2019) for intrusion detection, but without exploiting special characteristics about the networks in general or the mobile nodes in IoMTs in particular (Škrjanc et al., 2018). On the other side, it is observed that the

linkage between the anomaly behaviour of mobile nodes and detecting the capabilities of intrusion detection systems in IoMTs has not been exploited nor addressed. In the work of (Junfeng Tian et al., 2019), the moving things outlier detection algorithm has been proposed for anomaly detection in IoMTs. The distance of the moving things is equal to the weighted sum of the location distance and the multi-sensor distance; a three-step framework is used to detect the generalized anomaly using multi-sensor data generalization, moving things partitioning, and anomaly detection. However, it misses a linkage with an IDS for securing the network against attacks. Furthermore, its anomaly detection does not provide integration between process models and measurement models for state estimation which is needed for using an accurate estimation in anomaly detection.

Outsider attack detection is concerned with defending the IoT system against attacks from outside the network (Al-rimy et al., 2018); (Moustafa et al., 2018). This is usually accomplished by placing the IDS on the network's perimeter. As effective as this strategy may be in preventing external attacks (from the Internet, for example) from harming an IoT system, it has no effect on an attack initiated from within the system. It's possible for an attacker to avoid detection in this instance because it doesn't have to enter the network. As a result, the detection system would have no idea what was going on.

This technique defends against attacks initiated from within the network itself (Ghaleb et al., 2018; (Xia et al., 2019). In order to launch a series of attacks against the network, the attacker must employ legitimate but compromised nodes, which makes it harder to detect by existing solutions. Because the insider attacks are frequently performed out through authorized (trustworthy) nodes that the attackers have managed to corrupt, it becomes more difficult to identify the attacks.

Insider attacks have been the subject of several proposals for solutions (Thanigaivelan et al., 2018); (B. Ghaleb et al., 2018)(F. A. Ghaleb, Maarof, Zainal, Al-Rimy, et al., 2019); (Foley et al., 2020). These solutions attempt to identify discriminative patterns that can be used to distinguish attack traffic from normal traffic. By watching the traffic transferred between neighbouring nodes, such as packet size and data rate, these systems can identify any network inconsistencies. It's possible that this approach could help identify resource exhaustion and spoofing

attacks, but it can't identify attacks that change data and spread misleading information to other nodes in close proximity (F. A. Ghaleb, Maarof, Zainal, Al-Rimy, et al., 2019).

Researchers assume that attackers maintain a consistent threat regime and never change their behaviour in these investigations. Many of these solutions are unaware of the complex obfuscation techniques employed by skilled attackers and malicious software to evade detection. When it comes to devising solutions, they assume that there is a steady stream of attacks on their systems. Because of the dynamic nature of the Internet of Things (IoT), advanced attackers are adapting their attack methods to escape detection. As a result, the defence systems become obsolete because of concept drift.

1.2 Problem Statement

Reading the literature, it is observed that various approaches were developed for anomaly detection based on stream data clustering such as the work of (Islam et al., 2019); (Amini et al., 2016) (Škrjanc et al., 2018); (Weng & Liu, 2019). However, all of them were only made for clustering with identifying outliers (Amini et al., 2016) (Islam et al., 2019) or for intrusion detection but without exploiting special characteristics about the networks in general or the mobile nodes in IoMTs in particular (Škrjanc et al., 2018). On the other side, it is observed that linking between anomaly behaviour of mobile nodes and detecting capabilities of intrusion detection system in IoMTs has not been exploited nor addressed. In the work of (Junfeng Tian et al., 2019), The moving things outlier detection algorithm has been proposed for anomaly detection of the Internet of Moving Things. The distance of moving things, which is equal to the weighted sum of the location distance and the multi-sensor distance, and then apply a three-step framework to detect the generalized anomaly using multi-sensor data generalization, moving things partitioning, and anomaly detection. However, it misses a linking with an IDS for securing the network against attacks. Furthermore, its anomaly detection does not provide an integration between process model and measurement model for state estimation which is needed for using an accurate estimation in the anomaly detection.

There are six sub questions that must be answered in order to understand the research aim.

1. How to recognize the trustworthiness of transmitted/received data by IoMT nodes?
2. How to recognize the compromised nodes within IoMT system?
3. How to detect anomaly behaviour in IoMT based on the estimated trajectory?
4. How to generate IDs in IoMT environment ?
5. How to improve the detection accuracy?

1.3 Research Aim and Contributions

The aim of the proposed study is to develop an anomaly detection algorithm and IDS for IoMT. This algorithm exploits the mobility pattern of the individual nodes in the network by using Kalman filtering based sensor fusion. The used sensors are WiFi and inertial measurement units carried by the pedestrians for estimating their trajectories. Next, the predicted trajectories are used as stream data input to Cauchy based clustering algorithm for the functionality of anomaly detection. The anomaly detection is finally then used for extending the feature space of intrusion detection algorithm based on deep believe neural network model.

This Aim is accomplished based on the following contributions :

1. Create a framework for intrusion detection using mobility anomaly detection and trained model for attacks.
2. Develop a Kalman filter-based model for estimating the trajectory of pedestrians within an indoor environment based on fusing WiFi with IMU data.
3. Propose a Cauchy-based clustering method for detecting anomaly behaviour in IoMT based on the estimated trajectory by Kalman filter.
4. Propose an extreme learning machine-based IDS model for IoMT systems using the Cauchy clustering scheme proposed in (1) for the training phase of an extreme learning machine to improve the detection accuracy.
5. Evaluate the proposed algorithm using classification and networking metrics

1.4 Research Scope

The developed algorithm aims at detecting anomaly in general and IDS in particular, that happens in internet of moving things IoMTs.

1. The type of attacks that we are addressing are data tampering attacks. In data tampering, the attackers manipulate the user's information intentionally to disrupt their privacy using unwanted activities.
2. The IoT devices that carry important user's information such as location, fitness, billing price of smart equipment are in great danger to encounter these data tampering attacks (Junfeng Tian et al., 2019).
3. The considered sensors are devices occupied with accelerometers, gyros, GPS, and heart rate sensor (X. Li & Zou, 2021) .

Such attacks are threats to Identification, Authorization, Accessibility, Confidentiality, and Integrity.

1.5 Problem Formulation

Assuming that we have an indoor environment E combined of set of floors F given as:

$$F = \{f_1, f_2, .. f_n\} \tag{1.1}$$

Where

f_i denotes floor of ID i

n denotes the maximum ID for floors

And multi-room in each floor given as

$$R = \{r_{1,1}, r_{1,2}, .. r_{1,NR(f_1)}, r_{2,1}, r_{2,2}, .. r_{2,NR(f_2)}, ... r_{f_n,NR(f_n)}\} \tag{1.2}$$

Where

$NR(f_i)$ denotes the number of rooms in floor f_i

Set of Pedestrians denotes by

$$P = \{p_1, p_2, ... p_m\} \tag{1.3}$$

m denotes the number of Pedestrians.

The Pedestrians are expected to be moving within the environment while they carry smart device or nodes with an IoMT networking connectivity. The nodes contains WiFi sensor which enables receiving WiFi signal from various access points located at given locations in the environment. The set of access points is given as

$$AP = \{ap_1, ap_2, \dots ap_k\} \tag{1.4}$$

Where

k denotes the number of access points .

The WiFi signal is used to predict the location of the Pedestrians based on the fingerprint. Hence, we have IoMT network that is consisted of the Pedestrians devices. The goal is to build a system that uses the location information of the neighboring nodes for detecting anomaly behaviour in the mobility. Next, we secure the network with an IDS system that detects two types of attacks, namely, blackhole and DDoS.

1.6 Thesis Motivation

The application of anomaly detection in general has been addressed in different fields such as identifying abnormal behavior using Internet of Things (IoT), predictive maintenance in industrial environment, and detecting abnormal activities in networks, (Breitenbacher et al., 2019) etc. However, one special type of anomaly detection has not received adequate attention from researchers is the anomaly identification in internet of moving things (IoMT) (Gaddam et al., 2020). Considering the vulnerability of this networks in the aspect of falling under intrusion from outsiders, providing them with smart anomaly detection algorithm to assist the functionalities of intrusion detection is promising for increasing the security of them and alleviating the accuracy of attacks detection (Arshad et al., 2020).

1.7 Method of Study

This thesis tackles the issue of Vulnerabilities Internet of Moving Things (IoMTs) through building a Kalman (Fan et al., 2019) (Sung et al., 2018) and Cauchy

clustering for anomaly detection and using it for authentication nodes within IoMTs using extreme learning machine classifier (Škrjanc et al., 2018). The algorithm is combined of various components: firstly, Kalman filter-based model for estimating the trajectory of pedestrians within an indoor environment based on fusing WiFi with IMU data (X. Li & Zou, 2021). Secondly, trustworthiness assessment for detecting anomaly behaviour in IoMT based on the estimated trajectory by Kalman filter. Thirdly, trust IDS model for IoMT systems by integrating anomaly detection with online learning for attacks identification using online sequential extreme learning machine. The algorithm has been implemented and evaluated using TamperU dataset for WiFi fingerprinting (Lohan et al., 2017) and KDD99 for intrusion detection (Moustafa & Slay, 2015). Furthermore, a comparison with benchmarks for intrusion detection and anomaly detection proves the superiority in terms of all classification metrics.

1.8 Thesis Outlines

The rest of this thesis is organized as follows:

- Chapter two: Presents the Literature survey and related works
- Chapter three: Methodology
- Chapter four: presents the experimental design, results, and analysis
- Chapter five: presents the conclusion and future works.

2. LITERATURE REVIEW

Despite the crucial role of the Internet of Things (IoT) technology in our daily life and business activities, they have been associated with several security threats and concerns due to the heterogeneity and huge number of devices connected to the IoT, as well as the ineffectiveness of the conventional security protocols (Sha et al., 2020). Even though there are obvious differences in the connected devices in terms of their function, software application, computational capability; and network connectivity, it is necessary that they work together to achieve a target (Granjal et al., 2015). This implies that any fault in the security protocol of any of the connected devices will affect the performance of the entire system and prevent it from performing optimally. To ensure adequate protection of IoT systems, several security protocols have been developed; these security protocols can be categorized into reactive and proactive protocols. Regarding the proactive protocols, also called preventive approaches, they are developed to protect data and communications using authentication and cryptographic procedures (Chen et al., 2018). These preventive protocols can prevent attacks targeted at sensitive information and data on other nodes, but they cannot prevent a compromised node from propagating altered information with the rest of the network nodes (van der Heijden et al., 2018). On the other hand, the reactive protocols are developed to detect attacks and prevent them from occurring. One common example of the reactive protocols is the intrusion detection systems that identifies the existence of malicious network activities and identify the compromised devices.

A typical IoT network is a system of numerous devices and components that are interconnected to a network for effective exchange of data and information between such components as exhibited in Figure 2.1 (Lo et al., 2019). Such systems exhibit better performance in terms of effectiveness, efficiency and productivity. However, a major problem of such integration of components and devices is the issue of security threats. As these systems are connected to the Internet, they are prone to several attacks that can compromise the integrity, confidentiality, and authenticity of the exchanged data.

This literature provides a comprehensive explanation of Intrusion Detection Systems (IDS) in IoT and IoMT. Additionally, abuse-based detection and anomaly-based

detection which are the main classification of the Intrusion Detection System (IDS) had been explained in critical analysis.

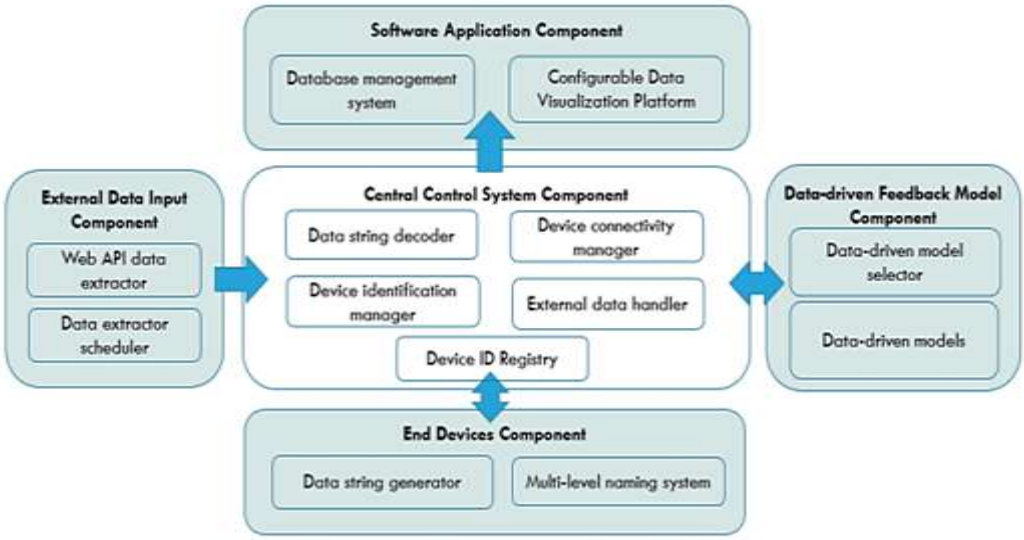

Figure 2.1. The main components of IoT architecture (Lo et al., 2019).

2.1 Anomaly Detection In IoMT

The existing literature contains a wide range of approaches developed for handling the problem of anomaly detection in general, and for the internet of mobile things in particular. From a general perspective, some approaches to anomaly detection were based on clustering. In the work of (Islam et al., 2019), a fully online density-based clustering algorithm for changing data streams was developed under the name of Buffer-based Online Clustering for Evolving Data Stream (BOCEDS). The micro-cluster radius is recursively updated to its local optimal using this approach. It also includes a buffer for storing unnecessary micro-clusters and a fully online pruning mechanism for extracting the buffer's momentarily irrelevant micro-cluster. Furthermore, BOCEDS presents an online micro-cluster energy-updating mechanism based on the data stream's spatial information. The algorithm contains an outlier buffer that can be used for rejecting anomaly behavior or detecting it. Other algorithms were based on the density change to detect anomaly behavior with the inclusion of the historical behavior of algorithms. In the work of (Amini et al., 2016), Multi-Density (MUDI)-Stream was developed; it consists of four basic components and an online-offline algorithm. It stores summary information about a growing multi-density data stream in the form of core mini-clusters during the online phase. The final clusters are generated using an adapted density-based clustering algorithm

in the offline phase. The grid-based approach is utilized as an outlier buffer to handle noise, anomaly behavior, and multi-density data while also reducing the clustering merging time. Other algorithms were based on mathematical density models. The study by (Škrjanc et al., 2018) introduced the eCauchy approach for classification issues, which is a unique evolving probabilistic Cauchy clustering method. This approach was used to monitor cyber-attacks on a broad scale. The described strategy results in a more flexible system for detecting attacks. In the work of (Weng & Liu, 2019), the statistical features of the subsequence of streams were considered. The work has provided an analysis of the differences between anomalies in single time series and anomalies in multi-dimensional time series. Next, it proposed a collective anomaly detection algorithm named iForestFS for a multi-dimensional stream based on iForest in a cloud environment. The Markov Jump Particle Filter was used in the work of (Slavic et al., 2020) to analyse multiple video situations where a semi-autonomous vehicle accomplishes a set of tasks in a closed environment. The implementation of an automated data engineering pipeline for anomaly detection in IoT sensor data was proposed in the work of (X. Li and Zou, 2021). The method employs IoT sensors, Raspberry Pis, Amazon Web Services (AWS), and a variety of machine learning approaches to discover anomalous cases in the smart home security system. The moving things outlier identification algorithm was proposed in (Wang et al., 2019) as a generalized approach for anomaly detection from the Internet of Moving Things. The distance of moving things was proposed as being equal to the weighted sum of the location distance and the multi-sensor distance; the generalized anomaly was detected using a three-step framework that included multi-sensor data generalization, moving things partitioning, and anomaly detection. A similarity metric Convolutional Neural Network (CNN) based on a channel attention model was proposed for traffic anomaly detection task in the work by Kang et al. (2019). The method consists mostly of (i) a Siamese network with a hierarchical attention model based on word embedding that can measure similarities between anomalies and templates selectively; (ii) a deep transfer learning algorithm can automatically annotate an unlabeled collection while fine-tuning the network; (iii) a background modeling method for anomaly extraction that combines geographical and temporal information. The association between the experience of sleep deprivation among drivers during lengthy journeys and CO_2 concentrations in automobiles was hypothesized in the study by (Chung & Kim, 2020). The assistant collects

multimodal signals using five sensors that detect CO, CO_2, and particulate matter (PM), as well as the temperature and humidity data. These signals are then sent to a server via the IoT where they are analysed by a deep neural network to determine the vehicle's air quality. To create an air quality anomaly detection model, the deep network uses long short-term memory (LSTM), skip-generative adversarial network (GAN), and variational auto-encoder (VAE) models. The deep learning models use LSTMs to acquire data, while the semi-supervised deep learning models use GANs and VAEs. The goal of this assistant is to deliver real-time vehicle air quality information to drivers, such as PM alarms and sleep-deprived driving alerts to prevent accidents. A summary of the related studies is presented in Table 2.1.

Table 2.1. A summary of the related studies of anomaly detection in IoMT.

Author(s)	Research Problem	Solution	Limitation(s)	Detection Type
Islam, et al. (2019)	Most of the algorithms are either fully offline, hybrid online/offline, or cannot handle the property of evolving data stream.	A fully online clustering algorithm for evolving data stream called CEDAS . The algorithm contains an outlier buffer that can be used for rejecting anomaly behavior or detecting it	CEDASrequires predefining the global optimal radius of micro-clusters, which is a difficult tasks	*Insider*
Amini et al. (2016)	Other algorithms were based on mathematical density models.	Multi-Density (MUDI)-Stream was developed; it consists of four basic components and an online-offline algorithm. It stores summary information about a growing multi-density data stream in the form of core mini-clusters during the online phase. The grid-based approach is utilized as an outlier buffer to handle noise, anomaly behavior, and multi-density data while also reducing the clustering merging time.	Streaming algorithms must have a consistent and minimal memory storage, as well as a short computation time. MUDI-overall Stream's complexity is determined by the complexities of its components.	*Insider*

Table 2.1. (Continued).

Author(s)	Research Problem	Solution	Limitation(s)	Detection Type
Škrjanc et al. (2018)	Developing cyber-attack monitoring methodology that can keep the classifier up to date.	Introduced the eCauchy approach for classification issues, which is a unique evolving probabilistic Cauchy clustering method. This approach was used to monitor cyber-attacks on a broad scale. The goal is to create an IDS that can categorize intrusions as well as regular behavior based on input variables. A label (Action) is assigned to each connection vector, indicating the sort of connection.	The method's current flaw is that it doesn't include a cluster merging mechanism, which might reduce the amount of clusters formed.	*Insider*
Weng and Liu (2019)	Due to the variations between anomaly detection in multidimensional and univariate time series data, there are numerous issues with collective anomaly detection for multidimensional streams.	Analysis of the differences between anomalies in single time series and anomalies in multi-dimensional time series. Next, it proposed a collective anomaly detection algorithm named iForestFS for a multi-dimensional stream based on iForest in a cloud environment.	The proposed method was just tested to see if it was suitable for collective anomaly detection in multidimensional streams, but it was not compared to other methods.	*Insider*
Slavic et al. (2020)	How anomalies were discovered and an algorithm was developed to detect anomalies at both the observation and prediction levels.	The Markov Jump Particle Filter was used in this work to analyse multiple video situations where a semi-autonomous vehicle accomplishes a set of tasks in a closed environment.	There are two types of abnormal situations: I Video sequences with previously unseen visuals; ii) Video sequences with previously seen images but new video dynamics/motions. This research was mostly focused on the first type.	*Insider*

Table 2.1. (Continued).

Author(s)	Research Problem	Solution	Limitation(s)	Detection Type
Li and Zou (2021)	The fast growth of System of Chip (SoC) technology, the Internet of Things (IoT), cloud computing, and artificial intelligence has opened up new avenues for improving and fixing current issues.	An automated data engineering pipeline for detecting anomalies in IoT sensor data is investigated and proposed. The method employs IoT sensors, Raspberry Pis, Amazon Web Services (AWS), and a variety of machine learning approaches to discover anomaly cases in the smart home security system.	It's employed in an anti-theft home security system in a closed environment for an automated data pipeline for IoT sensor anomaly detection.	*Insider*
Wang et al. (2019)	People's physical and emotional wellbeing are adversely harmed by poor air quality. Many practical elements will influence the change in smog concentration, and it will have nonlinear properties.	Approach for anomaly detection from the Internet of Moving Things. The distance of moving things was proposed as being equal to the weighted sum of the location distance and the multi-sensor distance; the generalised anomaly was detected using a three-step framework that included multi-sensor data generalisation, moving things partitioning, and anomaly detection.	It's possible that the data properties in this investigation weren't evident, or that the data varied so much that no acceptable model could be trained.	*Insider*
Kang et al (2019)	Because the number of traffic anomalies is so small, typical deep learning approaches suffer from a severe over-fitting problem.	For traffic anomaly detection, the study suggested a similarity metric Convolutional Neural Network (CNN) based on a channel attention model.	Used to tackle the shortfall of training data and overfitting.	*Insider*
Chung and Kim (2020)	Traffic accidents can be caused by sleeplessness (fatigue) and decreased cognition, and the number of traffic accidents caused by driver sleepiness and exhaustion is increasing every year.	The assistant collects multimodal signals using five sensors that detect CO, CO_2, and particulate matter (PM), as well as the temperature and humidity data. These signals are then sent to a server via the IoT where they are analysed by a deep neural network to determine the vehicle's air quality.	Excess notifications are not taken into account in this model.	*Insider*

2.2 IDS-related Works For IoT

The existing network intrusion detection systems can be divided into misuse-based and anomaly-based detection systems. To build a detection model for misuse-based IDS, they rely on already known attack signatures, however for anomaly-based IDS, they include identified patterns of the regular activities and treat all other patterns as malicious. These two categories of IDS are further detailed in the following sub-sections.

2.2.1 Misuse-based IDS

The misuse-based IDS contain patterns of already known attacks and new patterns that match such patterns are considered as attack. Various studies have proposed and developed misuse-based IDS; for instance Yang et al. (2018); developed an IoT intrusion detection technique based on active learning which relies on the human-in-the-loop concept to combine human intelligence with machine learning to ensure data sufficiency. The study addressed the issue of collecting numerous data for the training phase due to the limited wireless channel capacities between different IoT components by ensuring that only the helpful 'unlabeled' data are labeled. The developed approach iteratively executed a supervised technique on the labeled data, selected the following group of data from the unlabeled data set for labeling by the expert. Then, the recently labeled data are added to the ground truth for the training of the supervised learning technique. This process is repeated until a predefined level of accuracy is reached. However, the training process requires human intervention which makes it labor and time intensive and susceptible to error.

A real-time IDS called SEVLETE was developed by Raza et al. (2013); this system provides end-to-end security in 6LoWPAN networks using message security technologies such as IPSec and DTLS. The model was developed for the detection of routing attacks, such as spoofing, alteration, and sinkhole attacks. Hence, it can protect data from manipulation by identifying the malicious nodes in the network. One problem with this system is that it does not offer protection against tampering of data contained in the node itself (as in situations of a legitimate node been manipulated) and cannot stop the altered node from disseminating false information as shown in Figure 2.2.

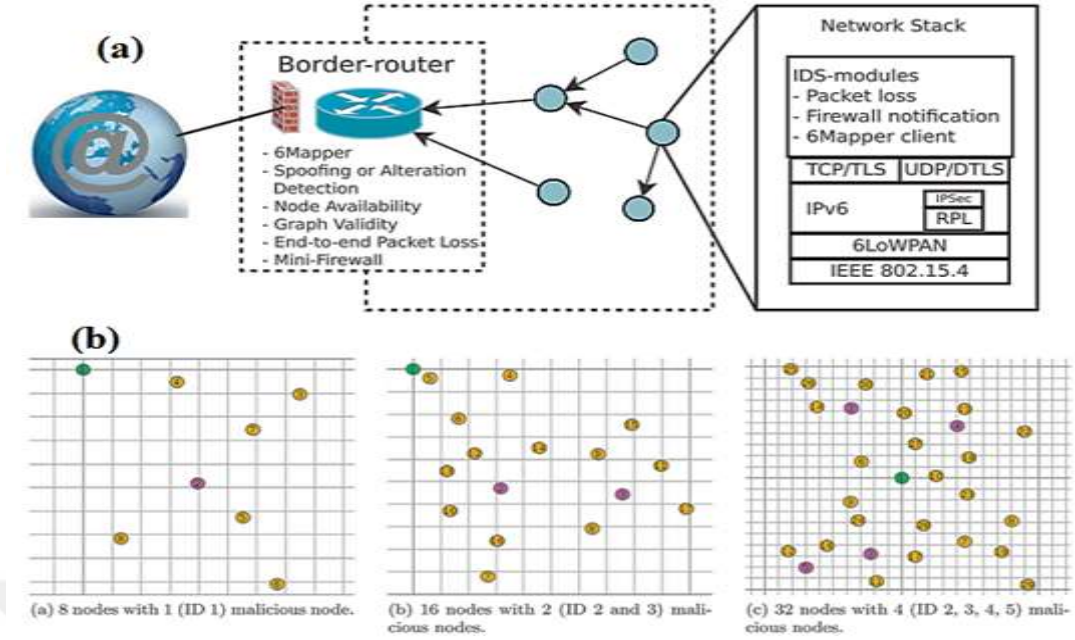

Figure 2.2. IoT setup using a) IDS modules b) Node placement according to the experimental results (Raza et al. 2013).

Moustafa et al. (2018) came up with an ensemble model (statistical flow features-based) for the protection of network traffic in IoT infrastructures. One feature of this approach is that it can aggregate the flow of traffic while ensuring the removal of the redundant observations those results from flooding attacks, thereby improving the efficiency and effectiveness of the proposed system. The first phase of the model is the extraction of a set of specialized features from the network protocols, such as MQTT, HTTP, and DNS. Then, the extracted features are employed for the training of several machine learning frameworks, such as Artificial Neural Networks, Naïve Bayes, and Decision Tree. During the training, the individual decisions serve as the basis for the production of the final decision via a voting strategy. This increases the diversity of the classification strategies and encourages assessment of the network traffic from various perspectives, thereby improving the detection accuracy of the system. Meanwhile, the proposed system requires the presence of mobile agents that must be activated individually at a time. One drawback of this approach is the inability to identify the coexistence of the context information generated by the nearby nodes of the nodes that are currently active. The coexistence of such information could be useful in the detection of complicated botnet attacks that work evasively and cooperatively to deliberately alter the current limits of the normal traffic profile.

Zhou et al. (2021) presented the first automata-based IDS for use in IoT; this novel approach focused on addressing the issue of heterogeneity in IoT nodes by striving towards building a unified security solution for the whole network. The proposed method extended the Labeled Transition System to uniformly describe the whole components of the IoT system. Terms and graphs were used to provide a characterization of the amalgamation of these components in the form of abstracted actions. Then, the comparison of the new observations with the already built profile was done using these abstracted actions. However, increases in the heterogeneity of the components increases the sparsely and dimensionality of the data owing to the variations amongst the features of the different components, making the devising of a unified representation a tedious task.

Arshad et al. (2020) proposed a lightweight IDS for energy conservation and preservation of communication overhead in IoT nodes with limited resources. In the proposed model, the nodes work together by sharing attacks info with an edge device which serves as the major IDS component as shown in Figure 2.3. Sustenance of network resources was ensured by allowing detection at both edge device and local nodes levels. However, the information propagated by the IoT nodes are not wholly trustworthy owing to the possibility of some of the nodes being manipulated and used by attackers for the transmission of adulterated or misleading information. Furthermore, internal attackers that sabotage the network from within using some compromised nodes may not have to worry about edge-based detection because it will be bypassed. Hence, the performance of the proposed system in attacks detection will be significantly reduced.

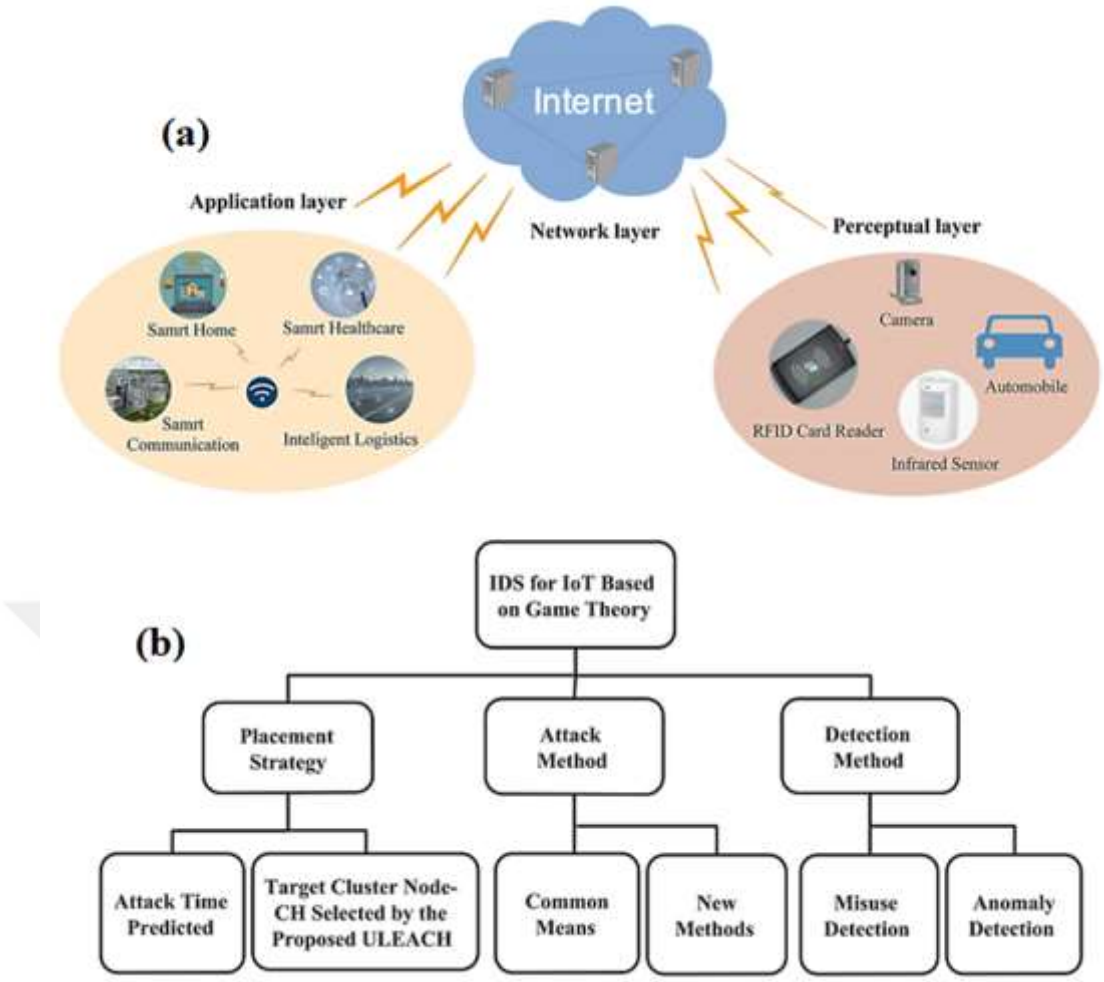

Figure 2.3. The architecture of the a) IoT b) Suggested framework (Li and Zou 2021).

Arshad et al. (2020) proposed a novel system for effective intrusion detection in machine-to-machine networks that does not impose significant energy and connectivity costs on the participating host and edge nodes. According to the results, the proposed solution has a low overhead in terms of energy usage and memory consumption. To overcome challenges of system are security problems, abnormality, and service failure, Atul et al. (2021) proposed an effective framework of energy aware smart home. The study employed the machine learning technique to distinguish the abnormality sources of the contact model. The said study that the result showed an 85% accuracy rate. Prabhakaran and Kulandasamy, (2021) proposed an attention-based recurrent convolutional neural network to detect intrusion or non-intrusion in text data. According to the authors, the proposed intrusion detection and safe data storage mechanism is highly secure and is never compromised by any kind of conspiracy attack. Tally and Amintoosi, (2021)

suggested a hybrid genetic algorithm approach for detecting intrusion and compared it to the traditional method. The proposed method outperformed the conventional method, according to the results.

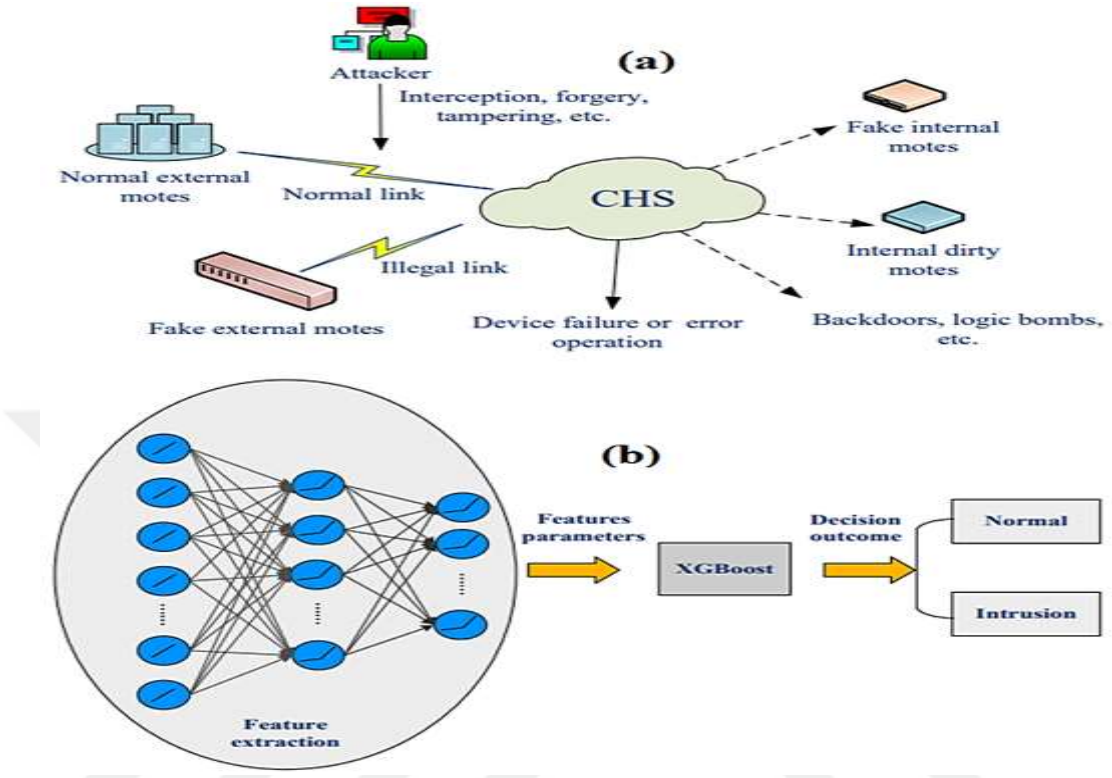

Figure 2.4. Novel IDS a) Security threats b) Proposed framework (He et al., 2019).

Based on a stacked auto encoder, He et al. (2019) proposed a novel intrusion detection scheme as shown in Figure 2.4. Despite the positive results of the proposed method, the authors pointed out that there is a gap in detection ability between lightweight intrusion detection and the deep learning method. To provide an easy-to-use security vulnerability checking and analysis solution for the IoT related developers and users, Hong et al. (2017) designed and enhanced an interface for a web based security analysis software for the internet of things. Halder et al. (2019) devised a new strategy for overcoming the limitations of both uniform and Gaussian deployments for energy-efficient and quick detection. The authors thoroughly investigated the effects of various network parameters on detection probability and concluded that when compared to other methods, the detection probability increased by more than 25%. Han et al. (2015) suggested a motion detection and tracking system based on passive radio frequency recognition tags for efficient intrusion

detection. Twins is accurate in detecting moving objects, with very low position errors of 0.75 m on average, according to the findings as shown in Figure 2.5.

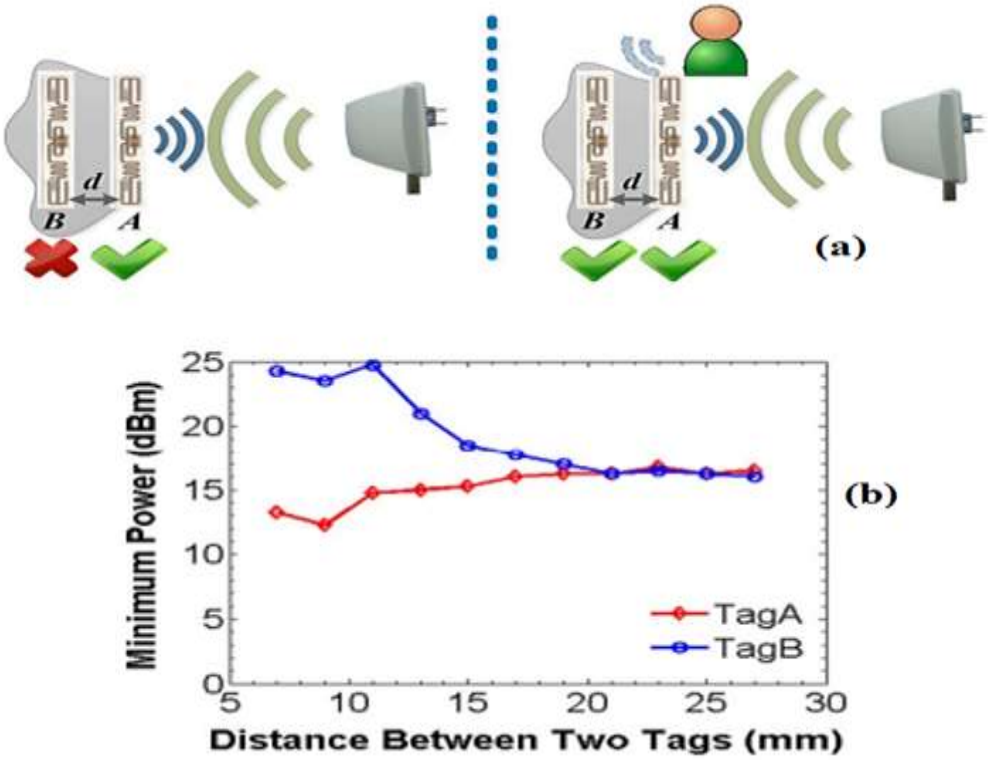

Figure 2.5. Accurate in detecting moving objects a) Critical state of Twins b) Minimum power vs. distance (Han et al., 2015).

Keung et al. (2012) investigated the intrusion detection issue in a mobile sensor network with an emphasis on k-barrier coverage against moving intruders. The study found that when compared to a static sensor area, coverage efficiency can be enhanced by an order of magnitude with the same number of sensors.

Jie Tian et al. (2014) proposed a hybrid deployment of mobile and static sensors to detect smart intruders with the purpose of crossing the monitored domain. The findings demonstrated that the proposed scheme would achieve a high detection probability while consuming little energy from mobile sensors. In another respect, to minimize the distance an external attacker would travel before being detected, Katneni et al. (2012) suggested a hybrid Gaussian-ring deployment. On the other hand, the results of proposal presented by Hu et al. (2014) regarding the quality of the sensor in the network, uncovered that the network's lifetime has improved by 17.4%, while the delay in detection has decreased by 101.6%. Shafiq et al. (2020) used an objective soft set technique to pick successful features, as well as a new

feature selection metric. The experimental findings obtained by proposed algorithms are promising, with more than 95% accuracy.

The study by Kumar et al. (2020), presented a self-adaptive misuse-based IDS for dynamic IoT platforms, especially networks that permits nodes to frequently join and leave the network. The proposed system relied on self-taught learning for the classification of the unknown events into harmful and non-harmful classes while features extraction from the observed instances was based on deep learning. Then, the extracted features were used for the determination of the event category though this adaptive approach lacks a system of determining the status (harmful or non-harmful) of the source of the unknown event, providing the room for the compromised node to disseminate harmful information with features that seem to be non-harmful. This adaptation process could cause a major deviation from the normal network profile and could be exploited by attackers to implement undetected attacks using the altered network profile. An IDS has been provided by (A. Li et al., 2020), for the protection of the perceptual layer of IoT systems owing to its proneness to numerous attacks, as well as the inability to implement a comprehensive IDS due to scarcity of resources. In the proposed solution, a clustering-based model was deployed on certain cluster heads for resource conservation while the model was built using Game theory and PSO. The model consolidated both misuse-based and anomaly-based approaches to enable high level attacks detection accuracy and low false alarms. However, the system is prone to internal attacks that can directly reach the IoT nodes because the deployment of the detection model on the cluster heads means that the scrutinization at the cluster heads will be bypassed.

Attacks on IoT systems are mainly launched using sophisticated strategies that can implement sustainable attacks like botnets and malware. Hence, researches have been devoted to the development of approaches for the prevention and detection of such attacks. The available approaches for attacks detection and prevention can be classified into anomaly-based, misuse-based, specification-based techniques as discussed further in the following sub-sections.

2.2.1.1 Drawbacks of misuse-based IDS

Misuse-based IDS are developed to only detect known attacks and not previously unknown attacks. This means that the misuse-based IDS have low accuracy level in the detection of zero-day attacks. Furthermore, attackers have access to more resources, and are highly intelligent to the point that they are also deploying measures to evade detection (Singh and Kumar, 2020). Most of the current attack forms cannot be accurately identified by the conventional misuse-based IDS; hence, there is a need to continuously update the existing attack detection methods to be at par with the trends in cyber-attacks (Al-rimy et al., 2018). Hence, misuse-based IDS cannot efficiently detect novel and emerging attack types.

2.3 Anomaly-based IDS In IoT

The anomaly-based IDS are developed with a normal profile of the normal system behavior for comparison and identification of abnormal system patterns. New network profiles are considered malicious if they deviate significantly from the already established normal profile, else, they are considered benign. The anomaly-based IDS for IoT are classified into those for external attack detection and those for internal attack detection; this classification is based on the source of the attacks. External attacks can be prevented using the conventional security techniques, such as encryption and intrusion prevention techniques, but for internal attacks, they are not easily detected using such traditional mechanisms. Hence, various studies have been devoted to the development of measures to counter such attacks and the available solutions are classified into those for protection against external attacks and those for protection against internal attacks. Protection against external attacks is aimed at protecting the system from attacks launched from outside the network while internal attack detection aims to prevent the success of attacks launched from within the network. The major problem with internal attacks is that the attacker launches the attack using legitimate but compromised nodes from the targeted network in a manner that cannot be detected by the existing detection methods (Li et al., 2019).

Several studies have been conducted on the detection of external and internal attacks on IoT networks and most of these studies proposed data-driven solutions that requires the collection of data during normal network operation or during attack

moments to build the detection models. With these data, models of the normal and attack profiles of the networks have been developed using several machine learning and AI algorithms to enable the detection and calculation of the parameters and thresholds.

Li et al. (2019) demonstrated that using blockchain technology to create a signature database will help improve the robustness and efficacy of signature-based intrusion detection systems in adversarial scenarios. Li et al. (2018) proposed a compact but efficient message verification approach based on collaborative intrusion detection network. The findings uncovered that suggested approach can identify malicious nodes. Similarly, Madsen et al. (2018) examined the influence of intrusion sensitivity in a simulated collaborative intrusion detection network environment as shown in Figure 2.6.

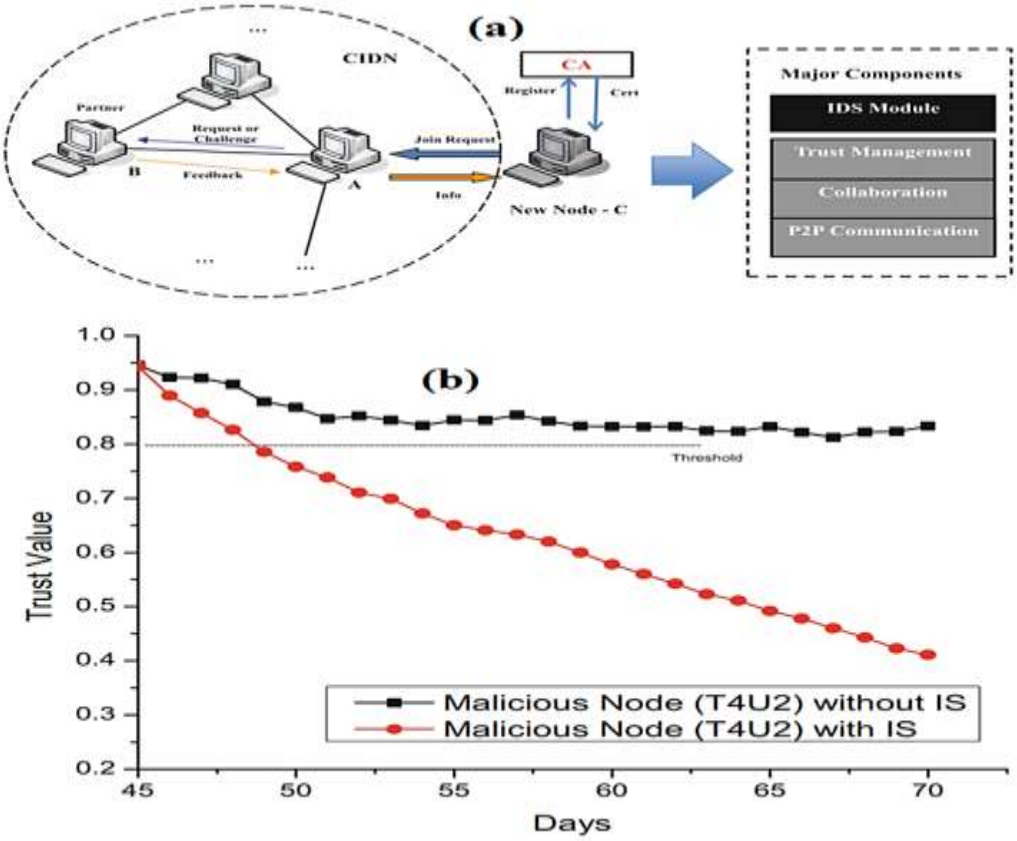

Figure 2.6. Improvement of the detection capabilities a) The typical high-level architecture of a challenge-based CIDN b) Trust values of malicious nodes (Madsen et al., 2018).

To improve the detection capabilities of a single intrusion detection device, Li et al. (2018) used collaborative intrusion detection networks. The outcomes demonstrated

that the proposed mechanism can detect malicious nodes. Based on anomaly and specification intrusion detection modules, Bostani and Sheikhan (2017) proposed a novel real-time hybrid intrusion detection framework. According to the results, the proposed hybrid approach achieved a true positive rate of 76.19%. Yahyaoui et al. (2019) suggested a novel anomaly protocol to combine machine learning detection with a statistical method for malicious node localization. The authors claimed that the proposed protocol had a high accuracy rate of more than 95%. Farzaneh (2019) introduced an anomaly-based lightweight Intrusion Detection System that was completely effective in detecting attacks and applicable to large-scale networks as shown in Figure 2.7.

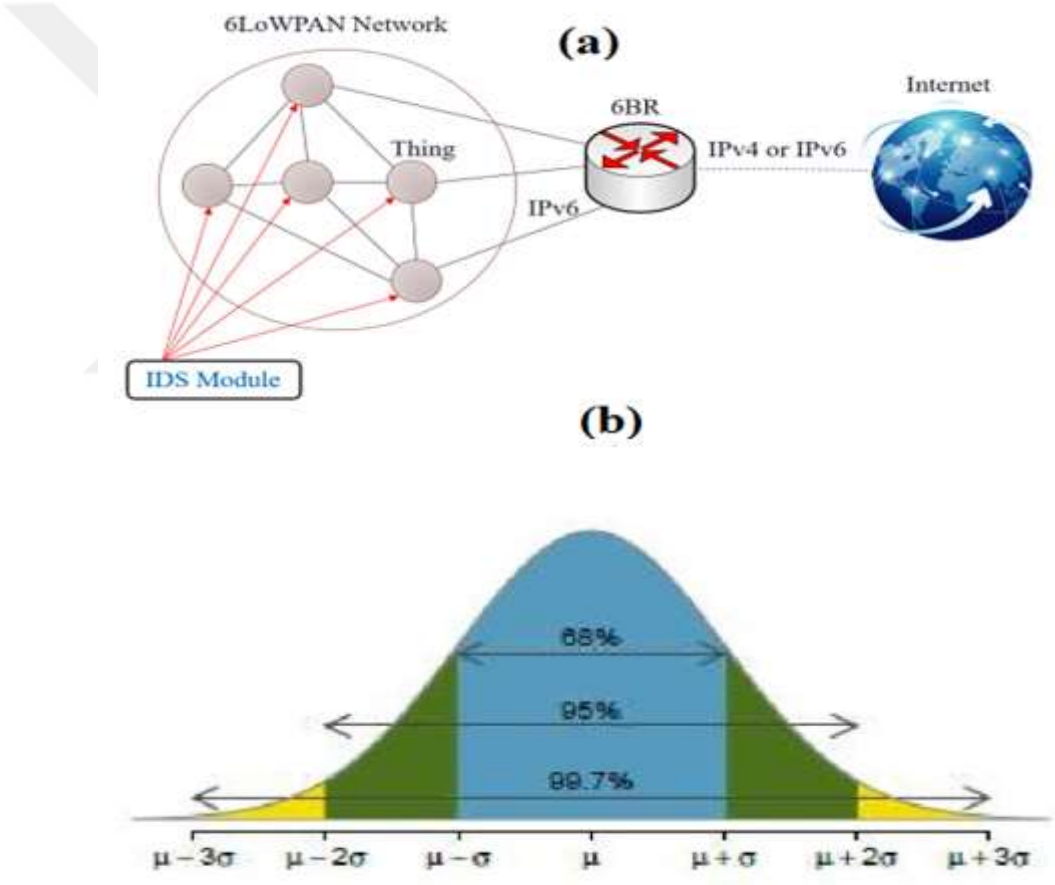

Figure 2.7. Improved an IDS a) Proposed model b) Normal Distribution (Farzaneh et al., 2019).

In the same way, Tama et al. (2019) improved an intrusion detection system based on hybrid feature selection and two-level classifier ensembles. The suggested model had an accuracy of 85.8%, a sensitivity of 86.8 %, and a detection rate of 88.0 %. Bagaa et al. (2020) proposed a new machine learning-based security architecture that

combines a monitoring agent and an AI-based reaction agent. The authors investigated existing possible solutions and claimed that the proposed model achieved 99.71 % accuracy in detecting anomalies. Nõmm and Bahşi (2018) reported that using Anomaly-based detection; it is possible to induce highly accurate unsupervised learning models with small feature set sizes. Verma and Ranga (2020) studied the prospects of using machine learning classification algorithms for securing Internet of Things against denial-of-service attacks. The authors employed Friedman and Nemenyi tests in their analyses and reported that classification and regression trees both are the suitable choice for building Internet of Things specific anomaly-based intrusion detection systems.

2.4 Detection of External Attacks

External attack detection techniques, as earlier stated, were developed for the detection and prevention of external network attacks. Such techniques are normally deployed at the main points of connection between the internal network and the external environment, such as at the cluster heads, gateways, and fog layer.

Akyildiz et al. (2006) developed a two-layer anomaly detection system for the automation systems in smart homes. The detection responsibility in the systems was divided between the service provider and the service user. On the side of the user, data collection is done at the local nodes and collected at the home gateway (HG). Being that the network traffic is monitored by the HG, the detection responsibility can be easily transferred to the HG to conserve the available computational resources at the nodes. Attack detection is done via comparison of the suspected pattern from a specific node with the related normal pattern already built for that node. On the side of the service provider, the anomalous pattern is further analyzed at the HG to establish similarities across the network. Hence, it is possible to detect similar abnormality by the other HGs, thereby helping in early detection of the attacks at the other premises. The study, however, assumes a stationary sensor and nodes operating environment, as well as limited data exchange between the nodes communication within the system; they also assumed a stable packet loss and small payload. This is not applicable to applications that co-interact and work collectively as the pattern of the whole system can be affected by a change in the pattern of one sensor. Furthermore, the HG only makes use of the packet header and some statistics on

packet information and this is not complex enough for efficient and accurate attacks detection.

A deep learning-based anomaly detection model has been presented by Meidan et al. (2018) for the detection of evasive botnets that alters their behavior upon every attack on IoT nodes. The extracted statistical features from the snapshots of the pattern of benign traffic are used by the model for the training of the deep auto-encoder model for every IoT device and for building the normal node profile. This profile contains the compressed representation at the potential layer of the auto-encoder and this profile serves as the basis for the determination of the threshold separating the normal & malicious patterns. The respective features vector is applied at the detection time on the observed snapshot. Then, the auto encoder attempts reconstruction of the snapshot and if failed, the snapshot is considered malicious. However, the behavior of the IoT is assumed to be stationary in the proposed model. However, this assumption is not applicable to most IoT systems whose behaviors are subject to changes with time. Therefore, the performance of the model and its detection accuracy will be affected as a result of reliance on fixed thresholds. The model further supposes that all the considered snapshots come with a total set of features but this may not be true as many factors are responsible for the availability of those features, such as the running environment, the capacity of the network, the availability of data, the absence of bad sensors, and the presence of noise.

The study by Breitenbacher et al. (2019) developed an anomaly-based IDS for use on IoT components following the whitelisting approach; this approach stipulates a range of legitimate applications that can be executed on the IoT node in a manner that only the listed applications it is possible to run on the nearby IoT node. Hence, when malicious software uses obfuscation techniques to hide its true pattern, the model can detect these adaptive attacks by making them resemble the benign ones. The issue with this technique is its reliance on rigid whitelist that cannot be possible in dynamic IoT platforms. Furthermore, there is no guarantee that the whitelist can be exhaustive because it might contain some of the legitimate programs. There is a chance of updating the whitelist with newly discovered programs during the operational stage, but this can only be done if deemed necessary by the network operator.

GARUDA was proposed by Aljawarneh and Vangipuram (2020) as a Fuzzy Gaussian dissimilarity feature section method when building IDS for IoT devices. The highpoint of the proposed method is that it addressed the issue of standard similarity measures, such as Euclidean distance which has no upper boundary or cosine similarity that has no consideration of the magnitude of the vectors when determining the similarity between two vectors. With GARUDA, training data dimensionality is reduced by clustering the features incrementally; this ensures that only a small set of unique features are selected. The determination of the number of clusters is based on a threshold that represents the allowable limit of intra-cluster features dissimilarity. However, the study considers all the data generated by the IoT devices to follow the Gaussian distribution, which is not applicable in most cases, especially in an environment that is highly dynamic. Hence, the proposed GARUDA may not perform the dissimilarity calculations accurately.

Pamukov et al. (2018) relied on the Negative Selection Algorithm (NSA) for normal behavior modeling using self and non-self-strategies. The proposed model comprised of two layers to ensure that the resource-constrained IoT nodes are protected from the burden of training since the normal training set is built by the first layer using the NSA, while the second layer relies on this training data to train the Neural Network (NN) algorithm. However, the study assumes a stationary definition of self and non-self which may not be applicable to IoT applications as their behavior evolve with time due to the dynamicity of IoT environments.

A fog-based IDS was proposed for IoT devices by Lyu et al. (2017) . The study strived for early detection of anomalies in IoT nodes by deploying hyper-ellipsoidal clustering at the fog layer in consideration of the based on the collected data at the nodes level. Hence, the processing overhead is relayed into the fog node for real-time accurate detection while ensuring low overhead on the IoT nodes. Each node is programmed to send data to the fog node at every predetermined time window either to implement sensor-level clustering by dealing with the data gathered by each sensor or fog level clustering by looking at the data together. Then, the new observations are categorized based on such clustering into anomalous or benign. The problem of this approach is that it did not consider the possible correlation that may exist between the fog node-connected IoT nodes; this correlation can be useful in the

propagation of the information about emerging attacks launched against one or more nodes.

A three-layer supervised anomaly-based IDS was proposed by Anthi et al. (2019) for IoT devices in smart homes. These layers tallied with the three consecutive layers (i.e., data link, network and transport layer) in the network OSI framework; this is to ensure the collection of the information related to the data units of these 3 layers and their subsequent use in training several ML classifiers, such SVM, NB, and DT for detection attack. The IoT system's usual network behavior was built by connecting all of the participating nodes through a single hub. The combination of the data from the 3 network layers ensures that the model can enrich the training dataset and perform accurate detection. But the issue with this method is that the extracted information is only the metadata for the segments, frames, and packets, meaning that the model may not accurately detect sophisticated attacks that involves data manipulation and legitimate nodes impersonation.

Moustafa et al. (2018) presented ensemble-based IDS for the protection of IoT nodes from attacks that exploit the lapses on application layer protocols, such as HTTP, DNS, and MQTT. Such protocols could facilitate the manipulation of the backend systems that IoT nodes are connected to, thereby providing room for launching of different forms of attacks, such leaking data, partial or complete system hijacking, and disabling system services. The proposed approach relies on the protocols characteristics to derive the statistical features via aggregation of the network traffic and building an AdaBoost classifier using ANN, DT, and NB algorithms. The proposed model was evaluated on two datasets, namely UNSW-NB15 and NIMS. The aggregation negatively affected the capability of the derived features to capture the intrinsic attack characteristics, thereby affecting the accuracy level of the model.

Being that IoT networks are distributed, the detection of external attacks is not important when dealing with internal attacks. This provides the invader an opportunity to evade the detection system by avoiding access through the network entry point; hence, cannot be detected by the detection system. Furthermore, internal attacks are usually launched by legitimate but compromised network nodes; hence, they are more difficult to detect.

2.5 Detection Of Internal Attacks

A distributed internal IDS for the detection of internally launched attacks from one or more local nodes in IoT was proposed by Thanigaivelan et al. (2016). The proposed system provides that each node monitors its surrounding nodes for deviations via monitoring of the packet size and data rate of the one-hop neighboring nodes. The study opted for information protection from manipulation via monitoring the features of the one-hop neighboring nodes rather than relying on cryptography; this enables identification of deviations in the network pattern. These observed characteristics helps in assigning weights to each surrounding node and nodes that did not reach a specific weight level are classified as anomalous. This approach makes the proposed technique capable of addressing the issue of internal attacks detection that the cryptography-based solutions cannot detect. However, the determination of abnormalities based on weights as a threshold is an ineffective approach as most of the misbehaving intruders and evasive attacks always strive to remain below the threshold level within the limits of the normal profile. Hence, the detection of compromised nodes that exhibits no suspicious features but still propagates false information may not be possible with this approach. Furthermore, the calculated weights by the proposed approach are static and may not be applicable to non-stationary scenarios with continuously changing nodes behavior.

Ghaleb et al. (2018) examined the results of the Destination Advertisement Object (DAO) control messages exploitation by a compromised node that sends false information to its parents which prompts flooding of the network with DAO messages and subsequent degradation of the IoT network performance. Such attack was prevented by proposing SecRPL for the restriction of the number of DAOs forwarded by a parent such that only a limited number of messages can be sent by each parent to a specific child node. The problem of this approach is that it can be easily fooled by attacks that flood a single child node by simultaneously exploiting numerous parents.

CA-DC-MDS was proposed by Ghaleb et al. (2019) as a multidimensional context-based anomaly detection system for deployment in smart vehicles. The proposed approach protects messages from internal attacks by relying as context references for capturing the non-stationary in these networks by using spatial-temporal correlations

of the consistency of cooperative awareness messages. Particle and Kalman filters were used to determine the spatial and temporal contextual thresholds. A major problem of this approach is that all the nodes are assumed to exhibit similar normal profile boundaries which are not true owing to the variation in the attitude and discretion of the situation based on the driver. As such, an anomalous situation to one vehicle could be considered normal for another vehicle.

A context-based IDS was developed by Pan et al. (2019) using the RIPPER algorithm and Bayesian Network for the protection of building control and automation networks. The system utilizes the heterogeneous data gathered from different network components to solve the issue of vulnerability of such systems by exploiting the increasing level of interaction between most of the weakly secured components. Hence, attackers may exploit these nodes to launch different forms of internal attacks. The model proposed in this study comprised of five different phases which as feature selection, context modeling, behavior analysis, threat assessment, & actions management. The study also proposed a novel Protocol Context-Aware data structure for context modeling which will aid building of the contextual array information that will serve as a segment of the training data. This will make the system capable of monitoring the whole aspects of a heterogeneous network for suspicious activities. However, the problem of the approach is the assumption of a stationary nature of the contextual information which is not applicable to heterogeneous scenarios like IoT where the misbehavior of one component affects the performance of several other components, thereby rendering such operating scenarios highly dynamic.

Li et al. (2019) presented a block chain-based system for internal attacks detection in IoT systems. The proposed system addressed the issue of vulnerabilities suffered by collaborative IoT networks by combining different weakly secured nodes that may be exploited by attackers to launch attacks against the neighboring nodes. The proposed model is a combination of block chains and distributed signature-based IDS in an IoT platform for the generation and verification of attack signatures. The block chain component aids in securely sharing of signatures among the IoT nodes. The study also assumed that for each node, the public/private key pair is preserved. This may be true, but it is not important because the attacker can decide to use the genuine

pairs without having to alter the existing signatures since the node is already under the control of the attacker. Despite the effectiveness of this approach in detecting spoofing and resource exhaustion attacks, it cannot detect the malfunctioning nodes propagated falsified information with their surrounding nodes. Attackers can exploit this misbehavior to manipulate the security profiles and thresholds of the network. Furthermore, such solutions are prone to concept drift due to the assumption that both normal and attack are stationary, which is not the case in dynamic IoT environments. A summary of the related studies is presented in Table 2.2 and Table 2.3.

2.6 Issues Of The Existing External And Internal Attack Detection Systems In IoT

External attacks detection, as can be seen, is easier because it relies on the strategy employed; the IDS is positioned at the network entry point to ensure efficient monitoring of the incoming traffic; meaning that the attack data is scrutinized at the network perimeter by the IDS. Contrarily, internal attack detection is a tedious task because of the participation if the attackers in the network activity and the generated attack data are not scrutinized by the IDSs. Despite the capability of host-based IDS to address this issue, they normally assume that the normal profile of the nodes is reliable, meaning that the existing host-based IDS are built on the premise of the legitimacy of the nodes in the internal network, but this is not true as attackers may hijack and use the legitimate nodes to launch attack on the normal node. Furthermore, the existing IDS in IoT assume the reliability of the data shared between nodes, but this does not hold as such nodes can be compromised and manipulated to share fake information with the other nodes. Sharing such fake information will make the nodes that receive such information to react to false events that could alter their security protocols and make them vulnerable to attacks. Hence, reliance on such compromised data to build detection models means the accuracy of the developed model will be low. Despite the effort to address these issues, some of the previous studies presented models that were built on the assumption a summary of the related studies of IDs bases IoT shown in Table 2.2 for outsider and Table 2.3 for insider bases of IoT that only few IoT nodes can be compromised by attackers,

but this is not the case. These days, attackers exploit the advanced attack strategies, such as botnets, to hijack numerous.

Table 2.2. A summary of the related studies of IDs for outsider bases IoT.

Author(s)	Research Problem	Solution	Limitation(s)	Detection Type
Yang et al. (2018)	The lack of sufficient ground truth (labeled) data necessary for machine learning classifiers to work properly.	Utilizing the active learning into model's training process by incorporating the concept of the human in the loop to enrich the training data and compensate the lack of labeled data in the training set.	- Involving human in labeling process is labor-intensive and error-prone.	***Outsider***
Raza et al. (2013)	Existing solutions do not provide end-to-end security for the messages travel through the network.	SVLETE incorporates IPSec and DTLS into RPL in 6LoWPAN networks to secure the data within the network.	- It doesn't safeguard against data tampering within the device itself (If a valid node is hacked, it does not prevent the infected device from sending misleading information.)	***Outsider***
Zeng et al. (2019)	Redundant observations in the traffic flows adversely affect the ability of detection solutions to detect attacks against IoT networks.	Statistical flow-based aggregation that summarizes the traffic flow and removes the redundant observations generated from flooding attacks such as DDoS.	-There is a dependence on workers who are active only on a single device, who can simultaneously produce context information produced on other nodes. This coexistence of data could be beneficial for detecting attacks by complex robots that work cooperatively and covertly to alter existing profile boundaries.	***Outsider***
Zhou et al. (2021)	The heterogeneity of IoT nodes makes it difficult to create a unified IDS detection solution.	An automata-based method that addresses the heterogeneity issue using an Labeled Transition System (LTS) extension helps build a consistent representation of all IoT system components	- When there are a greater variety of devices, the dimensionality and scarcity of data are also multiplied. As a result, it becomes increasingly difficult to leave a single, coherent legacy.	***Outsider***

Table 2.2. (Continued).

Author(s)	Research Problem	Solution	Limitation(s)	Detection Type
Lyu et al. (2017)	Existing IDSs in IoT suffer from the processing overhead due to limited resources in IoT nodes, which renders detecting the attacks at real time challenging.	The work employed the hyper ellipsoidal clustering at the fog layer to relay the processing overhead into the fog node and preserve the resources of IoT nodes.	- Any correlation between the IoT nodes that are connected through the fog node, which can be valuable for spreading information about emergent attacks taken out against one or more nodes, is ignored by this method.	***Outsider***
Akyildiz et al. (2006)	Misuse-based intrusion detection approach is not able to detect the novel, previously unseen attacks.	An anomaly-based detection model was trained with the normal behavior of the IoT nodes to build the normal profile such that, any deviation from the profile's boundaries is considered as attack.	- Presumes the operating environment is stationary, with small payload and stable packet losses limited to network traffic. The IoT networks are realistic in nature and dynamic. – To feel attacks with the required level of accuracy, the limited amount of information is not detailed. - The solution proposed suffers with the high fake alarm rate	***Outsider***
Meidan et al. (2018)	Existing Botnet detection solutions in IoT overlook the evasive nature of such botnets that change its behavior with each attack.	Deep Auto encoder was used to extract more robust and discriminative features that represent such evasive attacks.	-IoT behavior assumes to be stationary that does not maintain the IoT environment, given its dynamic nature. -It makes the unrealistic assumption that all features are present everywhere, regardless of network capacity, such as sensors and noise, both being faulty and in the system.	***Outsider***

Table 2.2. (Continued).

Author(s)	Research Problem	Solution	Limitation(s)	Detection Type
Breitenbacher et al. (2019)	Existing IDSs in IoT are prone to obfuscation strategies that sophisticated attackers employ to deceive the detection.	Whitelisting procedure that expounds on a set of legal applications allowed run on the IoT station was proposed.	- Due to the dynamic IoT environment, a rigid whitelist cannot function as a safety mechanism. - A whitelist cannot be comprehensive, as numerous legitimate and malicious programs may be omitted.	***Outsider***
Aljawarneh and Vangipuram (2020)	Although the choice of distance from centroid to the point of interest is imperative for IDS to judge an incoming observation as normal or abnormal, this challenge has been understudied and relatively less addressed in the research literature.	The proposed GARUDA is based on clustering feature patterns incrementally and then representing features in different transformation space through using a novel fuzzy Gaussian dissimilarity measure. The number of clusters made is made by a threshold that appears the allowable similarity of the features in the same group	- It makes the assumption that data from IoT devices is typically distributed, which is not the case in many cases, particularly those that operate in a highly dynamic environment.	***Outsider***
Pamukov et al. (2018)	Existing ISDs in IoT are unable to determine the zero-day attacks.	Negative Selection Algorithm (NSA) was employed to build two profiles, normal and malicious, using self/non-self-concept. The normal profile was then used to detect the zero-day attacks against IoT network.	- The definition of self/non-self is assumed to be stationary which could not be realistic since the behavior of IoT applications evolves over time as IoT environment dynamics.	***Outsider***

Table 2.2. (Continued).

Author(s)	Research Problem	Solution	Limitation(s)	***Detection Type***
Anthi et al. (2019)	Existing solutions focused on limited number of attacks and the data used for profiling those attacks are limited. They also have focused on detecting whether the packet is benign or malicious without knowing the type of the attack, which could be important for facilitating the countermeasure.	The 3-layer model corresponds to the 3 successive layers of the OSI Network model, i.e. In order to aggregate data about data units (frames, packets, and segments, respectively) from these three layers, it was suggested the data connection, network, and transport layer. Therefore, the dataset is enriched with various data.	- Frame, package, and part metadata is all that is extracted. Because of this, the model may not have been able to manage complicated attacks that can change (i.e. spoof) this data and mimic legal devices.	***Outsider***

The Table 2.3 is the summary of the related studies of Ids fon insider bases IoT for description type. The existing IDS in IoT assume the reliability of the data shared between nodes, but this does not hold as such nodes can be compromised and manipulated to share fake information with the other nodes.

Table 2.3. A summary of the related studies of IDs for insider bases IoT.

Author(s)	Research Problem	Solution	Limitation(s)	***Detection Type***
Thanigaivelan et al. (2016)	Cryptography based security cannot prevent the insider attackers that have the valid key or the necessary information to perform activities within the network.	One-hop neighbor node parameters like as packet size and data throughput are monitored by an internal anomaly detection system to seek for any network anomalies.	- Setting a weight threshold as an abnormality threshold is ineffective since many evasive attacks and invaders are attempting to stay inside the typical profile of the target. It's impossible to discover compromised nodes that aren't exhibiting any unusual behavior, yet are sending out fake information. - For non-stationary situations, weights estimated by the model given are static.	***Insider***

Table 2.3. (Continued).

Li et al. (2019)	Combining different types of weakly secured IoT nodes renders the network vulnerable to attacks.	A verifiable method of distributing signatures among IoT nodes is being developed by combining block chains with distributed signature-based IDSs.	- The attacker may use the parent pairs because he has already assumed control of the node.	***Insider***
Ghaleb et al. (2018)	Attackers can exploit Destination Advertisement Object (DAO) in An IoT node that has been compromised will broadcast misleading information to its parent nodes, resulting in a flood of DAO messages.	Because each parent can only transmit a limited amount of messages to each subnode using SecRPL, each parent can only redirect a certain number of DAOs at a time.	- It can be easily deceived by attacks that exploit different parent devices simultaneously to flood a particular sub-node.	***Insider***
Ghaleb et al. (2019)	Existing IDS solutions assume that the IoT context is stationary, which does not keep given the dynamic nature of IoT environment.	In order to reflect the non-stationary nature of these networks, dynamic thresholds were used to provide context references.	- If we assume that the majority of the nodes (vehicles) are secure, it is possible that adversaries can leverage the secure ones to perform similar attacks on other ones that are not secure, so that an adversary-based threat model may not be realistic.	***Insider***

3. METHODOLOGY

This chapter provides the developed methodology of the proposed Kalman and Cauchy (KC) clustering for anomaly detection based on the authentication of nodes within Internet of mobility networks (IoMTs) using extreme learning machines. In Sub-section 3.1., the problem situation and solution concept discussed and the simulation model and the problem definition were presented in Sub-section 3.2. While the system architecture will presents in Sub-section 3.3. The sensor fusion for trajectory estimation was presented in sub-section 3.4. Sub-section 3.5 presented the hampel filter in sub-section 3.6 Cauchy-based anomaly detection system was presented. The Sub-section 3.7 presented the intrusion detection system based on an extreme learning machine. Then in sub-section 3.8 the Unified Modeling Language (UML) diagram was described in details. Finally, in Sub-section 3.9 the evaluation metrics was listed and explained.

3.1 The Problem Situation and Solution Concept

Following the discussion in Chapter 1, this study is focused on addressing the problems associated with the existing host-based intrusion detection system (IDS) in terms of their inability to detect attacks internally launched against IoMT devices due to issues of the unreliability of the nodes in the IoMT network, as well as the lack of trust in the exchange of data between these nodes. Legitimate IoMT nodes can be manipulated by attackers in a bid to alter the integrity of the data it exchanges with the rest of the nodes in the system. The consequence of such manipulation is the outright corruption of the normal behavior of the nodes within the vicinity. Hence, there is no reliability in the data shared between these nodes and this affects the detection performance. This IoMT node compromise can also lead to the exploitation of the vulnerability of the compromised node to create and share unreliable data with other nodes to compromise the security features of such nodes, thereby increasing the vulnerability of the system.

Therefore, this study is aimed at providing the solution to the aforementioned issues that will usher in accurate and reliable IDS and anomaly detection solutions for IoMT systems. The proposed solution is conceptualized on the evaluation of the reliability of the IoMT nodes, as well as an assessment of the trustworthiness of the

shared data by these nodes. For the evaluation, the history of each node will be considered and the nodes will be adaptively re-evaluated from both historical and contextual perspectives. This ensures the detection of nodes that behave abnormally and the influence of the data shared by such nodes will be minimized. Furthermore, the impact of such data on the security status and profile of the model will be reduced, thereby keeping the model strong from manipulators that aim to compromise the security parameters and corrupt the knowledge-base of the models. A summary of the problems and the proposed solution concepts is presented in Table 3.1.

Table 3.1. Summary of problem situation and solution concept.

Problem Description	Solution Concept
The difficulty of assessing the reliability and trust levels of the data shared by the compromised nodes in IoMT systems.	Create a framework for intrusion detection using mobility anomaly detection and trained model for attacks
For estimating the trajectory of pedestrians within an indoor environment based on fusing WiFi with IMU data.	Develop a Kalman filter-based model
For detecting anomaly behaviour in IoMT based on the estimated trajectory by Kalman filter.	A Cauchy-based clustering method
For the training phase of an extreme learning machine to improve the detection accuracy.	An extreme learning machine-based IDS model for IoMT systems using the Cauchy clustering scheme.
Evaluate the proposed algorithms	Using classification and networking metrics

3.2 Simulation Model and Problem Definition

Assuming that we have an IoMT network consists of mobile wireless nodes with wireless fidelity (WiFi) accessibility. The WiFi combines one of the sensors that exists at each IoMT device for enabling the estimating its location prediction. The WiFi accessibility is achieved by set of access points (APs) installed in the environment at given points. In addition, the device contains inertial measurement unit IMU (accelerometer and gyro). The role of the accelerometer is to measure the acceleration and the role of the gyro is to measure the angular rate. The various sub-blocks or assisting blocks of the system: APs, map, pedestrian generation model,

pedestrian mobility model, traffic generation model and performance logging. The goal is to exploit the mobility information within IoMT for identifying nodes with abnormal mobility patterns which might be associated to threats or intruders that risk the network security. The simulation is depicted in Figure 3.1. We present an overview of the individual sub-blocks as follows.

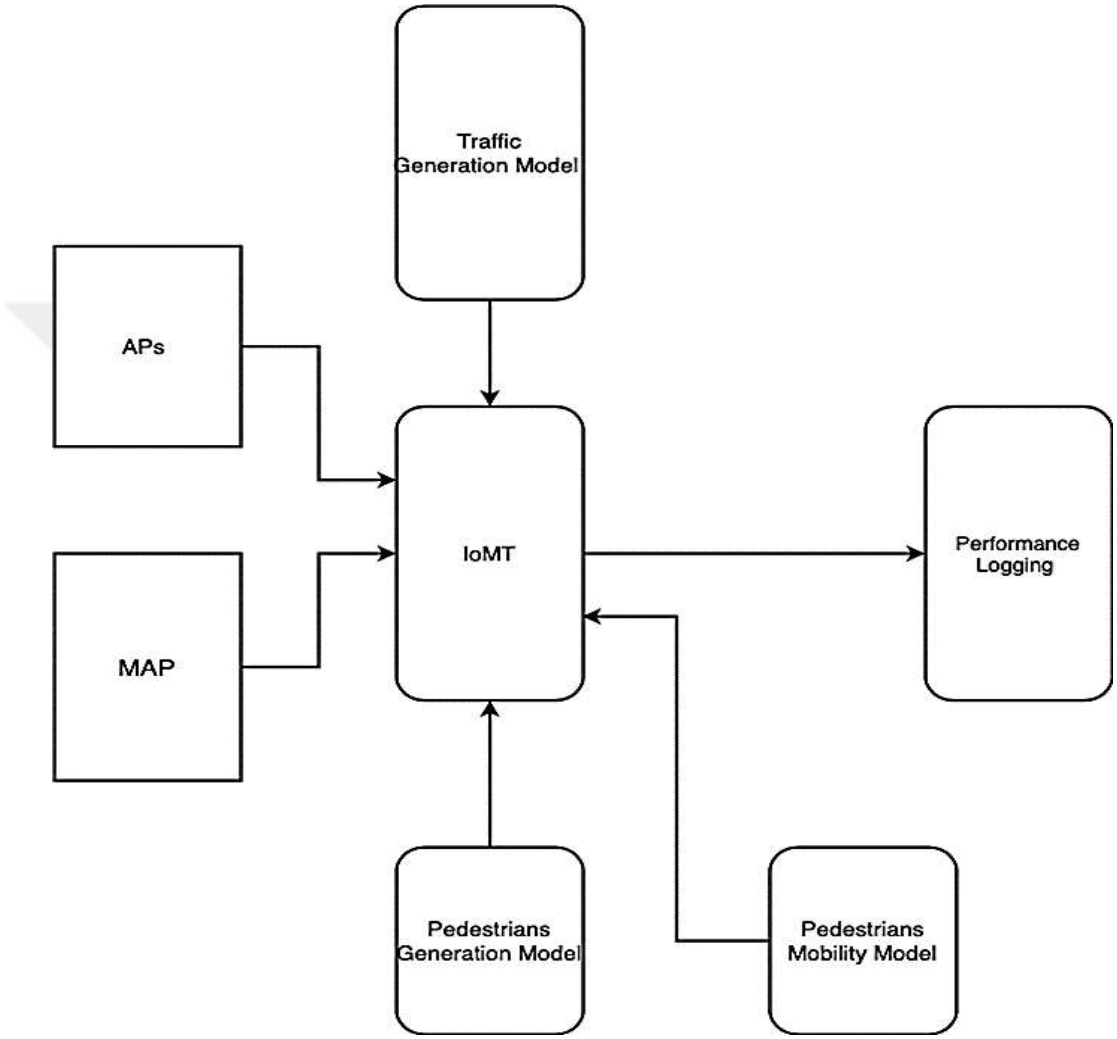

Figure 3.1. The simulation model for the IoMT.

3.2.1 Map

The map represents the indoor environment of 3D architecture. We consider that the environment consists of various rooms and multiple floors. In addition, we assume that there is a corridor or multiple corridors that connects between the rooms. The environment has fixed architecture for the operation of the model. However, the dynamical aspect of the environment is considered through the APs and the pedestrian's density within the environment. For the former, the characteristics of the

signal might change with time and for the latter the density is always subject to change.

3.2.2 Access points

The role of access points is to provide WiFi accessibility to the pedestrians within the environment. As it has been stated in the problem formulation, we have k APs distributed in the environment, and each one has coverage radius R which enables pedestrian's accessibility from their WiFi sensors to more than three APs at one time in order to enable location identification based on the fingerprint. The fingerprint of each location is stored in the environment based on an offline site survey. Hence, we assume that we have a dataset that represents the fingerprint throughout the environment.

3.2.3 Pedestrians generation model

The goal of the pedestrian generation model is to generate new pedestrians to enter the environment from an input point to the building (entrance). This model follows normal distribution with an expected value μ and standard deviation σ for the number of new arrived pedestrians and it follows exponential distribution for the time interval between one batch of pedestrians and other with an expected arriving rate λ. For the exit from the building, we have one point as exit which enables leaving the building in a probabilistic model with probability of 90% when the pedestrian passes by the point.

3.2.4 Traffic generation model

At each floor, the pedestrians combine sub-network that enables them exchanging various messages (considering that they are required to perform certain mission inside the building). The network is IoMT that needs to accomplish higher quality of service (QoS) and to be secured from attacks. The messages in the network are generated based on normal distribution with parameters μ_n, σ_n and with random time interval that follows exponential distribution with parameter of λ_n.

3.2.5 IoMT Sensing

The IoMT network is combined of set of nodes where each node represents an IoT device carried by the pedestrian while walking in the environment. The device has two important sensors: WiFi and inertial measurement unit (IMU). The role of the WiFi is to measure the WiFi fingerprint from the accessible APs which constructs a vector of Received Signal Strength Index (RSSI) signals as $RSSI = (rssi_1\ rssi_2 \ldots rssi_k)$ and the role of IMU is to measure the acceleration (a_x, a_y, a_z) and angular rate(w_x, w_y, w_z).

Where

a_x Denotes the acceleration on x axis

a_y Denotes the acceleration on y axis

a_z Denotes the acceleration on z axis

w_x Denotes the angular rate with respect to x axis

w_y Denotes the angular rate with respect to y axis

w_z Denotes the angular rate with respect to z axis

3.2.6 Performance logging

Performance Logging is an evaluation block that enables capturing the performance of the network and saves it time series for comparison. It uses the traditional network performance criteria, namely, Packet Delay Ratio (PDR), Received Packets, and End to End delay (E2E). The pseudocode is presented Algorithm 1. The inputs of the algorithm are list of the devices of the IoT network, the available packets in receiving buffer, the dropped packets by the nodes, and the current time. The outputs are the dropped packets, received packets, and E2E delay. In the beginning, the algorithm checks the available packets whether their lifetime is higher than the current time or not. In the case of lifetime is higher than the current time, this means that the packets are not yet expired so they are counted as received packets and the number of received packets is increased by one at the sink. Furthermore, the difference between the current time and the packet generation time is calculated as E2E delay. Otherwise, the packets are considered as dropped packets and the number of dropped packet is increased by one.

3.2.7 Mobility model

The pedestrian is assumed to move from in the corridor of the building in straight line trajectory with fixed speed and to enter a room from the available rooms based on Bernoulli distribution.

Algorithm 1 Sink Buffers processing.

Input:
(1)Devices: all devices that connected to the network.
(2) avilablePackets:array of the available packets in all devices.
(3)droppedPackets
(4)recivedPackets
(5)time
Output:
(1)droppedPackets
(2)recivedPackets
(3)E2EDelay
start algorithm

```
1: if any(avilablePackets) = 0 then
2:    sencdingDevice ⟵ selctRandom(avilablePackets):
3:    packet ⟵ extractP acket(Devices(sendingDevice)):
4:    if packet.lifeTime >0 then
5:       recivedPackets++
6:       E2EDelay = time-packet.generationTime.
8:    else
9:       droppedPackets++
10:  end if
11: end if
12: End algorithm
```

3.3 System Architecture

The role of the Kalman filter is to predict the location of the pedestrian with respect to time and to provide it to the anomaly detection algorithm (Li et al. 2019). The anomaly detection algorithm is used to detect any anomaly based on comparing the location of the subject node with the location of the nearby nodes and to provide an

anomaly index to the IDS which are used to filter out the malicious messages from the raw received packets buffer. The result of IDS is the legitimate received packets buffer which contains the legitimate messages after filtering out the malicious messages. In addition, the output of the IDS is connected to the performance logging block which is responsible of providing the performance metrics. The suggested framework of integrated Anomaly and Intrusion Detection for IoMT shown in Figure 3.2.

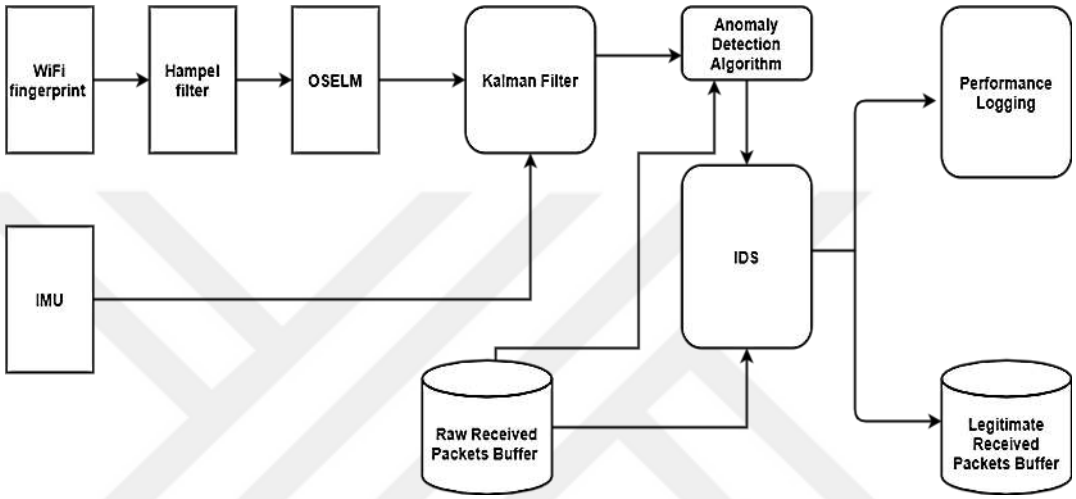

Figure 3.2. Framework of integrated anomaly and intrusion detection for IoMT.

3.4 Sensor Fusion for Trajectory Estimation

In order to estimate the location of the pedestrian, there are two sensors used by Kalman filter: the first is the WiFi data which does not give direct estimation of the location but it is used in a machine learning model for this purpose. The second one is the accelerometer data which is used inside the process model as an input vector.

Assuming that the fingerprint is $D = \{X_t, Y_t\}$ where X_t is a matrix with dimension of $n_l \times k$, n_l denotes the number of locations in the grid decomposition of the environment during the site survey, k denotes the number of APs

Y_t is a column with size of n_l We build a machine learning model based similar to the work of(Jiang et al., 2016) . We name this model as W and we used it for predicting the location based on the equation $z_{i,t} = f(W_{i,t}, x_{i,t})$ where $z_{i,t}$ denotes the predicted location of the pedestrian in the place$(x_{i,t}\ y_{i,t})^T$. W_i Represents the model built for the user i and $x_{i,t}$ denotes the sample obtained from the pedestrian i

at moment$t$. We use $\tilde{y}_{i,t}$ as part of the measurement vector directly inside Kalman. Hence, the measurement data of Kalman divided between z_t and $u_t = [a_{x,t}, a_{y,t}]$. The process model is given by the Equation (3.1).

$$\hat{X}_{t+1}^{-} = A\hat{X}_t + Bu_{t-1} \tag{3.1}$$

The state vector is defined by $X_{i,t} = (x_{i,t}\ y_{i,t}\ v_{x,i,t}\ v_{y,i,t})^T$. The set of Equations (3. 2-9) are applied.

$$x_{i,t+1} = x_{i,t} + Tv_{x,i,t} \tag{3.2}$$

$$v_{x,i,t} = v_{x,i,t-1} + Ta_{x,t-1} \tag{3.3}$$

$$x_{i,t+1} = x_{i,t} + T(v_{x,i,t-1} + Ta_{x,t-1}) \tag{3.4}$$

$$x_{i,t+1} = x_{i,t} + Tv_{x,i,t-1} + T^2a_{x,t-1} \tag{3.5}$$

$$y_{i,t+1} = y_{i,t} + Tv_{y,i,t} \tag{3.6}$$

$$v_{y,i,t} = v_{y,i,t-1} + Ta_{y,t} \tag{3.7}$$

$$y_{i,t+1} = y_{i,t} + T(v_{y,i,t-1} + Ta_{y,t-1}) \tag{3.8}$$

$$y_{i,t+1} = y_{i,t} + Tv_{y,i,t-1} + T^2a_{y,t-1} \tag{3.9}$$

Hence, we write the two matrices *A* and *B*

$$A = \begin{pmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.10}$$

$$B = \begin{pmatrix} T^2 & 0 \\ 0 & T^2 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{3.11}$$

The measurement model is given by Equation 3.12.

$$z_t = H\hat{X}_t^{-} \tag{3.12}$$

The measurement vector is defined based on the update from the WiFi that is calculated by Equation 3.13.

$$z_{i,t} = f(W_{i,t}, x_{i,t}) \tag{3.13}$$

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{3.14}$$

Hence, the predicted covariance matrix P_{t+1}^{-} is given based on the Equation (3.15)

$$P_{t+1}^{-} = AP_tA^T + Q \tag{3.15}$$

Kalman filter gain is given by Equation (3.16)

$$K_t = P_t^{-}H_t^T(H_tP_t^{-}H_t^T + R)^{-1} \tag{3.16}$$

Where

Q Denotes the process model covariance matrix

R Denotes the measurement model covariance matrix

Hence, after receiving the measurement, we correct both the state and the covariance matrix using the two Equations (3.17-18)

$$\hat{X}_t = \hat{X}_t^{-} + K\left(z_t - H\hat{X}_t^{-}\right) \tag{3.17}$$

$$P_t = (I - K_tH)P_t^{-} \tag{3.18}$$

For calculating $z_{i,t} = f(W_{i,t}, x_{i,t})$, we adopt feature adaptive online sequential extreme learning machine FA-OSELM given in.

3.5 Hampel Filtering

In order to improve the estimation of Kalman filter, we add Hampel filter at the output of the WiFi sensor. The role of Hampel filter is to remove outliers (Yao et al., 2019). It operates based on sliding window of the time series generated from the sensor at each APs RSSI reading. For each position of the sliding window, it calculated the median and standard deviation and replaces the samples that have value than three times of the standard deviation with the median. Such process considers that values that exceed the median deviation with three times of the standard deviation as outlier.

The usage of Kalman filter and Hampel filter for localization is presented in Algorithm 2. The input of the algorithm are: RSSIBuffer which represents the data for the last RSSI measurements collected inside a window *w* which represents the size of hample window, deviceGeometry which represents the geometrical data of device, namely, position, velocity and acceleration, and Online Sequential Extreme Learning Machine OSELM which represents a pre-trained model on the WiFi fingerprint (Zou et al., 2015). The output of the algorithm is currentPos which represents the current estimated position of the device. The operation of the algorithm consists of three steps:

1- Filtering out the outliers using hample filter.
2- Calculating the current location using OSELM and filtered RSSI.
3- Using the current position as input to Kalman filter measurement vector to fuse between RSSI and IMU.

Algorithm 2 Device Localization.

Input:

(1)RSSIBuffer: the RSSI data for the last w RSSI measurements where w is the size of hample filter window

(2)deviceGeometry: the geometrical data of device (Position, velocity, acceleration)

(3)OSELM: the pre-trained OSELM module

Output:

currentPos: the current estimated position of the device.

start algorithm

1: FRSSI ⟵ hample(RSSIBuffer)

2: predictedPos ⟵ eval(OSELM; FRSSI(:; end)) // estimate for the last RSSI

3: currentPos ⟵ kalmanFilter(predictedPos; deviceGeometry)

4: **end algorithm**

3.6 Cauchy Based Anomaly Detection

To propose a weighted trustworthiness assessment for detecting anomaly behaviour in IoMT based the estimated trajectory by Kalman filter and information gain calculated from the network features.

At every time moment, a new point is generated from Kalman filter are combined with the network features data in one vector and projected to clustering space as one point. The point is compared with the existing clusters and it is added to the cluster that combines with the point maximum Cauchy density considering that the density does not exceed certain threshold. Otherwise, the point is projected to an outlier. The outlier is saved until collecting minimum number of points in the outlier that enables converting the outlier to a cluster. The pseudocode used for this is given in Algorithm 3. The input of the algorithms are: C which represents the Cauchy clusters set, $pedestriansPos$ which represents the pedestrians position, $gammaC$ which denotes density threshold of clustering, $Anomaly$ which represents the detected anomaly, interval which denotes pruning interval time, threshold which denotes the anomaly threshold and the time. The outputs of the algorithm are C which denotes the updated Cauchy clusters set and $Anomaly$ also after being updated. The process of the algorithm starts by checking the position and it uses it for creating a new cluster in the case of the first sample or when the maximum density of the position with respect to other clusters is less than a threshold $gammaC$, otherwise it assigns it to the cluster associated with the maximum density and it calls for this the function $UpdateCuster$. The next step of the algorithm is label the anomaly samples based on the criteria of having number of members within their clusters less than the input threshold. Lastly, the packets that are associated with the anomaly samples are deleted.

Algorithm 3 Cauchy with anomaly detection.

Input:

(1)C: the cauchy clusters set.

(2)pedestriansPos:the pedestrians positions.

(3)gammaC: the density threshold of clustering.

(4)Anomaly: the detected anomaly

(5)interval: the pruning interval time

(6)threshold: the anomaly threshold

(7)Time

Output:

(1)C:updated cauchy clusters set.

(2)Anomaly:updated anomaly list.

start algorithm

```
1:for each P in pedestriansPos do
2:    if C.numberOfClusters == 0 then
3:       addNewCluster(p)
4:    else
5:       densities ← CalculateDensity(C; p)
6:       [maxGammaj, maxIndex]=Max(densities);
7:       if maxGammaj<gammaC then
8:          addNewCluster(p)
9:       else
10:         C = UpdateCluster(C(maxIndex),P)
11:      end if
12:   end if
13: end for
14: if mod(Time,interval)==0 then
15:    for each cluster ci in C do
16:       if ci.NumberOfMembers < threshold then
17:          Anomaly=[Anomaly ci.Members]
18:          delete(ci)
19:       end if
20:    end for
21: end if
22:End algorithm
```

The pseudocode of creating the new clusters is presented in Algorithm 4. The inputs of the algorithm are C which denotes the Cauchy clusters set and P which denotes the pedestrian position. The output of the algorithm is the updated Cauchy clusters set. The algorithm 5 creating new cluster from the input position, it uses its value for the cluster center and it increases the number of points inside the clusters by one. In addition, it initiates the value the parameters of the clusters.

Algorithm 4 addNewCluster.

Input:
(1)C: the cauchy clusters set.
(2)P:the pedestrian position.
Output:
(1)C:updated cauchy clusters set.
start algorithm
1: C.numberOfClusters++
2: C(C.numberOfClusters).numberOfMembers=1
3: C(C.numberOfClusters).Members = P
4: C(C.numberOfClusters).center = P
5: C.numberOfClusters).Sj = 0
6: C.numberOfClusters).segmaj = 0
7: **End algorithm**

Algorithm 5 CalculateDensity.

Input:
(1)C: the cauchy clusters set.
(2)P:the pedestrian position.
Output:
(1)C:updated cauchy clusters set.
(2)densities:the densities of clusters.
start algorithm
1: **for** each cluster ci in C do
2: d = distance(ci.center,P)
3: T = ((ci.numberOfMember-1)/(ci.numberOfMember))*trace(ci.segmaj)
4: r = T/(deltaS)
5: ci.gammaj = (1 + r)/(1 + (d/(deltaS)) + r)
6: **end for**
7: densities=[C.gammaj]
8: End algorithm

The pseudocode of the cluster head update is presented in Algorithm 6. It takes the input corresponding cluster set as input as well as the pedestrian position and it returns the updated cluster. It uses the position for updating the cluster center and other parameters as it is shown in the pseudocode.

Algorithm 6 UpdateCluster.

Input:

(1)ci: the cauchy cluste to be updated.

(2)P:the pedestrian position.

Output:

(1)C:updated cauchy clusters set.

start algorithm

1: ci.numberOfMembers++

2: ci.Members= [ci.Members P]

3: ej = P - ci.center

4: ci.center = ci.center + ((1/ci.numberOfMembers)*ej)

5: ci.Sj = ci.Sj + ej'*ej

6: ci.segmaj = (1/ci.numberOfMembers)*ci.Sj

7: **End algorithm**

3.7 Intrusion Detection System

The role of intrusion detection system is to operate with the anomaly detection that marks the data that arrives from nearby nodes as associated with anomaly or being normal. The prediction of anomaly that is carried by the anomaly detection which operates based on Cauchy clustering is further used with the network data that are obtained from the packets received buffer to give more confident decision about the non-legitimate data because of black-hole attack or distributed denial of service DDoS attack. This is used based on un-supervised learning approach carried by deep-believe neural network. Hence, the final decisions will one of two labels for every packet in the buffer: the first one is normal or legitimate packets and the second one is packets attached with attack. The IDS was integrated in the sink buffer processing algorithm 7.

Algorithm 7 Sink IDS Buffers processing.

Input:

(1)Devices: all devices that connected to the network.

(2)avilablePackets:array of the avilable packets in all devices.

(3)droppedPackets

(4)recivedPackets

(5)time

(6)ELMIDS: the trained ELM for detecting intrusions

Output:

(1)droppedPackets

(2)recivedPackets

(3)E2EDelay

start algorithm

1:**if** any(avilableP ackets) = 0 then

2: sendingDevice ⟵ selctRandom(avilableP ackets):

3: packet ⟵ extractP acket(Devices(sendingDevice)):

4: **if** packet.lifeTime > 0 ∧ ELMIDS(packet) == 0 then

5: recivedPackets++

6: E2EDelay = time-packet.generationTime.

7: else

8: droppedPackets ++

9: **end if**

10:**end if**

11:**End algorithm**

3.7.1 Intrusion Detection based on extreme learning machine OSELM

According to the work of (Singh et al. 2015) , we can built the model from the following steps :

1- We assume that we have a fingerprint dataset given as

$$\aleph = \{(\mathbf{x}_i, \mathbf{t}_i) \mid \mathbf{x}_i \in \mathbf{R}^k, \mathbf{t}_i \in \mathbb{N}, i = 1, \cdots, \mathrm{N}\}$$

2- We decompose the labeled fingerprint data into two parts: the first one is the boosting data which is given as $\aleph$ when $i = 0,1, \ldots N_B$ where N_B denotes the size of the boosting data, and the remaining is the training data.

3- We apply the boosting phase as given in the following steps :

3-1 We generate randomly weights for the input hidden layer neurons $\mathbf{w}_i$ and the hidden layers biases b_i.

3-2- We calculate the hidden output matrix using Equation (3.19).

$$\mathbf{H}_0 = [\mathbf{h}_1, \cdots, \mathbf{h}_{\widetilde{N}}]^T \tag{3.19}$$

Where

$\mathbf{h}_i = [g(\mathbf{w}_1 \cdot x_i + b_1), \cdots, g(\mathbf{w}_{\widetilde{N}} \cdot x_i + b_{\widetilde{N}})]^T$
$i = 1, \cdots, \widetilde{N}$

g Denotes the activation function

3-3- estimate the hidden output weights using Moore-Penrose equations (3.20-23).

$$\beta^{(0)} = \mathrm{M}_0 \mathrm{H}_0^T \mathrm{T}_0 \tag{3.20}$$

$$\beta^{(0)} = \mathrm{M}_0 \mathrm{H}_0^T \mathrm{T}_0 \tag{3.21}$$

$$\mathrm{M}_0 = (\mathrm{H}_0^T \mathrm{H}_0)^{-1} \tag{3.22}$$

$$\mathrm{H}_0{}^{\dagger} = (\mathrm{H}_0^T \mathrm{H}_0)^{-1} \mathrm{H}_0^T \tag{3.23}$$

3-4- For each new received chunk apply the recursive Equation (3.24).

$$\begin{gathered} \mathrm{M}_{j+1} = \mathrm{M}_j - \frac{\mathrm{M}_j \mathrm{h}_{j+1} \mathrm{h}_{j+1}^T \mathrm{M}_j}{1 + \mathrm{h}_{j+1}^T \mathrm{M}_j \mathrm{h}_{j+1}} \\ \beta^{(j+1)} = \beta^{(j)} + \mathrm{M}_{j+1} \mathrm{h}_{j+1} \left(\mathrm{t}_j^T - \mathrm{h}_{j+1}^T \beta^{(j)}\right) \\ j = j + 1 \end{gathered} \tag{3.24}$$

3-5- In case the number of features has changed, apply transfer learning equations to preserve old weights when the neural network inputs are changed to the new number.

3.8 MUDI Stream: Multi-Density The Clustering Algorithm

In this study we prefer to use the MUDI Stream which is an Online-Offline algorithm, this algorithm used to evaluate and benchmark the work of detecting anomaly behavior and the IDs with the proposed algorithm the Kalman Cauchy clustering. The MUDI Stream is an Online-Offline algorithm with four major components. Algorithm 8 used during the online phase, it stores brief information about the evolving Multi-Density data stream in the form of core Mini-Clusters. While the algorithm 9 works as an offline phase uses an adapted Density-Based clustering algorithm to generate the final clusters. The Grid-Based method is used as an Outlier-Buffer to manage both noise and Multi-Density data to differentiate between anomalies, normal packets and nodes, new point is generated from Kalman filter are combined with the network features data in one vector and projected to clustering space as one point. The point is compared with the existing clusters and it is added to the cluster that combines with the point maximum Multi-density considering that the density does not exceed certain threshold. Otherwise, the point is projected to an outlier while also reducing clustering merging time. The algorithm is evaluated on various synthetic and Real-World datasets using different quality metrics and scalability performance is compared. The experimental results show that the algorithm presented in the work of (Amini et al. 2016; Amini et al. 2014) enhances clustering quality in Multi-Density environments and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) as shown in Figure 3.3 . For this study we used the definition 1 to define the Neighboring grid (N_g), Two density grid; to detect the normal and legitimate nodes. And the definition 2 to calculate the Outlier Wight Threshold (OWT) which used to know which points exceeds the threshold to considered as anomaly. The algorithm 8 is for MUDI stream online phase, and algorithm 9 is for Multi- density-based spatial clustering of applications with noise (M-DBSCAN) for offline phase.

Definition1: Neighboring grid (N_g), Two density grid:

$g_1 = (j_1^1 , j_2^1 , \ldots., j_d^1)$ and $g_2 = (j_1^2 , j_2^2 , \ldots., j_d^2)$ are neighbors if there exists $m , 1 \leq m \leq d , j_i^1 = j_i^2 , i = 1 , \ldots m-1 , m+1 , \ldots . d ; and \, |j_m^1 - j_m^2| = 1$. Then g_1 and g_2 are neighboring grids in the mth dimension.

Definition 2: Outlier Wight Threshold (OWT):

If the last updated time of grid g is t_p then at the current time t_c, Outlier Weight Threshold (OWT) is defined as follows ($t_c > t_p$), where c is the amount of the time:

$$OWT\ (t_p\ , t_c) = \frac{\propto}{N} \sum_{i=0}^{t_c - t_p} 2^{\lambda\ i}\ = \frac{\propto(\ 1-2^{-\lambda(t_c - t_p+1)}\ )}{N(1-\ 2^{-\lambda})} \tag{3.25}$$

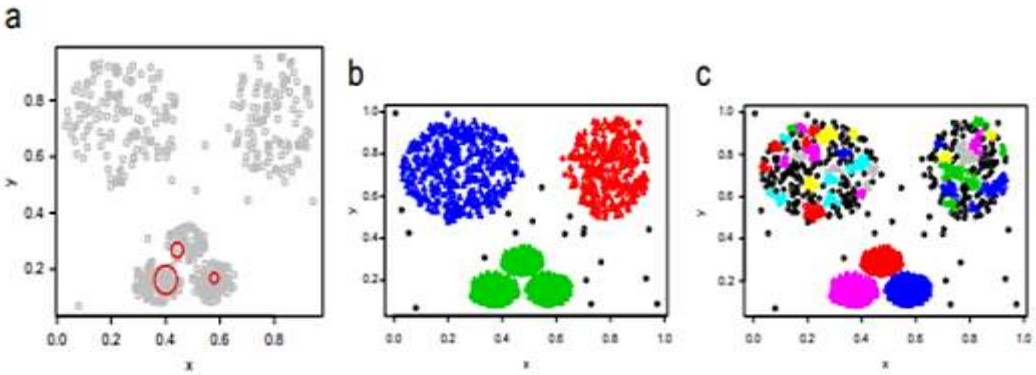

Figure 3.3. Clustering-multi-density data (a) denseStream , c = 0.05, (b) DBSCAN , c = 0.05 , (c) DBSCAN , c = 0.21 , MinPits = 5 (Amini et al., 2016).

Algorithm 8 MUDI-Stream online phase $(\alpha, \lambda, gridGranularity, N)$.

(1) **Input:** a data stream

(2) **Output:** core mini-clusters

1: $t_{pt=\left[\frac{1}{\lambda}\ log_2^{\alpha/(\alpha - N(1-2^{-\lambda}}\right]}$

2: $t_{c\leftarrow 0;}$ Initialize the grid structure $using\ gridGranularity$;

3: **while** not end of stream **do**

4: Read data point x from Data Stream;

5: $cmcs \leftarrow$ find the nearest cmc to x in cmc list;

6: **if** distance$(x\ , cms_s)\ _{\leq mcd_{cmc_s}}$ **then**

7: $cmc_s\ \leftarrow cmc_s\ + x$;

8: **else**

9: map the new data point x to the grid;

10: $n_{g\leftarrow} n_g + 1$;

11: $W_{g\leftarrow}\ 2^{-\lambda(t_c - t_p)})\ w_g\big(\ t_p\big) + 1$;

12: $t_{p\leftarrow} t_c$;

13: Update $GS\ n_g\ , t_p\ \ , w_g$;

14: **if** $n_g\ > 1$ and $w_g \geq \frac{\alpha}{N(1-2^{-\lambda})}$ **then**

15: $w_{cmc} \leftarrow w_g$;

16: $c_{cmc} \leftarrow \frac{\sum_i^n = \mathrm{f}(t_c - T_i)(P_i)}{w_{cmc}}$;

17: $r_{cmc} \leftarrow \frac{\sum_i^n = \mathrm{f}(t_c - T_i) distance(P_{ij}, C_{cmc})}{w_{cmc}}$;

18: **for** data points pi in the grid g **do**

19: $mcd_{cmc} \leftarrow$Maximum $distance\{(c_{cmc}, P_i)\}$;

20: **end for**

21: **end if**

22: **end if**

23: **if** $t_c \ mod \ t_{pt} == 0$ then

24: update the weight of all grids in grid list

25: $(w_g(t_c) = 2^{-\lambda(t_c - t_p)} * w_g(t_p))$

26: **for all** grid g **do**

27: $OWT(t_p, t_c) \leftarrow \frac{\alpha(1-2^{-\lambda(t_c - t_p + 1)})}{N(1-2^{-\lambda t_p})}$;

28: **if** $w_g < OWT$ **then**

remove grid g from grid list;

29: **end if**

30: **end for**

31: **for all** $\{cmc\}$ **do**

32: **if** $w_{cmc} < \frac{\alpha}{N(1-2^{-\lambda})}$ **then**

remove cmc from $\{cmc\}$;

33: **end if**

34: **end for**

35: **end if**

36: $t_c \leftarrow t_c + 1$;

39: **end while**

Algorithm 9 M-DBSCAN ($MinPts$) – MUDI-Stream's offline phase**.**

(1) Input: core mini-clusters

(2) Output: arbitrary shape clusters

1: mark all $cmcs$ as unvisited;

2: **repeat**

3: randomly choose an unvisited point cmc_p;

4: mark cmc_pas visited;

5: $\{N_g\ (cmc_p\ )\} \leftarrow cmc_p\ - grid - nieghborhood$;

6: **If** $|\{N_g\ (cmc_p\ )\}| \geq MinPts$ **then**

create a new cluster C, and add cmc_p to C;

7: $\{N_{core}\ \}$

$\leftarrow find\ MinPts - nearst\ nieghborhood\ cmc_p\ \{N_g\ (cmc_p\ )\}$ from cmc_p

8: Calculate $\mu(Dist_{core}\ )$, and $\sigma\ (Dist_{core}\ )$;

9: **For** each cmc_p in $\{N_{core}\ \}$ **do**

10: **If** cmc_q is unsupervised **then**

11: Mark cmc_q as visited ;

12: $\{N_g\ (cmc_p\ )\} \leftarrow cmc_q\ - grid - nieghborhood$;

13: **If** $|\{N_g\ (cmc_q)\}| \geq MinPts$ **then**

14: $\{N_{sh}\ (cmc_q)\}$ $\leftarrow$ find $MinPts$-nearest-nighbors in $\{N_g\ (cmc_q)\}$ from cmc_q;

15: Calculate $\mu(Dist_{cmc_q}\ )$,and $\sigma\ (Dist_{cmc_q}\ )$;

16: **If** $\mu(Dist_{cmc_q}\ ) \in [\ \mu(Dist_{core}\ ) - \sigma\ (Dist_{core}\ )\ ,\ \mu(Dist_{core}\ ) + \sigma\ (Dist_{core}\ )]$ **then**

17: $\{N_{core}\ \} \leftarrow \{N_{core}\ \} U\ \{N_{sh}\ (cmc_q)\}$;

18: Update $\mu(Dist_{core}\ )$, and $\sigma\ (Dist_{core}\ )$;

19: **End if**

20: **End if**

21: **End if**

22: **If** $cmc_p$is not assigned to any cluster **then**

23: **Add** $cmc_q$to cluster C ;

24: **End if**

25: **End for**

26: **Else**

27: Mark cmc_p as noise ;

28: **End if**

29: **Until** no cmc is unvisited ;

3.9 CEDAS: Fully Online Clustering Of Evolving Data Stream Into Arbitrary Shaped Clusters

In this study we used the CEDAS algorithm to detect the anomaly behavior and used with IDs to benchmark the results with the proposed algorithm to prove the best algorithm which capture the anomalies in online phase, we discussed this algorithm according to our proposal. It is a fully online algorithm for clustering an evolving data stream into arbitrary-shaped clusters. This algorithm helps to detect the abnormal and legitimate nodes in the floor , by using the following procedure ; It is a two-stage tactic that is accurate, noise-resistant, algorithmically and memory effective, and has a low time penalty as the number of data dimensions grows (Islam et al., 2019) . The first stage generates microclusters, and the second stage combines these microclusters to form macroclusters. That is means the algorithm help to calculate the E2E and PDR Dimensional stability and high speed are achieved by keeping the calculations simple and minimal by employing hyper spherical microclusters (Hyde et al., 2017) the work presents in Figure 3.4 (Dong et al., 2018).

The CEDAS approach defined as the following:

1. Cluster Graph: the structure that specifies which microclusters combine to form which macroclusters this is saved by noting the intersects of each micro-cluster in 'Edge,' as well as the appropriate macro-cluster assignment in 'Macro.'

2. Local density: the number of samples per micro-cluster

3. Macro-cluster: a cluster consisting of a number of intersecting microclusters.

4. Micro-cluster: a micro-cluster with a local density beneath the threshold.

5. Outlier-micro-cluster: a micro cluster with local density lower than the threshold.

6. Sample: any data point in “D” dimensions.

7. Threshold: the minimum number of samples within the micro-cluster radius of any sample to form a micro-cluster.

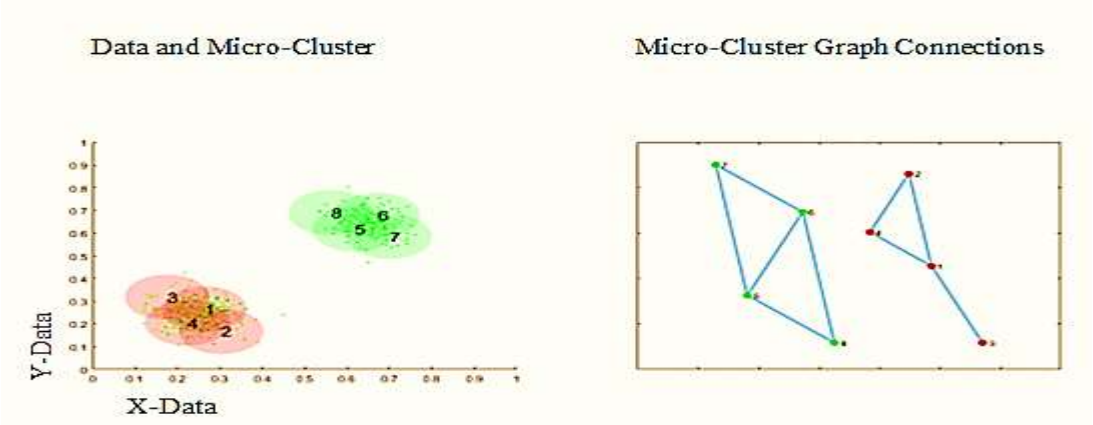

Figure 3.4. Example of the CEDAS algorithm micro-clusters and graph structure. The data together with two macro-clusters in red and green are shows the cluster graph structure with the nodes of the sub-graphs colored according to the macroclusters (Hyde et al., 2017).

CEDAS, in general, is a data-driven approach to divide the data space into shell regions and kernel based on a user defined radius, r_0. Each micro-cluster made up of a shell annulus region between radius $\frac{r_0}{2}$, r_0 and a kernel region being $r \leq \frac{r_0}{2}$. Any microcluster that exceeds a certain density threshold is considered for membership in the macrocluster. Macroclusters are formed when kernel regions of one microcluster intersect the shell region of another microcluster. Micro-clusters that are larger than the threshold but do not intersect; are also considered as macro-clusters. Shell regions are thought to be the edges of macroclusters. New data from the data stream will be classified into one of three categories:

1. Empty space, where it will form a new, outlier-micro-cluster represents in algorithm 10.

2. A micro-cluster shell region, where it will be assigned to the cluster, the cluster count updated and the micro-cluster center recursively updated to the mean of its samples represents in algorithms 11 and 12.

3. A micro-cluster kernel region, where it will be assigned to the micro-cluster and the cluster count updated as shown in algorithm 13.

Algorithm 10 CEDAS: Initialization.

Input: *x*, r_0

1: Create micro-cluster structure containing:
2: *C1(Centre) = x*
3: *C1(Count) = 1*
4: *C1(Macro) = 1*
5: *C1(Energy) = 1*
6: *C1(Edge) = 1*
7: Set number of micro-clusters to 1
8: Set modified micro-cluster number, for use updating the graph structure.

Algorithm 11 CEDAS: Update Micro-Cluster.

Input: *x*, *C*, r_0

1: find distance to nearest micro-cluster centre, d_{min}
2: **if** $d_{min} < r_0$ **then**
3: reset micro-cluster *Energy* to 1
4: increment number of samples contained in micro-cluster
5: **if** *data is within micro-cluster shell* **then**
6: recursively update micro-cluster centre
7: **end**
8: **else**
9: Create new micro-cluster
10: **end**

Algorithm 12 CEDAS; Kill Micro-Cluster.

Input: *C*, *Decay*

1: Reduce all *C(Energy)* by *Decay*
2: **if** *Any C(Energy) < 0* **then**
3: Remove micro-cluster
4: Remove all edges containing the micro-cluster
5: Decrement the number of micro-clusters
6: **End**

Algorithm 13 CEDAS: Update Graph.

1: **if** A *micro-cluster has been modified* **then**
2: **if** *the micro-cluster edge list has changed* **then**
3: Set a new macro-cluster number throughout the graph
4: **end**
5: **end**
6: **if** *Any micro-clusters have died* **then**
7: Set new macro-numbers for the sub-graphs of its previous edges
8: **End**

3.10 Deep Belief Networks

The other benchmark algorithm is a Deep Belief Networks (DBNs) it is one of the algorithms of convolution neural networks (CNN) which depends on the roles of neural networks. They are probabilistic generative models with multiple layers that can learn to extract a deep hierarchical representation of training data. We used DBNs for detecting the anomaly behavior by improves the performance of this algorithm after combined it with kalman filter and cauchy algorithm to improve the results of proposed model. DBNs are made up of several layers of restricted Boltzmann machines (RBMs) with a classifier on top. DBNs can be trained quickly by training multiple RBMs with the greedy layer-wise unsupervised training strategy (Hinton et al., 2006). Following network pre-training, the network parameters are fine-tuned using to achieve better classification results. RBMs are an energy model in which the visible–hidden layers are fully connected but the visible–visible and hidden–hidden layers are not (Ghojogh et al., 2021).

RBMs have found widespread application in classification, dimension reduction, feature extraction, topic modeling, and collaborative filtering. RBM maps the sample using visible and Bernoulli random-valued hidden units, and converts the sample data from a dimension m input space to a dimension m feature space n, where $n < m$, as shown in Figure 3.5. An RBM is an energy-based generation model that contains a layer of visible nodes ($v_1, v_2, \cdots, v_i, \cdots, v_m$) showing the data and a layer of hidden nodes ($h_1, h_2,, \cdots, h_j, , \cdots, h_n,$) learning to represent features, for every $v_i \in \{0, 1\}$ and $h_i \in \{0, 1\}$. We define that the biases of the visible nodes are ($b_1, b_2, \cdots, b_i, \cdots, b_m$), and the biases of the hidden nodes are ($c_1, c_2, \cdots, c_j, \cdot$

$\cdot\cdot$, c_n) (Ghojogh et al., 2021). The energy function $E(v, h)$ of the joint configuration $\{v, h\}$ is defined as follows:

$$E(v,h) = -\sum_{i=1}^{m} b_i\ v_i - \sum_{j=1}^{n} c_j\ h_j - \sum\sum_{i=1}^{m} w_{ij}\ v_i h_j \qquad .26)$$

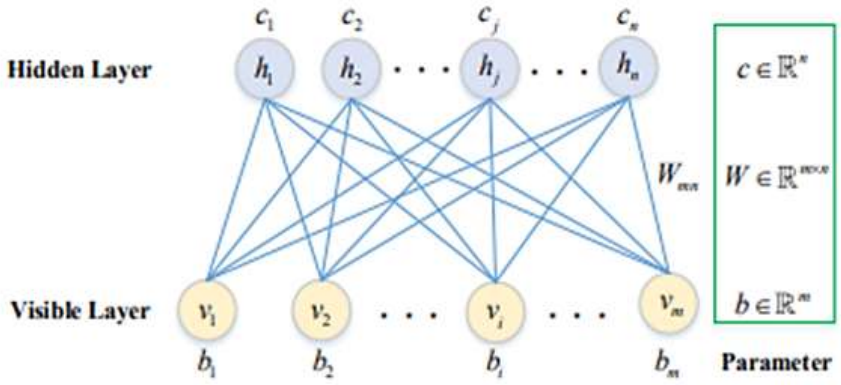

Figure 3.5. The graphical representation of an RBM with m visible nodes and n hidden nodes (Yang et al., 2019).

According to the energy function $E(v, h)$, the joint probability distribution for hidden and visible vectors can be defined as follows:

$$P(v,h) = \frac{1}{z}\ exp(-E(v,h)) \qquad (3.27)$$

Where Z is the partition function which is the sum over all possible pairs of visible and hidden vectors.

$$Z = \sum_v \sum_h \exp(-E(v,h)). \qquad (3.28)$$

The probability assigned to a visible vector v is given by summing over all possible binary hidden vectors h, as follows:

$$P(v) = \sum_h P(v,h) = \frac{1}{z}\ exp\big(-E(v,h)\big). \qquad (3.29)$$

Because the same layers in an RBM do not have direct connections, the hidden units are independent of the visible units, and vice versa and depends of the concepts of the Machine Learning (ML). As a result, the conditional probability of the hidden vector h given the visible vector v is:

$$P(h|v) = \prod_{j=1}^{n} p(h_i\,|v)\ . \qquad (3.30)$$

Similarly, given the hidden vector h, the conditional probability of the visible vector v is given by:

$$P(v|h) = \prod_{i=1}^{m} p(v_i|h). \quad (3.31)$$

The activation state of each hidden unit is conditionally independent because of a given a visible vector$v$. At this point; $hj \in \{0,1\}$ and the activation probability of the jth hidden unit is described as follows:

$$P(h_j = 1|v) = Sigm(\sum_{i=1}^{m} w_{ij}\ v_i + c_j) \quad (3.32)$$

Where $Sigm(x) = \frac{1}{1+e^{-x}}$ is the logistic sigmoid function.

As a result, when given a hidden vector$h$, the activation probability of each visible unit is conditionally independent:

$$P(v_i = 1|h) = Sigm(\sum_{j=1}^{n} w_{ij}\ h_j + b_i) \quad (3.33)$$

An RBM is trained to decrease the energy in Equation (3.27) by finding the values of the network parameters $\theta = (W, b, c)$; the energy of the network is decreased if the RBM has been trained. And the probability in Equation (3.34) is increased. To increase the log-likelihood of $P(v)$, its gradient with respect to the network parameters θ can be calculated as follows (Yang et al. 2019):

$$\frac{\partial \log P(v)}{\partial \theta} = \mathbb{E}_{P(h|v)}\left[\frac{\partial \mathbb{E}(v,h)}{\partial \theta}\right] + \mathbb{E}_{p(v',h')}\left[\frac{\partial \mathbb{E}(v',h')}{\partial \theta}\right] \quad (3.34)$$

The expectation operator is denoted by the operator$\mathbb{E}$. The expectation on the left hand side of Equation (3.34) can be calculated precisely, while the expectation on the right hand side is difficult to calculate. The Contrastive Divergence (CD) approach for estimating log-likelihood gradient was devised to overcome the difficulties of computing expectations. To update the network parameters, the CD-k algorithm approximates the expectation by restricted k ($typically\ k = 1$) iterations of Gibbs sampling.$\theta = (W, b, c)$. The Persistent Contrastive Divergence (PCD) algorithm is an upgraded version of the CD algorithm 14 that makes the training process more

efficient (Aldwairi et al. 2018; Tieleman 2008).The update process of parameter θ is as follows:

$$\frac{\partial \log p(v)}{\partial_{w_{ij}}} = p(h_i{=}1|v).\, v_i \, - \, \sum_{v'} p(v')p(h_j' = \, 1\big|v' \,).\, v_i' \tag{3.35}$$

$$\frac{\partial \log \, P(v)}{\partial b_i} = \, v_i - \, \sum_{v''} p(v').\, v' \tag{3.36}$$

$$\frac{\partial \log \, P(v)}{\partial c_j} = \, P\big(h_j = 1\big|v\big) - \, \sum_{v''} p(v')\, P(h'_j = 1|v') \tag{3.37}$$

An RBM can be layered on top of another RBM after it has been trained. The first RBM's hidden layer output is used as the visible layer input of the second RBM. As a result, multiple layers of RBMs can be layered to automatically extract different characteristics that indicate an increasingly more complicated structure in the data. In practice, we stack RBMs and use a greedy layer-wise unsupervised learning approach to train them. Each additional hidden layer is trained as an RBM during the training phase. The network settings are set once the DBNs have been trained. The network parameter, $\theta = (W, b, c)$ are used to set up a multi-layer feed-forward neural network's weights. The network parameters are $\theta = (W, b, c)$ fine-tuned using the back propagation technique to improve the detect performance of the neural network (Yang et al. 2019). The DBNs structural model is shown in Figure 3.6.

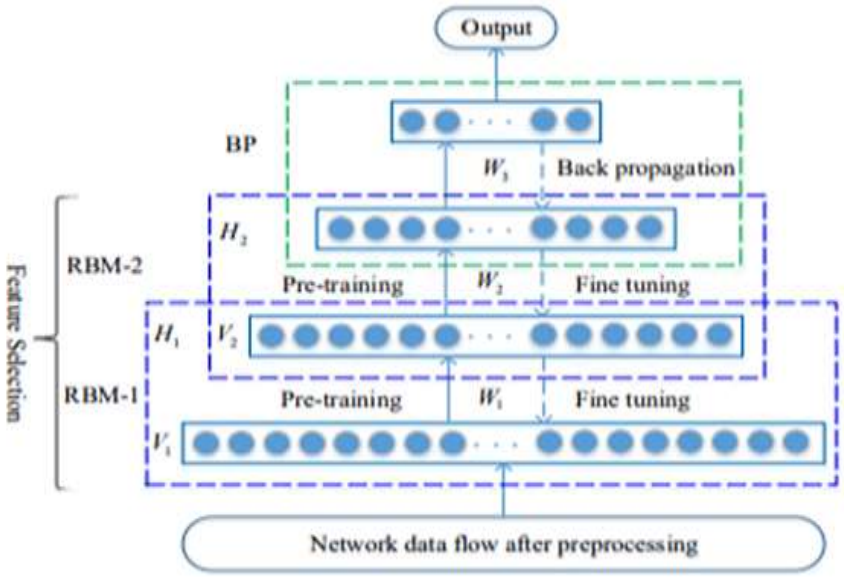

Figure 3.6. The DBNs structural model (Yang et al., 2019).

Algorithm 14 Training a deep belief network.

Input: training data $\{x_i\}_{i=1}^{n}$

// pre-training:

1: **for** l from $\ell - 1$ **do**

2: **if** $l = 1$ **then**

3: $\{v_i\}_{i=1}^{n} \leftarrow \{x_i\}_{i=1}^{n}$

4: **else**

// generate n hidden variables of previous

5: RBM:

6: $\leftarrow \{h_i\}_{i=1}^{n}$Algorithm 1 for $(l - 1)$ −th

7: RBM $\leftarrow \{v_i\}_{i=1}^{n} \leftarrow \{h_i\}_{i=1}^{n}$

8: $\boldsymbol{W_l}, \boldsymbol{b_l}, \boldsymbol{b_{l+1}} \leftarrow$ Algorithm 2 for $l - th$ RBM

9: $\leftarrow \{v_i\}_{i=1}^{n}$

// fine-tuning using backpropagation:

10: Initialize network with weights $\{\boldsymbol{W}_l\}_{l=1}^{\ell-1}$ and

11: biases $\{b_l\}_{l=2}^{\ell}$.

12: $\{\boldsymbol{W_l}\}_{l=1}^{\ell-1}$, $\{b_l\}_{l=1}^{\ell} \leftarrow$ Backpropagate the error of loss from several epochs.

The Table 3.2 is the Summary of suggested contributions and methodologies concept.

Table 3.2. Summary of suggested contributions and methodologies concept.

Research Objective	**Theme**	**Research Question**	**Methodology**	**Performance Measures**
For estimating the trajectory of pedestrians within an indoor environment	Develop a Kalman filter-based model for estimating the trajectory of pedestrians within an indoor environment based on	How to assess the trustworthiness of data sent/received by IoMT nodes? How to identify the compromised nodes within IoMT system?	Kalman filter-based model and fusing WiFi with IMU data.	Detection Accuracy, recall ,precision, Specifity, NPV, G-mean, F-measure
For detecting anomaly behaviour in IoMT	Trustworthiness assessment	How to detect anomaly behaviour in IoMT based on the estimated trajectory? How to generate IDs in IoMT environment?	A Cauchy-based clustering method for detecting anomaly behaviour in IoMT based on the estimated trajectory by Kalman filter. And an extreme learning machine-based IDS model for IoMT systems using the Cauchy clustering scheme for the training phase of an extreme learning machine to improve the detection accuracy.	Detection Accuracy, recall ,precision , Specifity, NPV, G-mean, F-measure
For the trust IDS model for IoMT systems	Evaluate the proposed algorithm using classification and networking metrics	How to improve the detection accuracy? How to evaluate the proposed algorithm?	Integrating anomaly detection with online learning for attacks identification using an Online Sequential Extreme learning machine (OSELM).	Detection Accuracy, recall ,precision , Specifity, NPV, G-mean, F-measure

3.11 UML Design

Unified Modeling Language is the class diagram of the platform illustrated in Figure 3.7. It consists of the following classes; pedestrian class, building class, device class, sink class, performance logging class and packet buffer class.

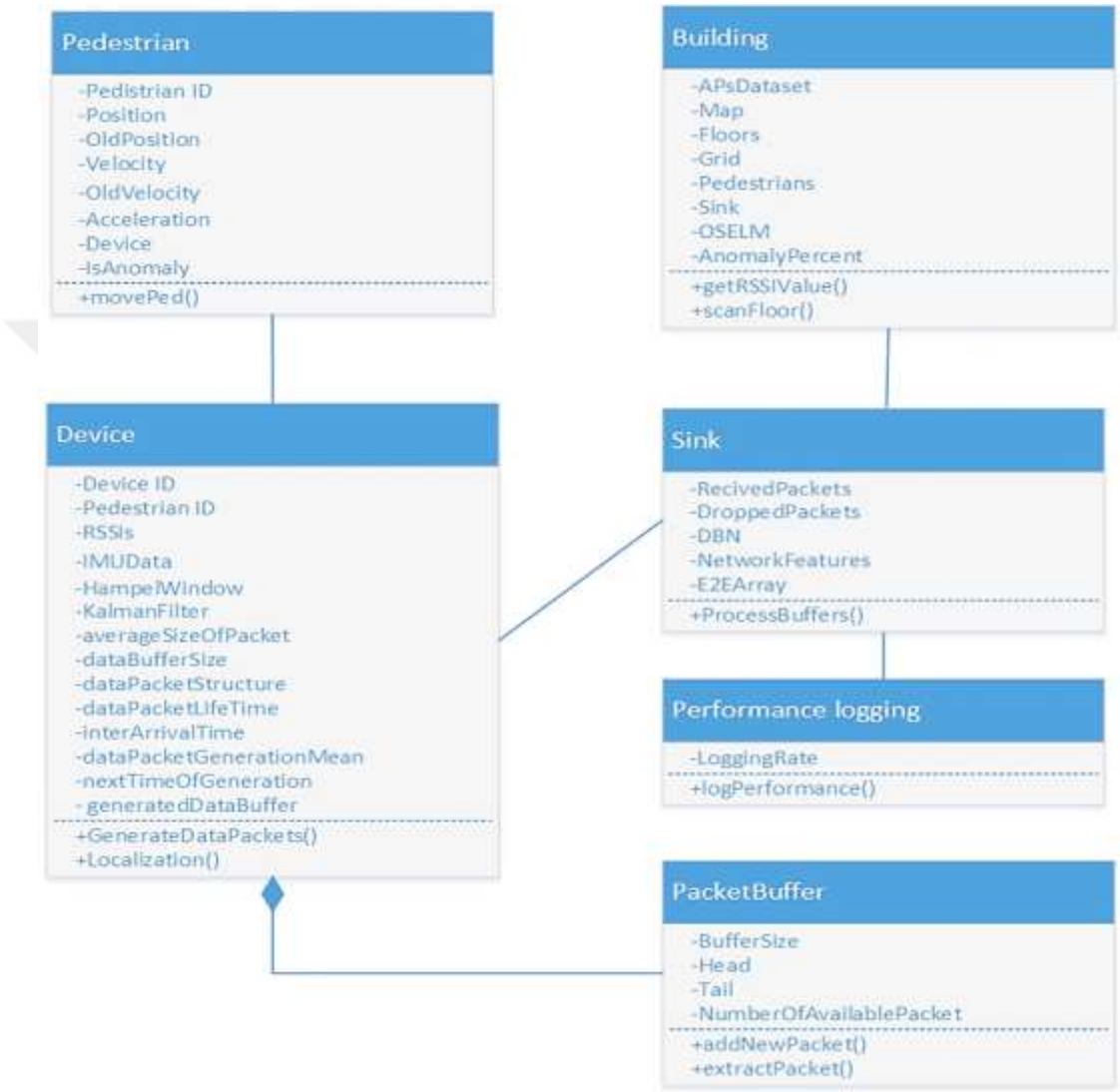

Figure 3.7. System class diagram.

3.12 WiFi Fingerprinting

The distribution of measurement points for first floor is presented in Figure 3.8.The measurement points means all the smart devices which are carried by the pedestrians and the fixed devices available in the first floor. And the distribution of the pedestrian movement simulator is presented in Figure 3.9 which showing the

movement of pedestrians through the first floor by built the simulator by using matlab program to simulate the reality.

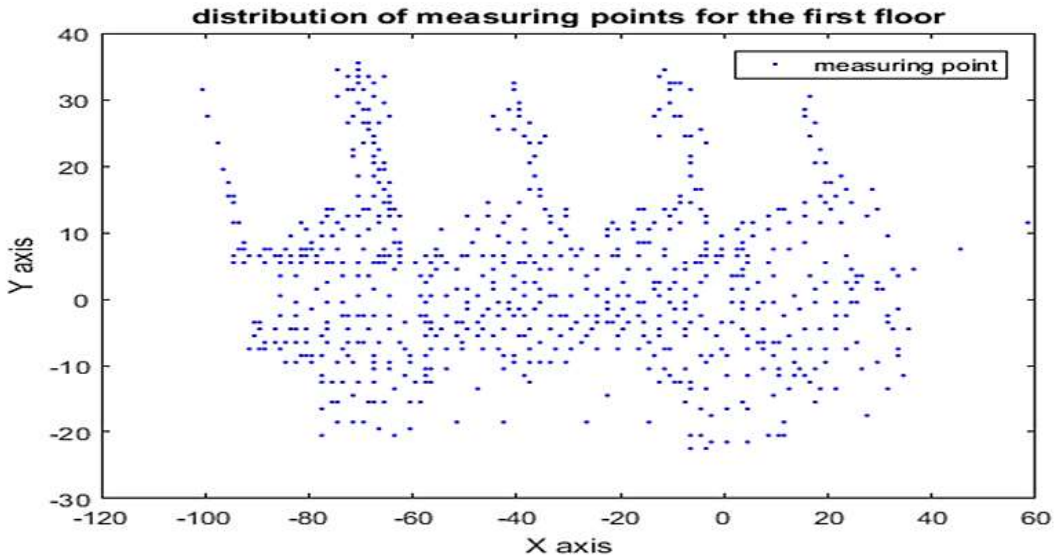

Figure 3.8. First floor layout.

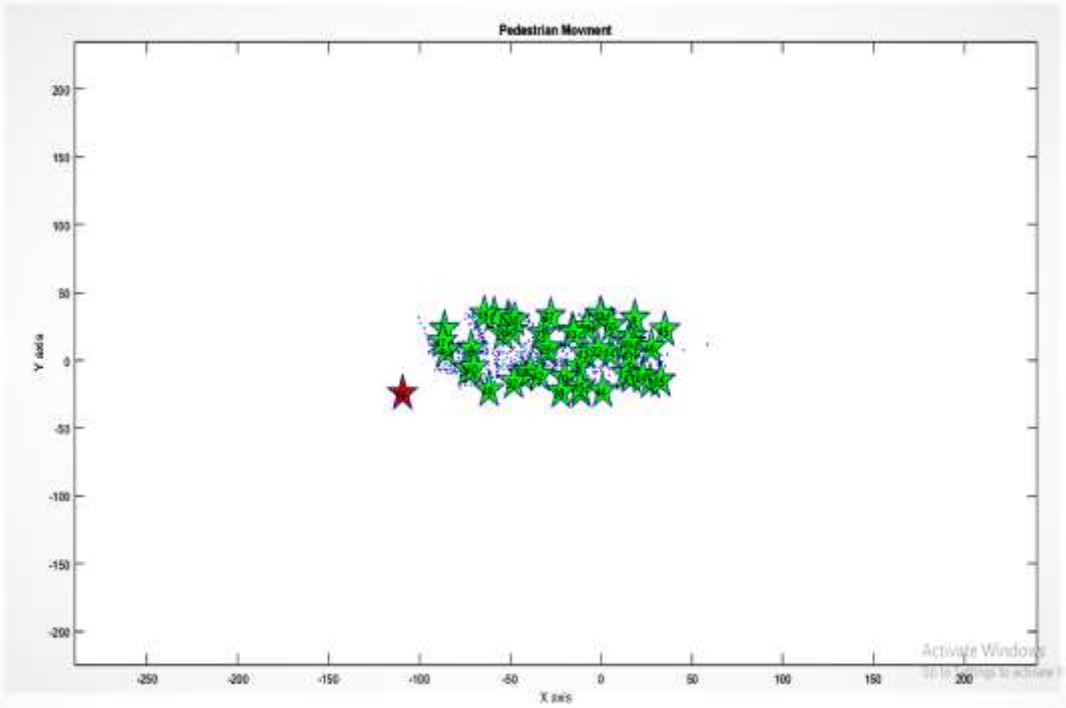

Figure 3.9. Pedestrian movement Simulator.

3.13 Dataset

We used two types of data sets in this study , to achieve the results of the first part which is related to anomaly detection we used the Tampere University Wi-Fi

fingerprint dataset (Lohan et al., 2017). And for the second part which is related to intrusion detection system we used the Knowledge Discovery and Data mining (KDD99) data set (Moustafa & Slay, 2015).

3.13.1 Anomaly detection dataset

We used Tampere University Wi-Fi fingerprint dataset, comprised of 4648 fingerprints collected with 21 devices in a university building in Tampere, Finland. We used the data of the first floor only as Aps Dataset. The floor layout illustrated in Figure 3.8. And the Simulator layout illustrated in Figure 3.9.

3.13.2 IDS dataset

The second part of this study is to implement IDS using Extreme Learning Machine (ELM) from the Online Sequential Extreme Learning Machine (OSELM). The KDD99 dataset was used to emulate the intrusions of DoS type. 48562 records split into 10000 records for training ELM -which has 50 hidden neurons and sig activation function and random records selected from the other part and used to emulate the network features during the simulation.

3.14 Evaluation Metrics

This section provides the various metrics used for evaluating our developed algorithm and its comparison with state-of-the-art approaches.

3.14.1 Accuracy

The accuracy is given by Equation (3.38).It is used to calculate the accuracy for anomaly and IDS by using anomaly percentage, numbers of pedestrians and the speed.

$$\mathrm{ACC} = \frac{(True\ positive)\mathrm{TP}+(True\ negative)\mathrm{TN}}{\mathrm{P}+\mathrm{N}} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+(Faks\ positive)\mathrm{FP}+(False\ negative)\mathrm{FN}} \tag{3.38}$$

3.14.2 Precision

Precision denotes the predictive position values for the anomaly and IDS by using anomaly percentage, numbers of pedestrians and the speed. And it is given by Equation (3.39).

$$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}} = 1 - (False\ positive\ rate)\text{FPR} \tag{3.39}$$

3.14.3 Recall

Recall denotes the true positive ratio for anomaly and IDS by using anomaly percentage, numbers of pedestrians and the speed and it is given by the Equation (3.40).

$$\text{TPR} = \frac{\text{TP}}{(\text{precision})\text{P}} = \frac{\text{TP}}{\text{TP} + \text{FN}} = 1 - (\ False\ negative\ rate\ )\text{FNR} \tag{3.40}$$

3.14.4 Specifity

Specifity denotes the false negative ratio for anomaly and IDS by using anomaly percentage, numbers of pedestrians and the speed and it is given by the Equation (3.41).

$$\text{(False positive rate) FPR} = \frac{\text{TN}}{\text{TN+FP}} = 1 - \text{(True positive rate) TPR} \tag{3.41}$$

3.14.5 G-Mean

Denotes the geometric mean of precision and recall after calculate each measure for anomaly and IDS by using anomaly percentage, numbers of pedestrians and the speed. It is given based on the Equation (3.42).

$$G - \text{ Mean } = \sqrt{\text{precision recall}} \tag{3.42}$$

3.14.6 F-measure

It combines the precision and recall which are calculated before; for anomaly and IDS by using anomaly percentage, numbers of pedestrians and the speed based on the Equation (3.43).

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{3.43}$$

3.14.7 NPV

It denotes to the percentage of true negative to total of negative records for anomaly and IDS by using anomaly percentage, numbers of pedestrians and the speed. It is given based on the Equation (3.44).

$$\text{(Negative predictive value)NPV} = \frac{\text{TN}}{\text{TN+FN}} \tag{3.44}$$

4. RESULT AND DISCUSSION

This section provides the experimental results and the analysis; it consists of experimental designs and setups as provided in subsection 4.1, and the analysis of the results as provided in Section 4.2.

4.1 Experimental Design And Setup

The experimental design is comprised of two parts; the first one is the design of the anomaly detection model and the second one is the design of the entire algorithms(KC-ELM , CEDAS , MUDI) for both anomaly and intrusion detection. Table 4.1 presents the parameters settings used in this work for the simulator like; TimeUnit ,Floor,Grid,HampleWindow and LoggingRate then we have to give value for each parameter ;the value of TimeUnit is 0.1 sec , the value for floor is one because we used only one floor which is the first floor,the grid is the dimensions of the program window which is represent the simulation of the floor,HampleWindow is the size of the hample window, the value of LoggingRate is two sec means the period of logging form pedestrian to Aps dose not exceeds two seconds. While Table 4.2 presents the settings of the algorithms like the parameters which includes the Decay, Radius, minThreshold, Streamspeed,C_m,, Cl, Lamda, gridGranularity, MinPts, Horizon, deltaS, gamaC .Each algorithm used it's parameters as shown in Table 4.2.

Table 4.1. Simulator Settings.

Parameters	Value	Description
TimeUnit	0.1 [Sec]	simulator time step
Floors	1	number of floors
Grid	[-90,40,-25,35,4]	[minX,maxX,minY,maxY,grid granularity]
HampleWindow	10	hample window size
LoggingRate	2 [Sec]	loggig data rate

Table 4.2. Algorithms parameters.

Parameters	KC-ELM	CEDAS (Islam et al. 2019)	MUDI (Amini et al., 2016)
Decay	-	Number of pedestrians	-
Radius	-	0.01	-
minThreshold	-	3	-
Stream speed	-	-	Number of pedestrians
C_m	-	-	1
Cl	-	-	0.5
Lamda	-	-	0.998
gridGranularity	-	-	10
MinPts	-	-	3
Horizon	-	-	2
deltaS	0.5	-	-
gammaC	0.66	-	-

4.2 Results Analysis

The results analysis consists of two sub-sections; the first is the analysis of the results of the anomaly detection model as given in Sub-section 4.2.1 while the second is the analysis of the results of the intrusion detection model as given in sub-section 4.2.2.

4.2.1 Anomaly detection results

The accuracy metric was generated for each of KC, MUDI, and CEDAS; firstly, the accuracy of the models based on the scenarios of inserted anomalies was presented in Table 4.3. It indicates the percentage of the true predicted records (anomaly versus non-anomaly) to the total number of records. Three scenarios were used for testing; the first one is for an anomaly percentage of 5%, the second one is for an anomaly percentage of 10%, and the third one is for an anomaly percentage of 20%. The analysis showed that KC maintained accuracy of over 85% for all the scenarios which was superior to the accuracies of MUDI and CEDAS; for anomaly detection of 5%, KC reached the accuracy of 94%. Additionally, the accuracy of the models was presented based on the number of pedestrians in Table 4.4. Like the cases of the percentage of inserted anomalies, it was found that for all scenarios, the accuracy was higher than 85% and reached 91% for 25 pedestrians. This showed that the integration of the Kalman-based state estimation with Cauchy in KC enabled anomaly detection and sustained the performance despite the increase in the number

of pedestrians or the increase in the percentage of anomalies. Table 4.5 provides the classification metrics for the models based on the average speed of pedestrians. Notably, KC performed better than the rest of the models (MUDI and CEDAS) in all the scenarios. The best-achieved accuracy was 86% for KC when the average speed was 1. Furthermore, the other classification metrics were also generated for the scenarios of three anomalies percentages, number of pedestrians, and average speed as presented in Tables 4.3, 4, and 5 respectively. It was also observed that KC performed better than MUDI and CEDAS in all the scenarios based on the considered metrics. The zero values presenting in some fields of tables which are generated by the program and the (-) mean that there are unknown values generated by the program. All figures and tables of the results in details shown in Appendix A which includesAppendix A: Anomaly Detection Results Diagrams And Tables (Appendix A- 1 : Metric Anomaly for 5%, 10% and 20% , Appendix A- 2 :Metric Pedestrians for 25, 50 and 100 pedestrians , Appendix A- 3 : Metric Speed for 1, 2 and 3 m\sec) .

Table 4.3. Summary of the evaluation metrics for our KC and its comparison with the benchmark in terms of anomaly detection for three anomalies percentages 5%, 10% and 20%.

Percentage	Measure	KC	MUDI	CEDAS
5%	Accuracy	94%	64%	86%
10%		93%	33%	83%
20%		86%	48%	84%
5%	Precision	24%	1%	0%
10%		54%	1%	10%
20%		60%	16%	51%
5%	Recall	100%	22%	-
10%		100%	13%	88%
20%		91%	37%	82%
5%	Specificity	94%	65%	86%
10%		93%	34%	83%
20%		85%	50%	85%
5%	NPVs	100%	98%	100%
10%		100%	86%	100%
20%		97%	76%	96%
5%	F measure	39%	2%	-
10%		70%	2%	17%
20%		72%	22%	63%
5%	G-mean	49%	5%	-
10%		74%	4%	29%
20%		74%	24%	64%

Table 4.4. Summary of the evaluation metrics for our KC and its comparison with the benchmark in terms of anomaly detection for three scenarios of number of pedestrians 25, 50 and 100.

Number of pedestrians	Measure	KC	MUDI	CEDAS
25	Accuracy	91%	20%	73%
50		86%	48%	84%
100		86%	27%	86%
25	Precision	68%	10%	42%
50		60%	16%	51%
100		57%	13%	60%
25	Recall	82%	30%	89%
50		91%	37%	82%
100		93%	45%	63%
25	Specificity	93%	16%	69%
50		85%	50%	85%
100		85%	22%	91%
25	NPVs	97%	42%	96%
50		97%	76%	96%
100		98%	62%	92%
25	F measure	75%	15%	57%
50		72%	22%	63%
100		71%	20%	62%
25	G-mean	75%	18%	61%
50		74%	24%	64%
100		73%	24%	62%

Table 4.5. Summary of the evaluation metrics for our KC and its comparison with the benchmark in terms of anomaly detection for three scenarios of average speed 1, 2 and 3 [m/s].

Average speed of pedestrian	Measure	KC	MUDI	CEDAS
1	Accuracy	86%	48%	84%
2		87%	56%	78%
3		88%	52%	79%
1	Precision	60%	16%	51%
2		61%	17%	41%
3		64%	15%	42%
1	Recall	91%	37%	82%
2		91%	32%	82%
3		90%	29%	83%
1	Specificity	85%	50%	85%
2		86%	62%	77%
3		87%	58%	78%
1	NPVs	97%	76%	96%
2		97%	78%	96%
3		97%	77%	96%
1	F measure	72%	22%	63%
2		73%	22%	55%
3		75%	20%	56%
1	G-mean	74%	24%	64%
2		75%	23%	58%
3		76%	21%	59%

4.2.2 IDS result

The results of the IDS were presented in different tables; Table 4.6 presents the results for the first scenarios that represent the percentages of inserted anomalies; Table 4.7 presents the results based on different numbers of pedestrians; and Table 4.8 that presents the results for different average speeds of the pedestrians. For the first set of scenarios given in Table 4.6,when the Kalman Cauchy estimate with the Extreme Learning Machine, KC ELM was found to be superior in terms of accuracy (achieving 43% accuracy for 20 % of inserted attacks) compared to 31% for Kalman Cauchy estimation with Deep Belief Network KC DBN, 29% for MUDI, and 21 % for CEDAS. Considering the imbalance in the dataset (that is KDD 99) in terms of having a low number of instances for some classes, and considering that KC is integrated with ELM which is a supervised classifier, this has enabled better prediction of performance. Additionally, we emphasize the fact that the classifier is learning in an online way, meaning that it started with almost zero knowledge and the knowledge keeps increasing with the arrival of more labeled batches to the

system. And all figures and tables details shown in Appendix B which includes; Appendix B: IDS Result Diagrams and Tables (Appendix B-1: Metric Anomaly for 5%, 10% and 20%, Appendix B- 2: Metric Pedestrians for 25, 50 and 100 pedestrian, Appendix B- 3: Metric Speed for 1, 2 and 3 m\sec).

Table 4.6. Summary of the evaluation metrics for our KC ELM and its comparison with the benchmark in terms of intrusion detection for three anomalies percentages 5%, 10% and 20%.

Percentage	Measure	KC ELM	KC DBN	MUDI	CEDAS
5%	Accuracy	30%	11%	29%	1%
10%		31%	13%	11%	5%
20%		43%	31%	29%	21%
5%	Precision	3%	2%	1%	1%
10%		9%	7%	6%	5%
20%		31%	27%	23%	21%
5%	Recall	100%	100%	100%	100%
10%		100%	100%	100%	89%
20%		100%	100%	100%	100%
5%	Specificity	28%	9%	29%	0%
10%		25%	6%	6%	0%
20%		23%	7%	10%	0%
5%	NPVs	100%	100%	100%	-
10%		100%	100%	100%	0%
20%		100%	100%	100%	-
5%	F measure	5%	4%	2%	1%
10%		17%	14%	12%	10%
20%		47%	42%	37%	34%
5%	G-mean	16%	15%	9%	8%
10%		30%	27%	25%	22%
20%		56%	52%	48%	46%

In addition to the scenario of different percentages of inserted anomalies, the comparisons were also presented in Table 4.7 based on different numbers of pedestrians. Observably, KC ELM achieved the best accuracy compared to the other models; its accuracy was the highest when the number of pedestrians was equaled to 50.

Table 4.7. Summary of the evaluation metrics for our KC ELM and its comparison with the benchmark in terms of intrusion detection for three scenarios of number of pedestrians 25, 50 and 100.

Number of Pedestrians	Measure	KC ELM	KC DBN	MUDI	CEDAS
25	Accuracy	42%	25%	31%	23%
50		43%	31%	29%	21%
100		41%	28%	35%	21%
25	Precision	24%	20%	23%	21%
50		31%	27%	23%	21%
100		26%	23%	24%	21%
25	Recall	100%	100%	100%	100%
50		100%	100%	100%	100%
100		100%	100%	100%	100%
25	Specificity	29%	8%	14%	4%
50		23%	7%	10%	0%
100		25%	9%	19%	0%
25	NPVs	100%	100%	100%	100%
50		100%	100%	100%	-
100		100%	100%	100%	-
25	F measure	39%	33%	37%	34%
50		47%	42%	37%	34%
100		42%	37%	39%	34%
25	G-mean	49%	45%	47%	46%
50		56%	52%	48%	46%
100		51%	47%	49%	45%

Similarly, the KC-ELM achieved the best accuracies for the scenario of average speed, reaching an accuracy value of 43% compared to the other algorithms. The comparisons were also presented in Table 4.8 based on different numbers of average speed. Furthermore, it was independent of the average speed of pedestrians while the accuracy for MUDI decreased from 2% at the average speed of 1 m/s to 22% at the average speed of 3 m/s.In addition to accuracy, the tables provided the other classification metrics, namely precision, recall, specificity, NPV, F-measure, and G-mean. The results showed that KC-ELM achieved higher performance metrics compared to KC-DBN, MUDI, and CEDAS.The E2E Delay and PDR for anomaly detection for the pedesterians ,anomaly and speed metrics shown in details in Appendix C, while the E2E Delay and PDR for IDS for the pedesterians ,anomaly and speed metrics shown in details in Appendix D.

Table 4.8. Summary of the evaluation metrics for our KC ELM and its comparison with the benchmark in terms of intrusion detection for three scenarios of number of pedestrians 1, 2 and 3.

Average speed of pedestrian	Measure	KC ELM	KC DBN	MUDI	CEDAS
1	Accuracy	43%	31%	29%	21%
2		43%	31%	24%	21%
3		43%	31%	22%	21%
1	Precision	31%	27%	23%	21%
2		31%	27%	21%	21%
3		31%	27%	21%	21%
1	Recall	100%	100%	100%	100%
2		100%	100%	100%	100%
3		100%	100%	100%	100%
1	Specificity	23%	7%	10%	0%
2		23%	7%	4%	0%
3		23%	7%	2%	0%
1	NPVs	100%	100%	100%	-
2		100%	100%	100%	-
3		100%	100%	100%	-
1	F measure	47%	42%	37%	34%
2		47%	42%	35%	34%
3		47%	42%	35%	34%
1	G-mean	56%	52%	48%	46%
2		56%	52%	46%	46%
3		56%	52%	46%	46%

5. CONCLUSION AND FUTURE WORK

5.1 Conclusion

This thesis has provided a novel approach for handling intrusion on the Internet of Mobility Things networks by exploiting the anomaly generated from the intruder and reflected in its mobility behavior. Hence, this study has included a location prediction model based on Wi-Fi location prediction using fingerprint trained online sequential extreme learning machine for indoor environment and based on fusion with IMU using the developed process and measurement model for Kalman filter. Next, the result of the mobility estimation produced by the developed Kalman is fed into Cauchy-based stream clustering for identifying anomalies. The results of the predicted anomalies recognized from location prediction are added to the generated packets in the network as security features and used for improving the results of the intrusion detection (this is another online sequential extreme learning machine). The developed algorithm was compared with two existing models for anomaly detection, namely, a multi-density clustering algorithm for evolving data stream (MUDI) and fully online clustering of evolving data streams into arbitrarily shaped clusters (CEDAS). The results proved the superiority of the developed algorithm in this study in terms of anomaly and intrusion detection under three different scenarios that include different percentages of added anomalies, different numbers of pedestrians, and different average speeds of pedestrians. This study uses the Extreme Learning Machine (ELM) classifier to develop a Kalman and Cauchy clustering for anomaly detection and uses it for authenticating nodes within IoMTs. The Kalman filter-based model for calculating the trajectory of pedestrians within an interior environment is based on merging WiFi with IMU data from the algorithm's many components. Then comes the evaluation of the validity of the Kalman filter-based predicted trajectory for IoMT abnormal behavior detection. Finally, the Online Sequential Extreme Learning Machine-based Trust IDS Model for IoMT Systems integrates anomaly detection with online learning for attackers' identification (OSELM). The TamperU dataset for WiFi fingerprinting and KDD99 for intrusion detection have both been used in the implementation and evaluation of the OSELM algorithm.

5.2 Future Work

Future works can be directed to the evaluation of the system from the perspective of networking performance to show the enhanced achieved by the developed algorithm on the networking metrics. The researchers may increase the range of distances between the trajectory pedestrians by using more floors or buildings to detect the intrusions. Or may use more effective algorithms to evaluate the trajectory pedestrians if they have normal or abnormal anomaly behaviour.

REFERENCES

Akyildiz, I. F., Lee, W.-Y., Vuran, M. C. and Mohanty, S., 2006. NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey. Computer Networks, 50, 13, 2127–2159.

Al-rimy, B. A. S., Maarof, M. A. and Shaid, S. Z. M., 2018. Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions. Computers and Security, 74, 144–166.

Albahri, A. S., Alwan, J. K., Taha, Z. K., Ismail, S. F., Hamid, R. A., Zaidan, A. A., Albahri, O. S., Zaidan, B. B., Alamoodi, A. H. and Alsalem, M. A., 2021. IoT-based telemedicine for disease prevention and health promotion: State-of-the-Art. Journal of Network and Computer Applications, 173, 102873.

Aldwairi, T., Perera, D. and Novotny, M. A., 2018. An evaluation of the performance of Restricted Boltzmann Machines as a model for anomaly network intrusion detection. Computer Networks, 144, 111–119.

Aljawarneh, S. A. and Vangipuram, R,., 2020. GARUDA: Gaussian dissimilarity measure for feature representation and anomaly detection in Internet of things. The Journal of Supercomputing, 76, 6, 4376–4413.

Allig, C., Leinmüller, T., Mittal, P. and Wanielik, G., 2019. Trustworthiness estimation of entities within collective perception., 2019 IEEE Vehicular Networking Conference (VNC), 1–8.

Althobaiti, O. S., 2020. Internet of (Moving) Things: Paradigms and Opportunities. NB! ICT Innovation, Regulation, Multi Business Model Innovation and Technology, 237–262.

Amini, A., Saboohi, H., Herawan, T. and Wah, T. Y., 2016. MuDi-Stream: A multi density clustering algorithm for evolving data stream. Journal of Network and Computer Applications, 59, 370–385.

Amini, A., Saboohi, H., Ying Wah, T. and Herawan, T., 2014. A fast density-based clustering algorithm for real-time internet of things stream. The Scientific World Journal, 2014.

Anthi, E., Williams, L., Słowińska, M., Theodorakopoulos, G. and Burnap, P., 2019. A supervised intrusion detection system for smart home IoT devices. IEEE Internet of Things Journal, 6, 5, 9042–9053.

Arshad, J., Azad, M. A., Abdeltaif, M. M. and Salah, K., 2020. An intrusion detection framework for energy constrained IoT devices. Mechanical Systems and Signal Processing, 136, 106436.

Atul, D. J., Kamalraj, R., Ramesh, G., Sankaran, K. S., Sharma, S. and Khasim, S., 2021. A machine learning based IoT for providing an intrusion detection system for security. Microprocessors and Microsystems, 82, 103741.

Bagaa, M., Taleb, T., Bernabe, J. B. and Skarmeta, A., 2020. A machine learning security framework for iot systems. IEEE Access, 8, 114066–114077.

Borgia, E., 2014. The Internet of Things vision: Key features, applications and open issues. Computer Communications, 54, 1–31.

Bostani, H. and Sheikhan, M., 2017. Hybrid of anomaly-based and specification-based IDS for Internet of Things using unsupervised OPF based on MapReduce approach. Computer Communications, 98, 52–71.

Breitenbacher, D., Homoliak, I., Aung, Y. L., Tippenhauer, N. O. and Elovici, Y., 2019. HADES-IoT: A practical host-based anomaly detection system for IoT devices. Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security, 479–484.

Chen, Y., Kar, S. and Moura, J. M. F., 2018. The internet of things: Secure distributed inference. IEEE Signal Processing Magazine, 35, 5, 64–75.

Chung, J. and Kim, H.-J., 2020. An automobile environment detection system based on deep neural network and its implementation using IoT-enabled in-vehicle air quality sensors. Sustainability, 12, 6, 2475.

Dai, X., Cheng, J., Gao, Y., Guo, S., Yang, X., Xu, X. and Cen, Y., 2020. Deep belief network for feature extraction of urban artificial targets. Mathematical Problems in Engineering, 2020.

De Benedetti, M., Leonardi, F., Messina, F., Santoro, C. and Vasilakos, A., 2018. Anomaly detection and predictive maintenance for photovoltaic systems. Neurocomputing, 310, 59–68.

de Souza, A. M., Yokoyama, R. S., Maia, G., Loureiro, A. A. F. and Villas, L. A., 2015. Minimizing traffic jams in urban Centers using vehicular ad hoc networks., 2015 7th International Conference on New Technologies, Mobility and Security (NTMS), 1–5.

Dong, S., Liu, J., Liu, Y., Zeng, L., Xu, C. and Zhou, T., 2018. Clustering based on grid and local density with priority-based expansion for multi-density data. Information Sciences, 468, 103–116.

Fan, Q., Zhang, H., Pan, P., Zhuang, X., Jia, J., Zhang, P., Zhao, Z., Zhu, G. and Tang, Y., 2019. Improved pedestrian dead reckoning based on a robust adaptive Kalman filter for indoor inertial location system. Sensors, 192, 294.

Farzaneh, B., Montazeri, M. A. and Jamali, S., 2019. An anomaly-based IDS for detecting attacks in RPL-based internet of things., 2019 5th International Conference on Web Research (ICWR), 61–66.

Foley, J., Moradpoor, N. and Ochenyi, H., 2020. Employing a machine learning approach to detect combined internet of things attacks against two objective functions using a novel dataset. Security and Communication Networks.

Gaddam, A., Wilkin, T., Angelova, M. and Gaddam, J., 2020. Detecting sensor faults, anomalies and outliers in the internet of things: A survey on the challenges and solutions. Electronics, 9, 3, 511.

Ghaleb, B., Al-Dubai, A., Ekonomou, E., Qasem, M., Romdhani, I. and Mackenzie, L., 2018. Addressing the DAO insider attack in RPL's Internet of Things networks. IEEE Communications Letters, 23, 1, 68–71.

Ghaleb, F. A., Maarof, M. A., Zainal, A., Al-Rimy, B. A. S., Saeed, F. and Al-Hadhrami, T., 2019. Hybrid and multifaceted context-aware misbehavior detection model for vehicular ad hoc network. IEEE Access, 7, 159119–159140.

Ghaleb, F. A., Maarof, M. A., Zainal, A., Rassam, M. A., Saeed, F. and Alsaedi, M., 2019. Context-aware data-centric misbehaviour detection scheme for vehicular ad hoc networks using sequential analysis of the temporal and spatial correlation of the consistency between the cooperative awareness messages. Vehicular Communications, 20, 100186.

Ghojogh, B., Ghodsi, A., Karray, F. and Crowley, M., 2021. Restricted Boltzmann Machine and Deep Belief Network: Tutorial and Survey. ArXiv Preprint ArXiv:2107.12521.

Granjal, J., Monteiro, E. and Silva, J. S., 2015. Security for the internet of things: a survey of existing protocols and open research issues. IEEE Communications Surveys and Tutorials, 17, 3, 1294–1312.

Halder, S., Ghosal, A. and Conti, M., 2019. Efficient physical intrusion detection in Internet of Things: A Node deployment approach. Computer Networks, 154, 28–46.

Han, J., Qian, C., Wang, X., Ma, D., Zhao, J., Xi, W., Jiang, Z. and Wang, Z., 2015. Twins: Device-free object tracking using passive tags. IEEE/ACM Transactions on Networking, 24, 3, 1605–1617.

He, D., Qiao, Q., Gao, Y., Zheng, J., Chan, S., Li, J. and Guizani, N., 2019. Intrusion detection based on stacked autoencoder for connected healthcare systems. IEEE Network, 33, 6, 64–69.

Hinton, G. E., Osindero, S. and Teh, Y.-W., 2006. A fast learning algorithm for deep belief nets. Neural Computation, 18, 7, 1527–1554.

Hong, S., Kim, Y. and Kim, G. J., 2017. Developing usable interface for Internet of Things (IoT) security analysis software. International Conference on Human Aspects of Information Security, Privacy, and Trust, 322–328.

Hu, Y., Dong, M., Ota, K., Liu, A. and Guo, M., 2014. Mobile target detection in wireless sensor networks with adjustable sensing frequency. IEEE Systems Journal, 10, 3, 1160–1171.

Hyde, R., Angelov, P. and MacKenzie, A. R., 2017. Fully online clustering of evolving data streams into arbitrarily shaped clusters. Information Sciences, 382, 96–114.

Islam, M. K., Ahmed, M. M. and Zamli, K. Z., 2019. A buffer-based online clustering for evolving data stream. Information Sciences, 489, 113–135.

Jiang, X., Liu, J., Chen, Y., Liu, D., Gu, Y. and Chen, Z., 2016. Feature adaptive online sequential extreme learning machine for lifelong indoor localization. Neural Computing and Applications, 27, 1, 215–225.

Kang, X., Song, B. and Sun, F., 2019. A deep similarity metric method based on incomplete data for traffic anomaly detection in IoT. Applied Sciences, 9, 1, 135.

Katneni, N., Pandit, V., Li, H. and Agrawal, D. P., 2012. Hybrid gaussian-ring deployment for intrusion detection in wireless sensor networks., 2012 IEEE International Conference on Communications (ICC), 549–553.

Keung, G. Y., Li, B. and Zhang, Q., 2012. The intrusion detection in mobile sensor network. IEEE/ACM Transactions On Networking, 20,4, 1152–1161.

Kumar, V. V., Devi, M., Raja, P. V., Kanmani, P., Priya, V., Sudhakar, S. and Sujatha, K., 2020. Design of peer-to-peer protocol with sensible and secure IoT communication for future internet architecture. Microprocessors and Microsystems, 78, 103216.

Li, A., Liu, W., Zhang, S. and Xie, M., 2020. Fast multicast with adjusting transmission power and active slots in software define IoT. IEEE Access, 8, 226352–226369.

Li, B., Hao, Z. and Dang, X., 2019. An indoor location algorithm based on Kalman filter fusion of ultra-wide band and inertial measurement unit. AIP Advances, 9, 8, 85210.

Li, W., Meng, W., Wang, Y., Han, J. and Li, J., 2018. Towards securing challenge-based collaborative intrusion detection networks via message verification. International Conference on Information Security Practice and Experience, 313–328.

Li, W., Meng, W., Wang, Y., Kwok, L. F. and Lu, R., 2018. Identifying passive message fingerprint attacks via honey challenge in collaborative intrusion detection networks., 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), 1208–1213.

Li, W., Tug, S., Meng, W. and Wang, Y., 2019. Designing collaborative blockchained signature-based intrusion detection in IoT environments. Future Generation Computer Systems, 96, 481–489.

Li, X. and Zou, B., 2021. An Automated Data Engineering Pipeline for Anomaly Detection of IoT Sensor Data. ArXiv Preprint ArXiv:2109.13828.

Lo, S. K., Liew, C. S., Tey, K. S. and Mekhilef, S., 2019. An interoperable component-based architecture for data-driven IoT system. Sensors, 19, 20, 4354.

Lohan, E. S., Torres-Sospedra, J., Leppäkoski, H., Richter, P., Peng, Z. and Huerta, J., 2017. Wi-Fi crowdsourced fingerprinting dataset for indoor positioning. Data, 2, 4, 32.

Lyu, L., Jin, J., Rajasegarar, S., He, X. and Palaniswami, M., 2017. Fog-empowered anomaly detection in IoT using hyperellipsoidal clustering. IEEE Internet of Things Journal, 4, 5, 1174–1184.

Madsen, D., Li, W., Meng, W. and Wang, Y., 2018. Evaluating the impact of intrusion sensitivity on securing collaborative intrusion detection networks against SOOA. International Conference on Algorithms and Architectures for Parallel Processing, 481–494.

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D. and Elovici, Y., 2018. N-baiot—network-based detection of iot botnet attacks using deep autoencoders. IEEE Pervasive Computing, 17, 3, 12–22.

Moustafa, N. and Slay, J., 2015. The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems., 2015 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), 25–31.

Moustafa, N., Turnbull, B. and Choo, K.-K. R., 2018. An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. IEEE Internet of Things Journal, 6, 3, 4815–4830.

Nõmm, S. and Bahşi, H., 2018. Unsupervised anomaly based botnet detection in IoT networks., 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 1048–1053.

Pamukov, M. E., Poulkov, V. K. and Shterev, V. A., 2018. Negative selection and neural network based algorithm for intrusion detection in IoT., 2018 41st International Conference on Telecommunications and Signal Processing (TSP), 1–5.

Pan, Z., Hariri, S. and Pacheco, J., 2019. Context aware intrusion detection for building automation systems. Computers and Security, 85, 181–201.

Prabhakaran, V. and Kulandasamy, A., 2021. Integration of recurrent convolutional neural network and optimal encryption scheme for intrusion detection with secure data storage in the cloud. Computational Intelligence, 37, 1, 344–370.

Qadri, Y. A., Nauman, A., Zikria, Y. Bin, Vasilakos, A. V, and Kim, S. W., 2020. The future of healthcare internet of things: a survey of emerging technologies. IEEE Communications Surveys and Tutorials, 2, 22, 1121–1167.

Raza, S., Wallgren, L. and Voigt, T., 2013. SVELTE: Real-time intrusion detection in the Internet of Things. Ad Hoc Networks, 11, 8, 2661–2674.

Salman, O., Elhajj, I. H., Chehab, A. and Kayssi, A., 2019. A machine learning based framework for IoT device identification and abnormal traffic detection. Transactions on Emerging Telecommunications Technologies, e3743.

Shafiq, M., Tian, Z., Bashir, A. K., Du, X. and Guizani, M., 2020. IoT malicious traffic identification using wrapper-based feature selection mechanisms. Computers and Security, 94, 101863.

Singh, R., Kumar, H. and Singla, R. K., 2015. An intrusion detection system using network traffic profiling and online sequential extreme learning machine. Expert Systems with Applications, 42, 22, 8609–8624.

Singh, T. and Kumar, N., 2020. Withdrawn: machine learning models for intrusion detection in iot environment: a comprehensive review. Elsevier.

Škrjanc, I., Ozawa, S., Ban, T. and Dovžan, D., 2018. Large-scale cyber attacks monitoring using evolving cauchy possibilistic clustering. Applied Soft Computing, 62, 592–601.

Slavic, G., Campo, D., Baydoun, M., Marin, P., Martin, D., Marcenaro, L. and Regazzoni, C., 2020. Anomaly detection in video data based on probabilistic latent space models., 2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), 1–8.

Sung, K., Lee, D. K. and Kim, H., 2018. Indoor pedestrian localization using iBeacon and improved Kalman filter. Sensors, 18, 6, 1722.

Tally, M. T. and Amintoosi, H., 2021. A hybrid method of genetic algorithm and support vector machine for intrusion detection. International Journal of Electrical and Computer Engineering (2088-8708), 11, 1.

Tama, B. A., Comuzzi, M. and Rhee, K.-H., 2019. TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system. IEEE Access, 7, 94497–94507.

Teng, H., Liu, Y., Liu, A., Xiong, N. N., Cai, Z., Wang, T. and Liu, X., 2019. A novel code data dissemination scheme for Internet of Things through mobile vehicle of smart cities. Future Generation Computer Systems, 94, 351–367.

Thanigaivelan, N. K., Nigussie, E., Kanth, R. K., Virtanen, S. and Isoaho, J., 2016. Distributed internal anomaly detection system for Internet-of-Things., 2016 13th IEEE Annual Consumer Communications and Networking Conference (CCNC), 319–320.

Thanigaivelan, N. K., Nigussie, E., Virtanen, S. and Isoaho, J., 2018. Hybrid internal anomaly detection system for IoT: Reactive nodes with cross-layer operation. Security and Communication Networks, 2018.

Tian, Jie, Wang, G., Yan, T. and Zhang, W., 2014. Detect smart intruders in sensor networks by creating network dynamics. Computer Networks, 62, 182–196.

Tian, Junfeng, Ding, W., Wu, C. and Nam, K. W., 2019. A Generalized Approach for Anomaly Detection from the Internet of Moving Things. IEEE Access, 7, 144972–144982. https://doi.org/10.1109/ACCESS.2019.2945205

Tieleman, T., 2008. Training restricted Boltzmann machines using approximations to the likelihood gradient. Proceedings of the 25th International Conference on Machine Learning, 1064–1071.

Verma, A. and Ranga, V., 2020. Machine learning based intrusion detection systems for IoT applications. Wireless Personal Communications, 111, 4, 2287–2310.

Wang, B., Kong, W., Guan, H. and Xiong, N. N., 2019. Air quality forecasting based on gated recurrent long short term memory model in Internet of Things. IEEE Access, 7, 69524–69534.

Weng, Y. and Liu, L., 2019. A collective anomaly detection approach for multidimensional streams in mobile service security. IEEE Access, 7, 49157–49168.

Xia, H., Hu, C., Xiao, F., Cheng, X. and Pan, Z., 2019. An efficient social-like semantic-aware service discovery mechanism for large-scale Internet of Things. Computer Networks, 152, 210–220.

Yahyaoui, A., Abdellatif, T. and Attia, R., 2019. Hierarchical anomaly based intrusion detection and localization in IoT., 2019 15th International Wireless Communications and Mobile Computing Conference (IWCMC), 108–113.

Yang, K., Ren, J., Zhu, Y. and Zhang, W., 2018. Active learning for wireless IoT intrusion detection. IEEE Wireless Communications, 25, 6, 19–25.

Yang, Y., Zheng, K., Wu, C., Niu, X. and Yang, Y., 2019. Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks. Applied Sciences, 9, 2, 238.

Yao, Z., Xie, J., Tian, Y. and Huang, Q., 2019. Using hampel identifier to eliminate profile-isolated outliers in laser vision measurement. Journal of Sensors, 2019.

Zeng, M., Yadav, A., Dobre, O. A. and Poor, H. V., 2019. Energy-efficient joint user-RB association and power allocation for uplink hybrid NOMA-OMA. IEEE Internet of Things Journal, 6(3), 5119–5131.

Zhou, M., Han, L., Lu, H. and Fu, C., 2021. Intrusion detection system for IoT heterogeneous perceptual network. Mobile Networks and Applications, 26, 4, 1461–1474.

Zou, H., Lu, X., Jiang, H. and Xie, L., 2015. A fast and precise indoor localization algorithm based on an online sequential extreme learning machine. Sensors, 15, 1, 1804–1824.

APPINDEX LIST

APPENDIX A : Anomaly Detection Results Diagrams And Tables

APPENDIX A- 1 : Metric Anomaly for 5%, 10% and 20%.

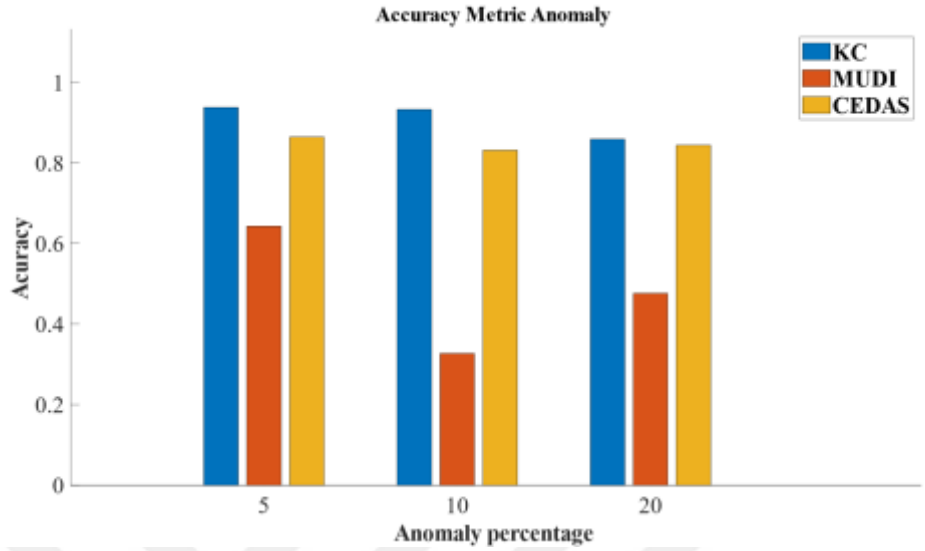

Figure A1.1. Accuracy Metric Anomaly.

Table A1.1. Accuracy Metric Anomaly.

KC	MUDI	CEDAS
0.937	0.642	0.863
0.932	0.326	0.831
0.858	0.475	0.843

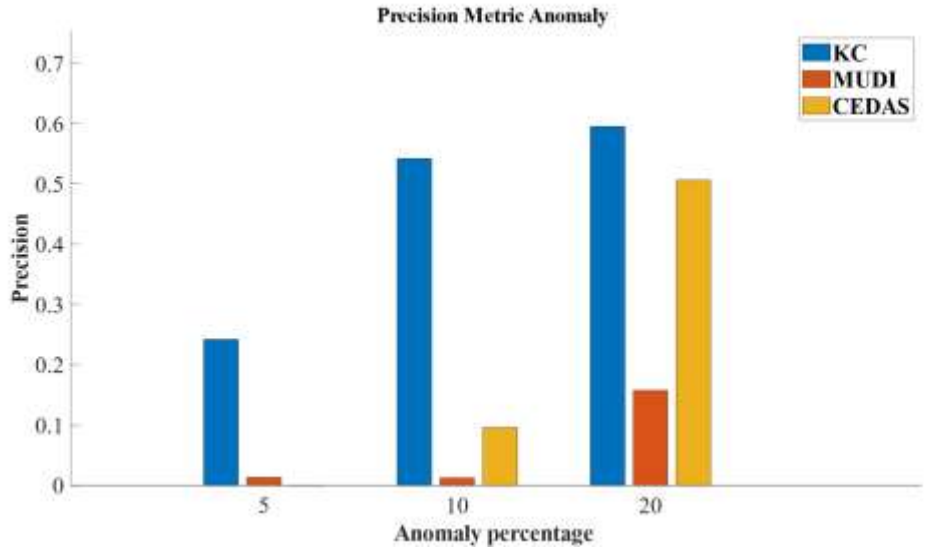

Figure A1.2. Precision Metric Anomaly.

Table A1.2. Precision Metric Anomaly.

KC	MUDI	CEDAS

0.241	0.012	0
0.541	0.011	0.095
0.595	0.157	0.506

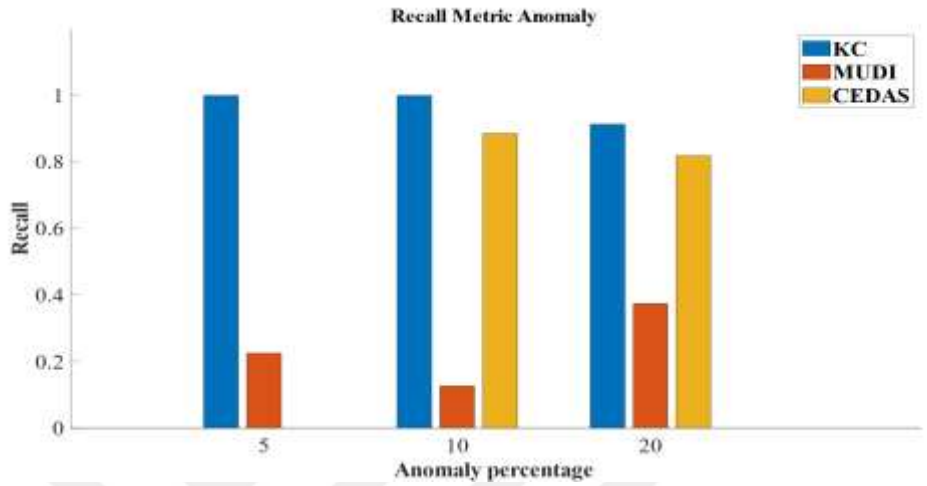

Figure A1.3. Recall Metric Anomaly.

Table A1.3. Recall Metric Anomaly.

KC	MUDI	CEDAS
0.997	0.224	-
0.997	0.125	0.883
0.912	0.373	0.817

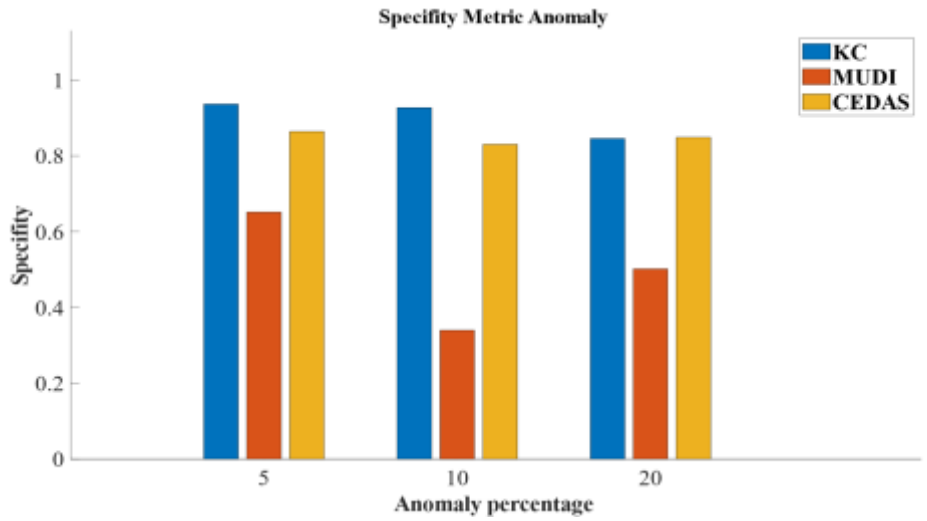

Figure A1.4. Specifity Metric Anomaly.

Table A1.4. Specifity Metric Anomaly.

KC	MUDI	CEDAS

0.936	0.651	0.863
0.926	0.339	0.830
0.845	0.501	0.848

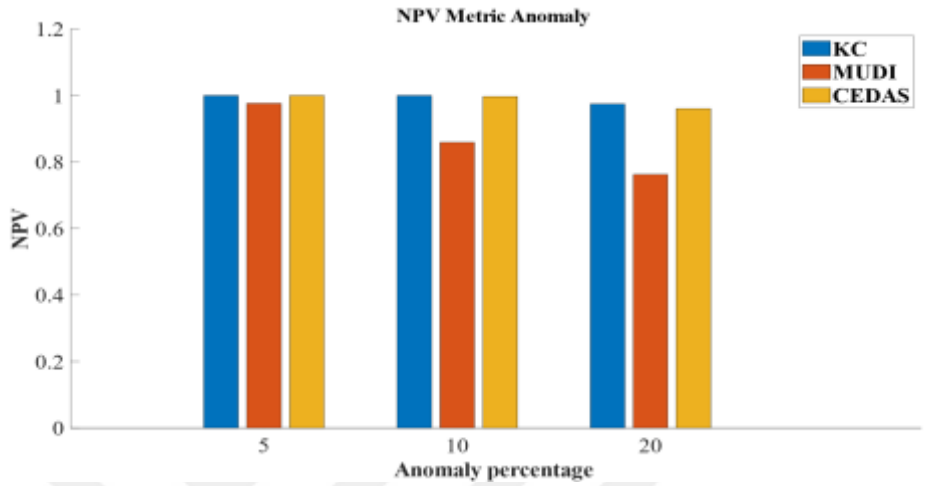

Figure A1.5. NPVs Metric Anomaly.

Table A1.5. NPVs Metric Anomaly.

KC	MUDI	CEDAS
0.999	0.976	1.000
0.999	0.859	0.997
0.974	0.762	0.960

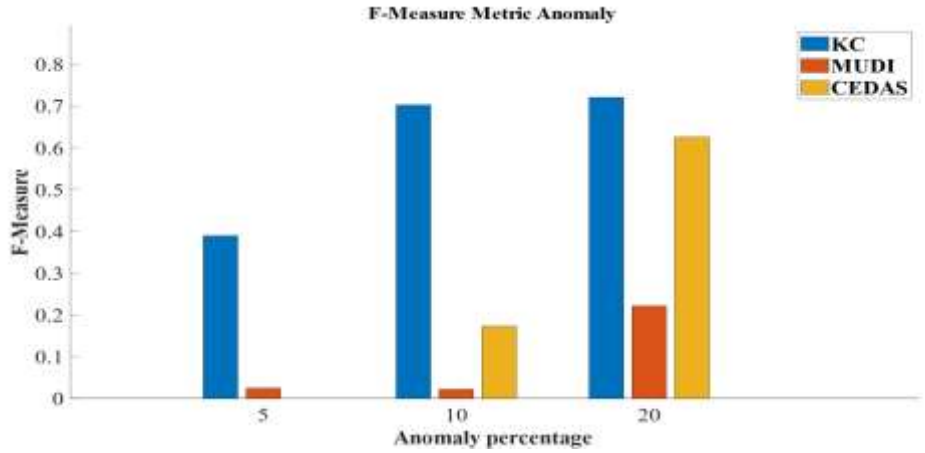

Figure A1.6. F-Measure Metric Anomaly.

Table A1.6. F-Measure Metric Anomaly.

KC	MUDI	CEDAS
0.3893	0.024	-

0.702	0.021	0.172
0.720	0.221	0.625

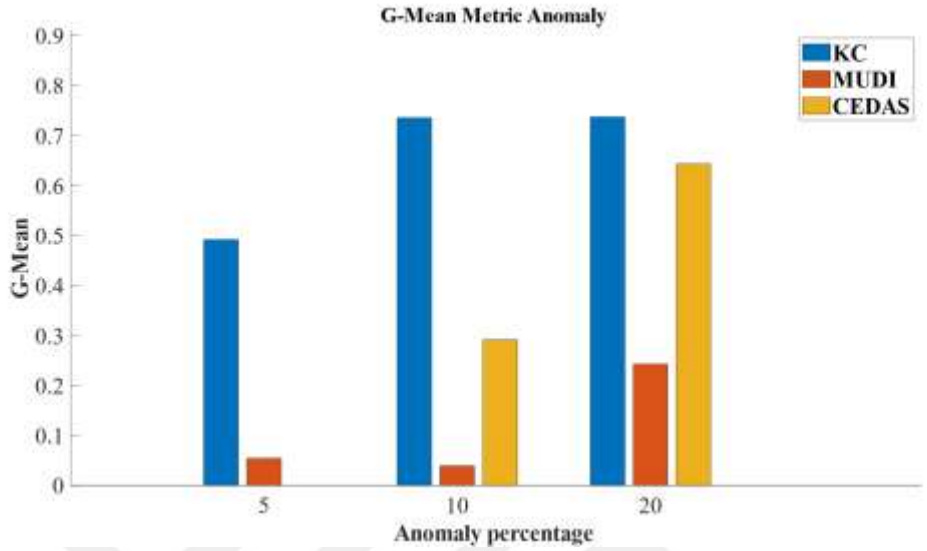

Figure A1.7. G-Mean Metric Anomaly.

Table A1.7. G-Mean Metric Anomaly.

KC	**MUDI**	**CEDAS**
0.491	0.053	-
0.735	0.038	0.291
0.736	0.242	0.643

APPENDIX A- 2 : Metric Pedestrians for 25, 50 and 100 pedesterians.

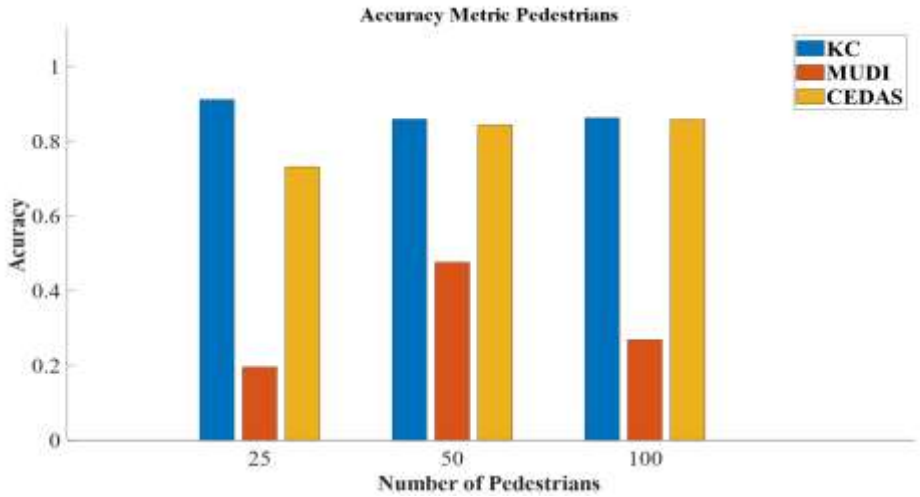

Figure A2.1. Accuracy Metric Pedestrians.

Table A2.1. Accuracy Metric Pedestrians.

KC	MUDI	CEDAS
0.910	0.195	0.731
0.858	0.475	0.843
0.861	0.268	0.857

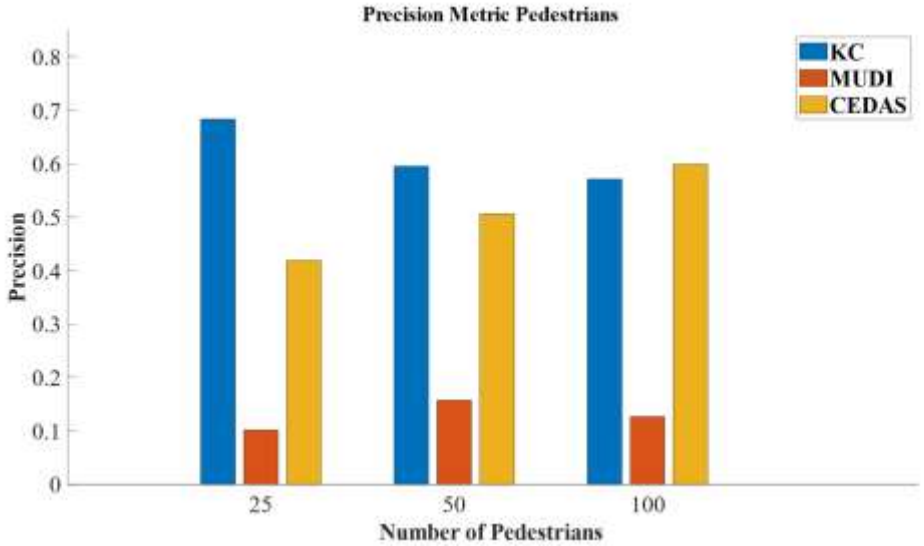

Figure A2.2. Precision Metric Pedestrians.

Table A2.2. Precision Metric Pedestrians.

KC	MUDI	CEDAS
0.683	0.101	0.418
0.595	0.157	0.506
0.571	0.126	0.598

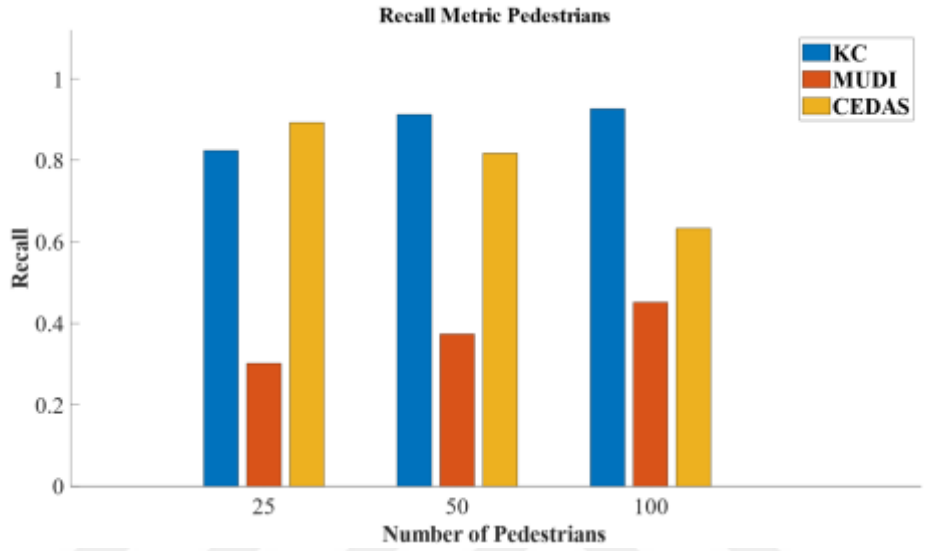

Figure A2.3. Recall Metric Pedestrians.

Table A2.3. Recall Metric Pedestrians.

KC	MUDI	CEDAS
0.823	0.301	0.892
0.912	0.373	0.817
0.9261	0.451	0.632

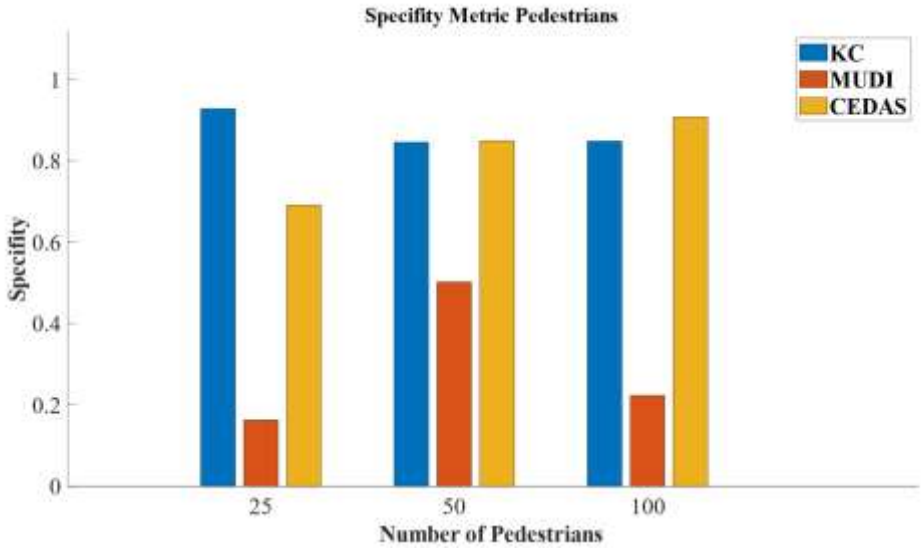

Figure A2.4. Specifity Metric Pedestrians.

Table A2.4. Specifity Metric Pedestrians.

KC	MUDI	CEDAS
0.927	0.162	0.690
0.845	0.501	0.848
0.847	0.223	0.907

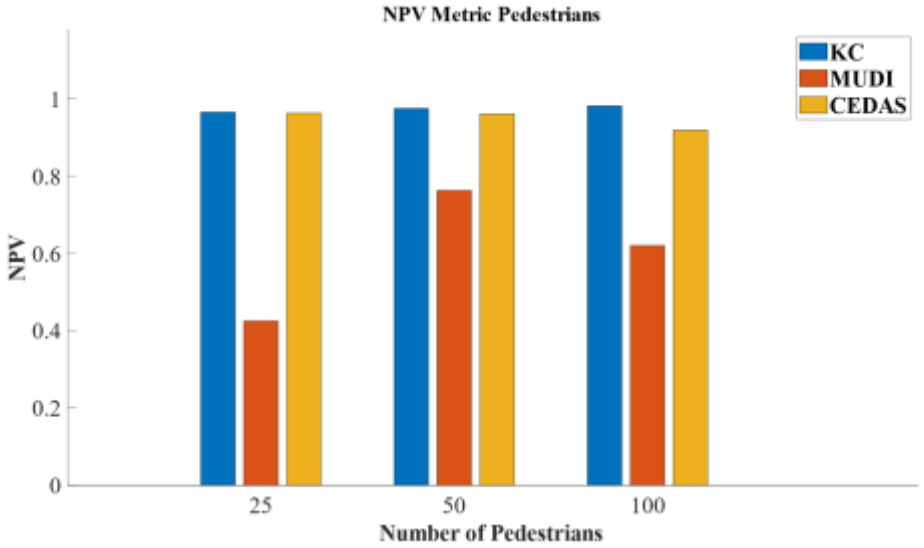

Figure A2.5. NPVs Metric Pedestrians.

Table A2.5. NPVs Metric Pedestrians.

KC	MUDI	CEDAS

0.965	0.424	0.962
0.974	0.762	0.960
0.981	0.619	0.918

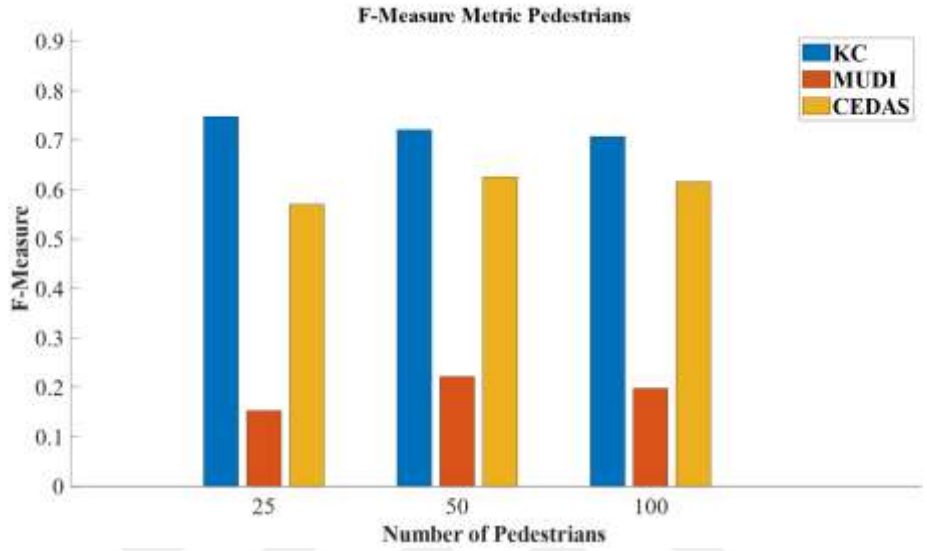

Figure A2.6. F-Measure Metric Pedestrians.

Table A2.6. F-Measure Metric Pedestrians.

KC	**MUDI**	**CEDAS**
0.747	0.152	0.569
0.720	0.221	0.625
0.706	0.197	0.615

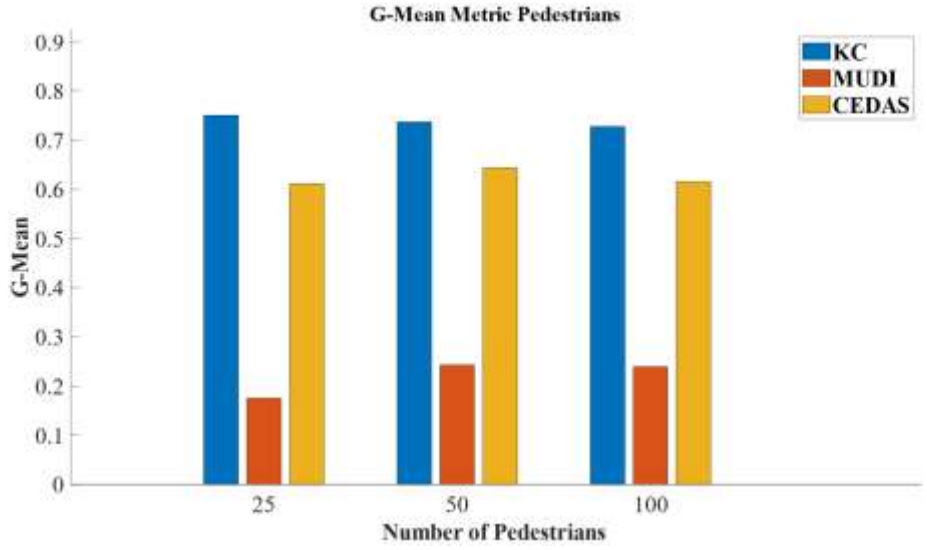

Figure A2.7. G-Mean Metric Pedestrians.

Table A2.7. G-Mean Metric Pedestrians.

KC	MUDI	CEDAS
0.750	0.175	0.611
0.736	0.242	0.643
0.727	0.239	0.615

APPENDIX A- 3 : Metric Speed for 1, 2 and 3 m\sec.

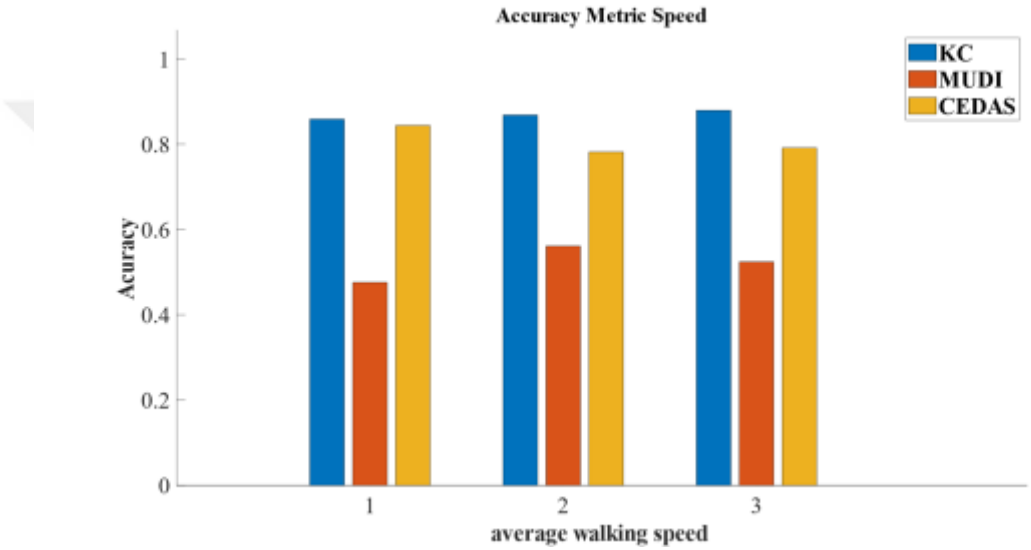

Figure A3.1. Accuracy Metric Speed.

Table A3.1. Accuracy Metric Speed.

KC	MUDI	CEDAS
0.858	0.475	0.843
0.868	0.560	0.781
0.879	0.524	0.791

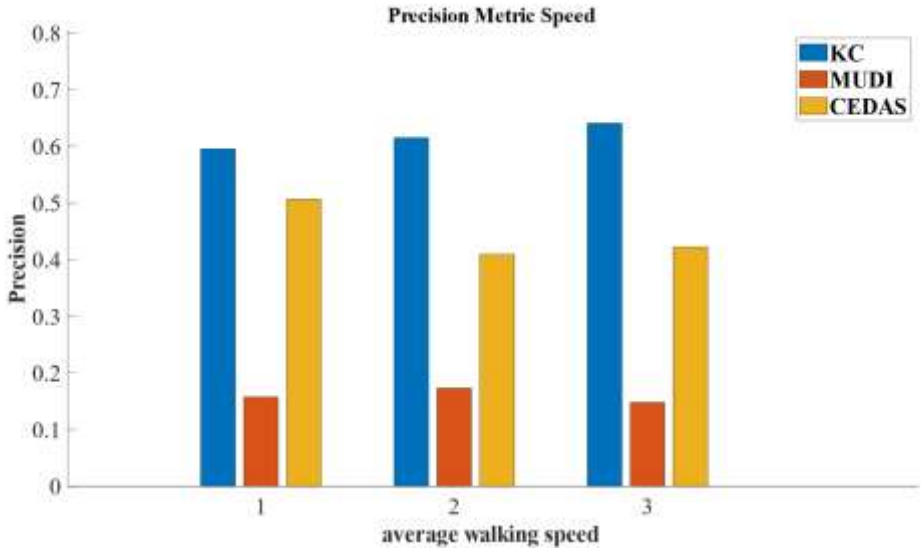

Figure A3.2. Precision Metric Speed.

Table A3.2. Precision Metric Speed.

KC	MUDI	CEDAS
0.595	0.157	0.506
0.614	0.172	0.408
0.640	0.147	0.422

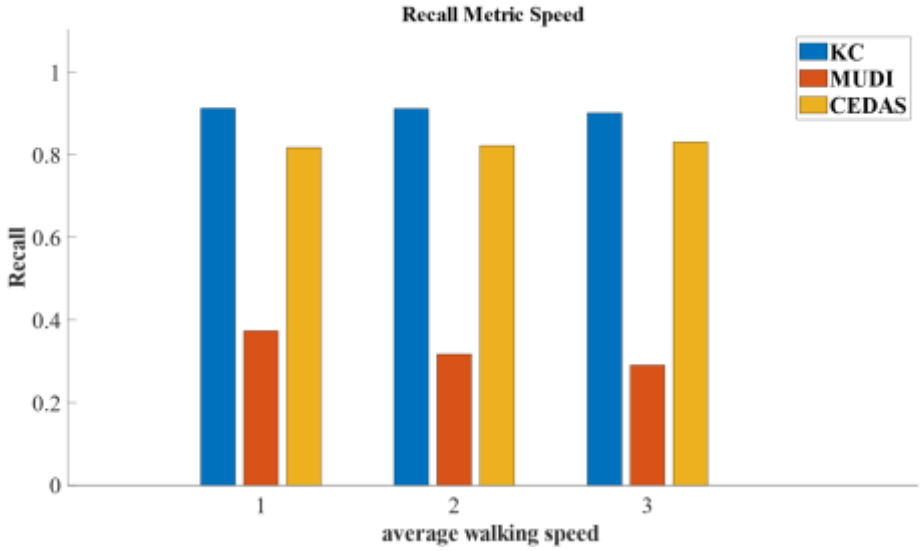

Figure A3.3. Recall Metric Speed.

Table A3.3. Recall Metric Speed.

KC	MUDI	CEDAS
0.912	0.373	0.817

0.911	0.317	0.821
0.900	0.290	0.830

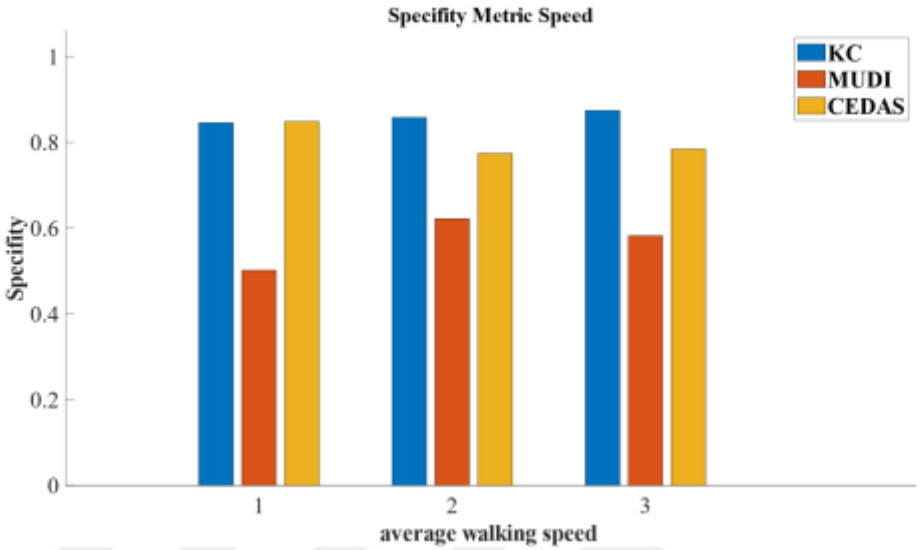

Figure A3.4. Specifity Metric Speed.

Table A3.4. Specifity Metric Speed.

KC	MUDI	CEDAS
0.845	0.5015	0.848
0.857	0.621	0.774
0.874	0.5823	0.783

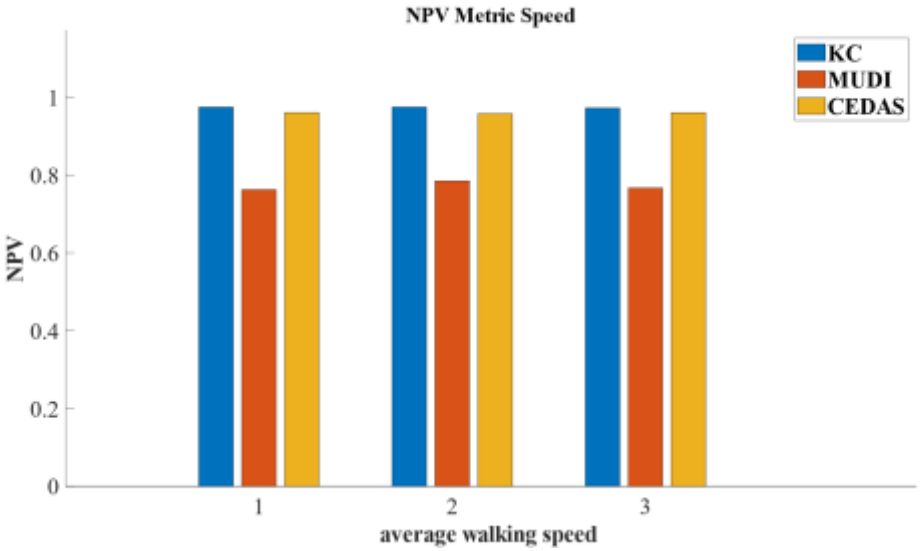

Figure A3.5. NPVs Metric Speed.

Table A3.5. NPVs Metric Speed.

KC	MUDI	CEDAS
0.974	0.762	0.960
0.974	0.784	0.958
0.972	0.766	0.960

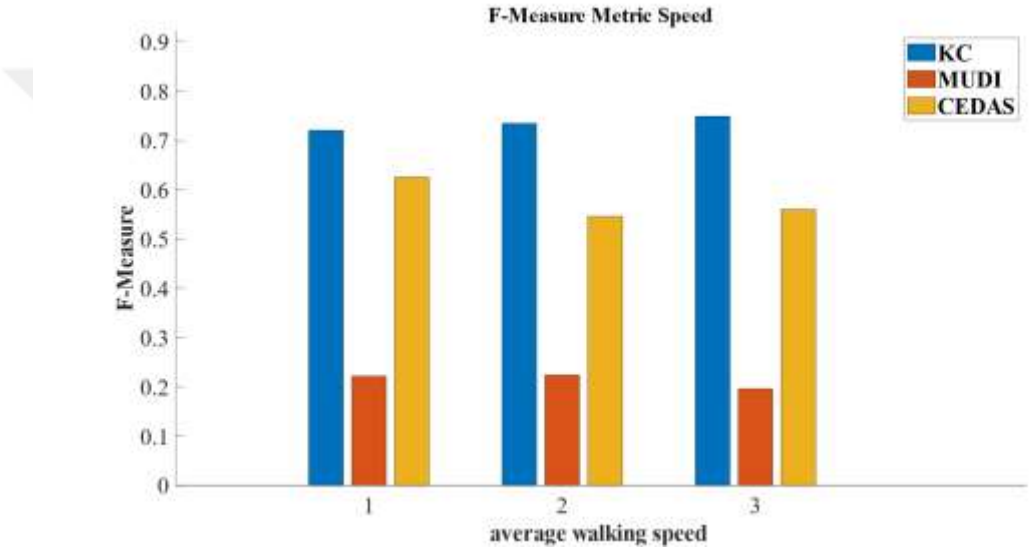

Figure A3.6. F-Measure Metric Speed.

Table A3.6. F-Measure Metric Speed.

KC	MUDI	CEDAS
0.720	0.221	0.625
0.734	0.223	0.545
0.748	0.195	0.559

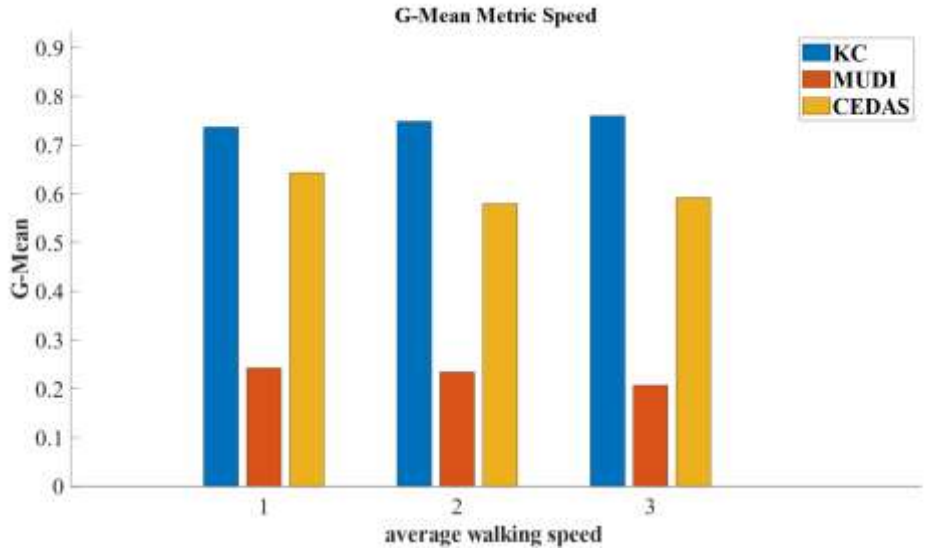

Figure A3.7. G-Mean Metric Speed.

Table A3.7. G-Mean Metric Speed.

KC	MUDI	CEDAS
0.736	0.242	0.643
0.748	0.234	0.579
0.759	0.207	0.592

APPENDIX B : IDS Result Diagrams And Tables.

APPENDIX B- 1 : IDS Metric Anomaly for 5%, 10% and 20%.

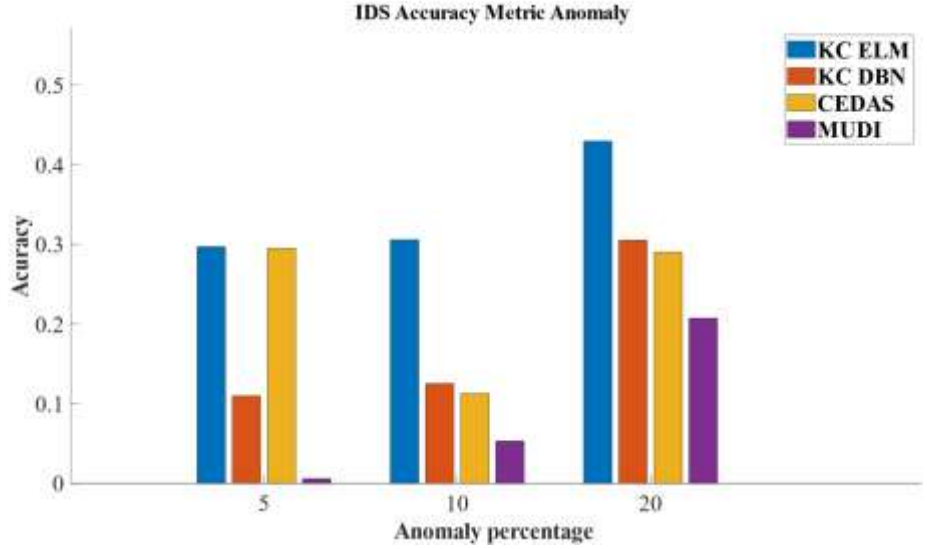

Figure B1.1. IDS Accuracy Metric Anomaly.

Table B1.1. IDS Accuracy Metric Anomaly.

KC ELM	KC DBN	MUDI	CEDAS
0.296	0.109	0.294	0.005
0.305	0.125	0.112	0.052
0.429	0.305	0.289	0.207

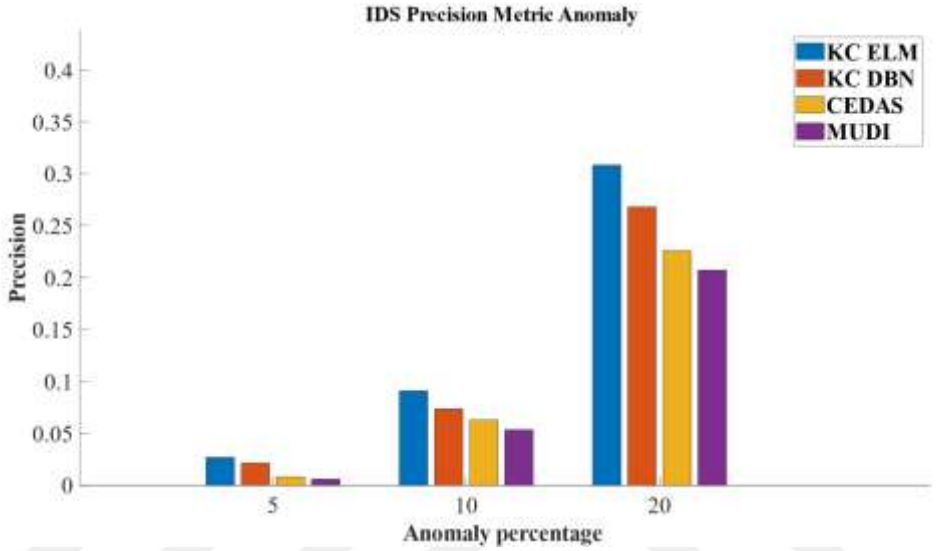

Figure B1.2. IDS Precision Metric Anomaly.

Table B1.2. IDS Precision Metric Anomaly.

KC ELM	KC DBN	MUDI	CEDAS
0.026	0.021	0.008	0.005
0.090	0.073	0.062	0.053
0.308	0.267	0.225	0.207

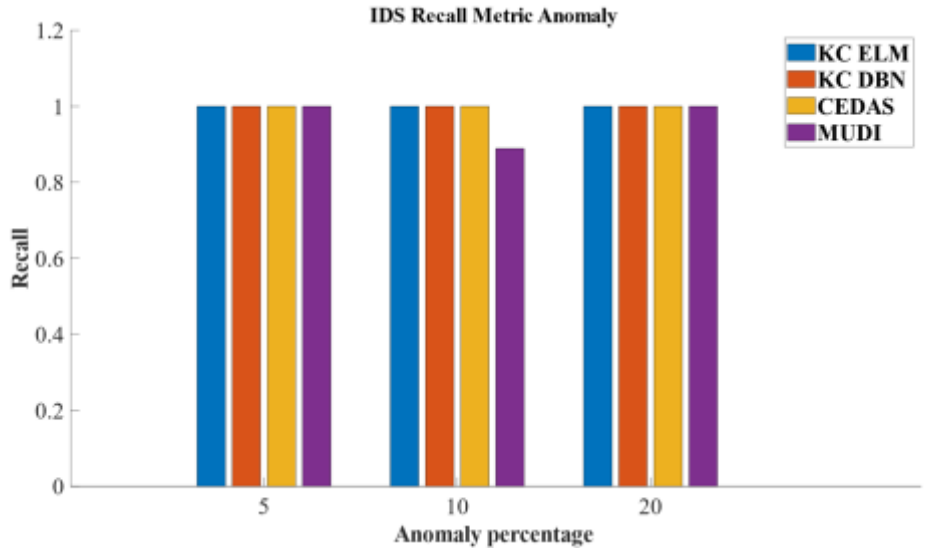

Figure B1.3. IDS Recall Metric Anomaly.

Table B1.3. IDS Recall Metric Anomaly.

KC ELM	KC DBN	MUDI	CEDAS
1.000	1.000	1.000	1.000
1.000	1.000	1.000	0.888
1.000	1.000	1.000	1.000

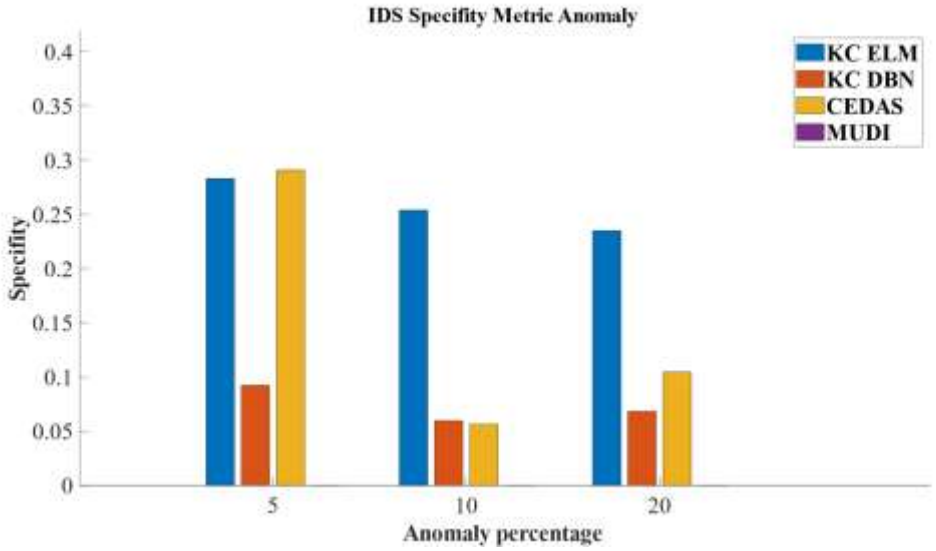

Figure B1.4. IDS Specifity Metric Anomaly.

Table B1.4. IDS Specifity Metric Anomaly.

KC ELM	KC DBN	MUDI	CEDAS
0.282	0.092	0.290	0
0.253	0.059	0.056	0
0.234	0.068	0.104	0

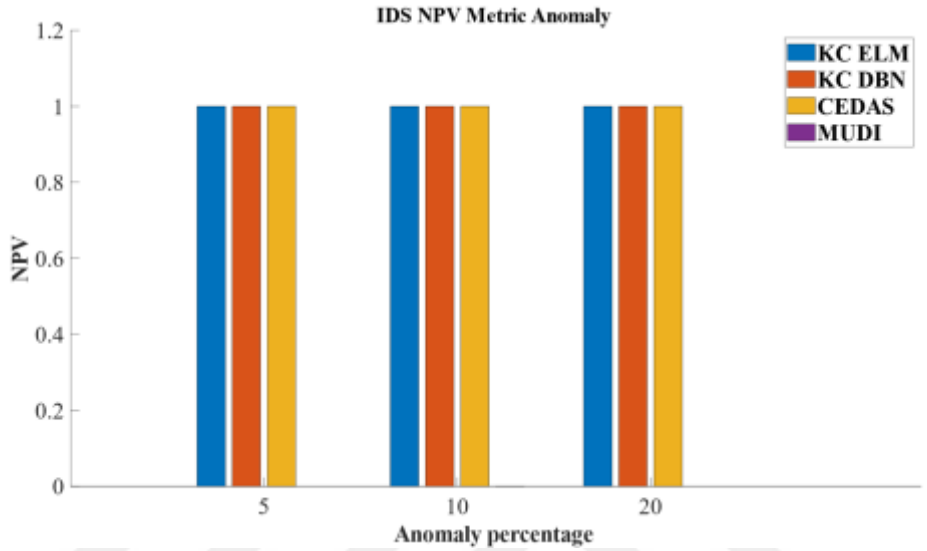

Figure B1.5. IDS NPVs Metric Anomaly.

Table B1.5. IDS NPV Metric Anomaly.

KC ELM	KC DBN	MUDI	CEDAS
1.000	1.000	1.000	-
1.000	1.000	1.000	0
1.000	1.000	1.000	-

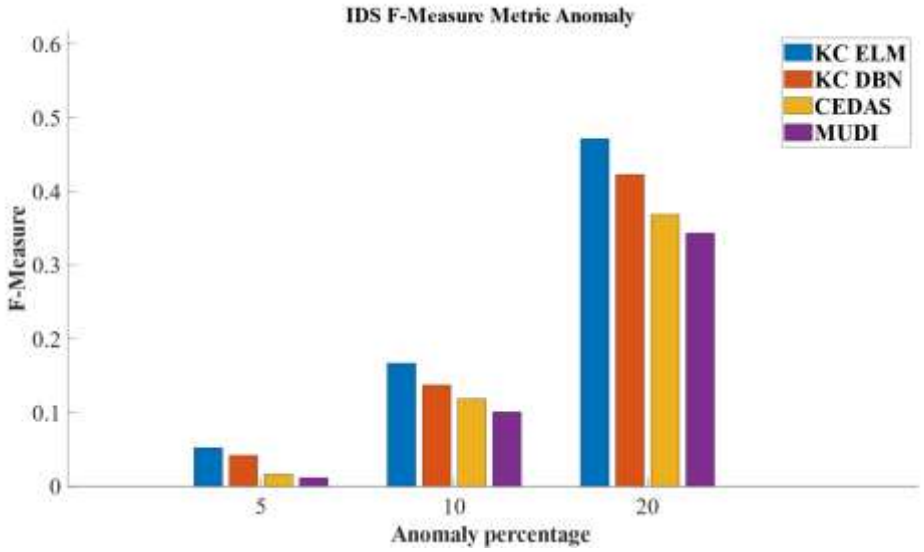

Figure B1.6. IDS F-Measure Metric Anomaly.

Table B1.6. IDS F-Measure Metric Anomaly.

KC ELM	KC DBN	MUDI	CSDAS
0.052	0.041	0.016	0.011
0.166	0.136	0.118	0.100
0.4712	0.422	0.368	0.343

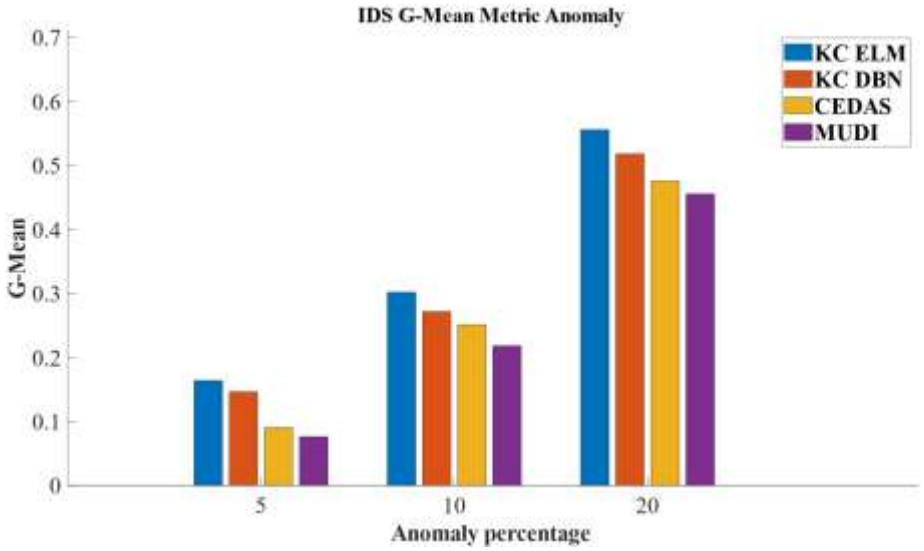

Figure B1.7. IDS G-Mean Metric Anomaly.

Table B1.7. IDS G-Mean Metric Anomaly.

KC ELM	KC DBN	MUDI	CEDAS
0.163	0.145	0.090	0.076
0.301	0.271	0.250	0.217
0.555	0.517	0.475	0.455

APPENDIX B- 2 : IDS Metric Pedestrians for 25, 50 and 100 pedesterians.

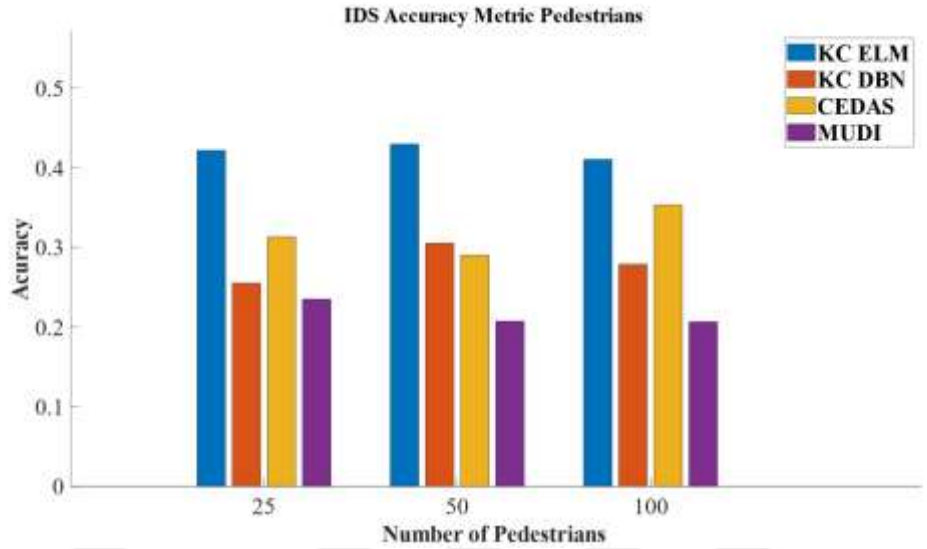

Figure B2.1. IDS Accuracy Metric Pedestrians.

Table B2.1. IDS Accuracy Metric Pedestrians.

KC ELM	KC DBN	MUDI	CEDAS
0.421	0.254	0.313	0.234
0.429	0.305	0.289	0.207
0.410	0.278	0.353	0.206

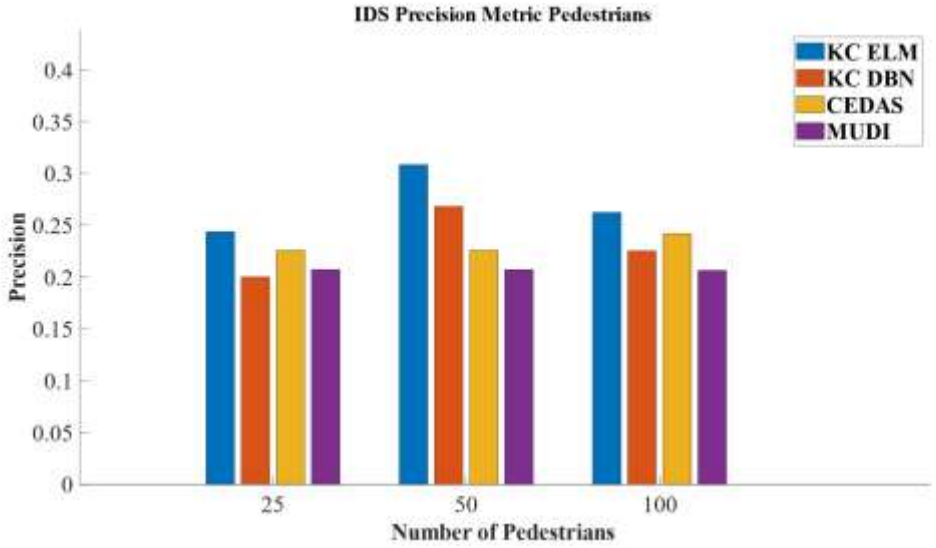

Figure B2.2. IDS Precision Metric Pedestrians.

Table B2.2. IDS Precision Metric Pedestrians.

KC ELM	KC DBN	MUDI	CEDAS
0.243	0.201	0.225	0.207
0.308	0.267	0.225	0.207
0.262	0.225	0.241	0.206

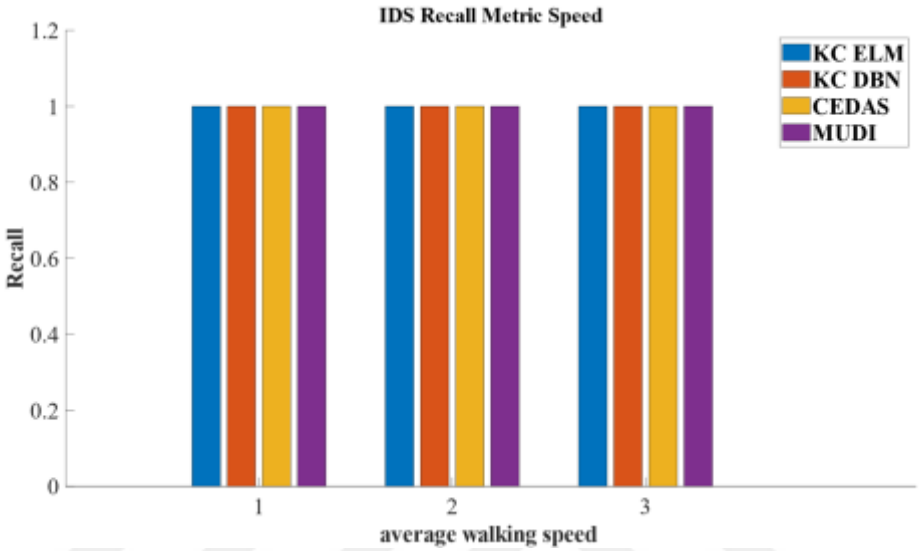

Figure B2.3. IDS Recall Metric Pedestrians.

Table B2.3. IDS Recall Metric Pedestrians.

KC ELM	KC DBN	MUDI	CEDAS
1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000

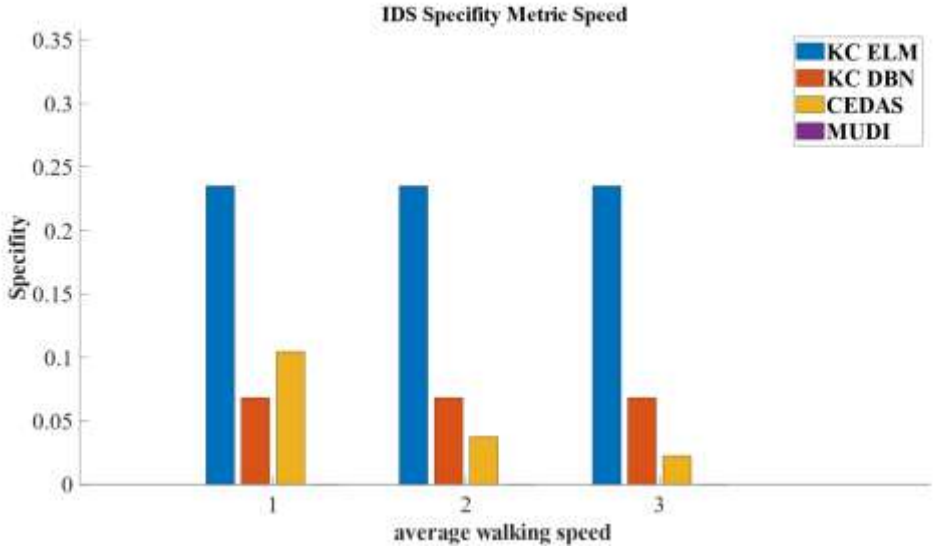

Figure B2.4. IDS Specifity Metric Pedestrians.

Table B2.4. IDS Specifity Metric Pedestrians.

KC ELM	KC DBN	MUDI	CEDAS
0.289	0.084	0.141	0.043
0.234	0.068	0.104	0
0.253	0.087	0.185	0

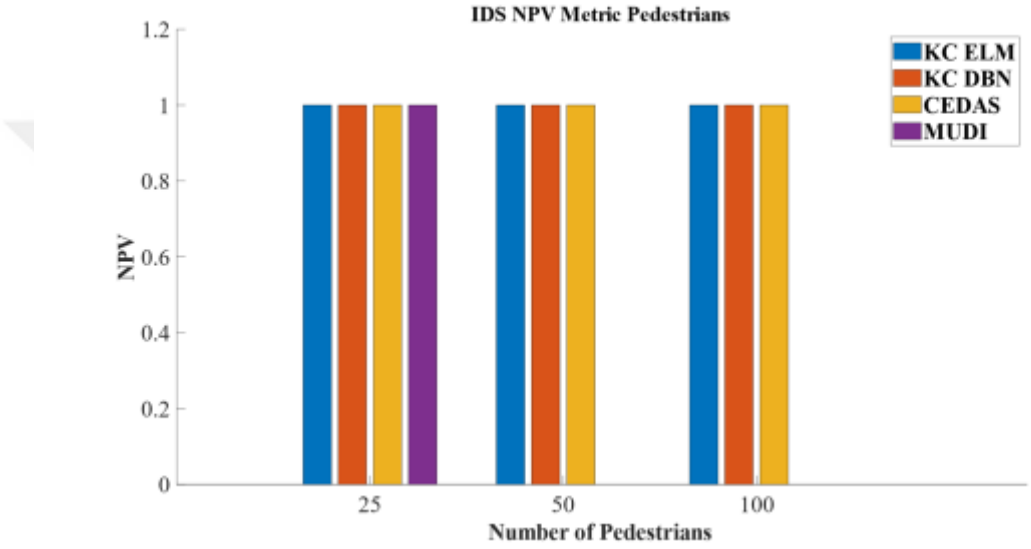

Figure B2.5. IDS NPVs Metric Pedestrians.

Table B2.5. IDS NPVs Metric Pedestrians.

KC ELM	KC DBN	MUDI	CEDAS
1.000	1.000	1.000	1.000
1.000	1.000	1.000	-
1.000	1.000	1.000	-

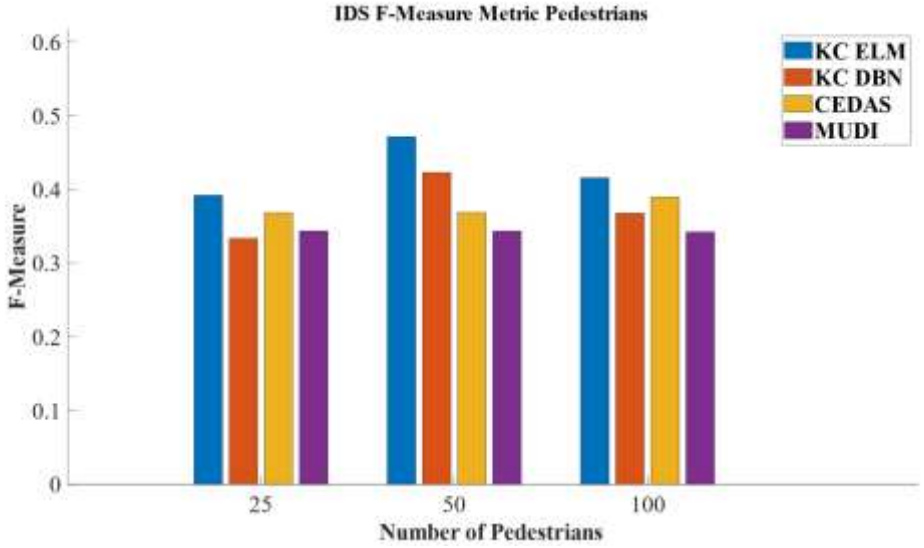

Figure B2.6. IDS F-Measure Metric Pedestrians.

Table B2.6. IDS F-Measure Metric Pedestrians.

KC ELM	KC DBN	MUDI	CEDAS
0.391	0.333	0.368	0.343
0.471	0.422	0.368	0.343
0.415	0.367	0.389	0.342

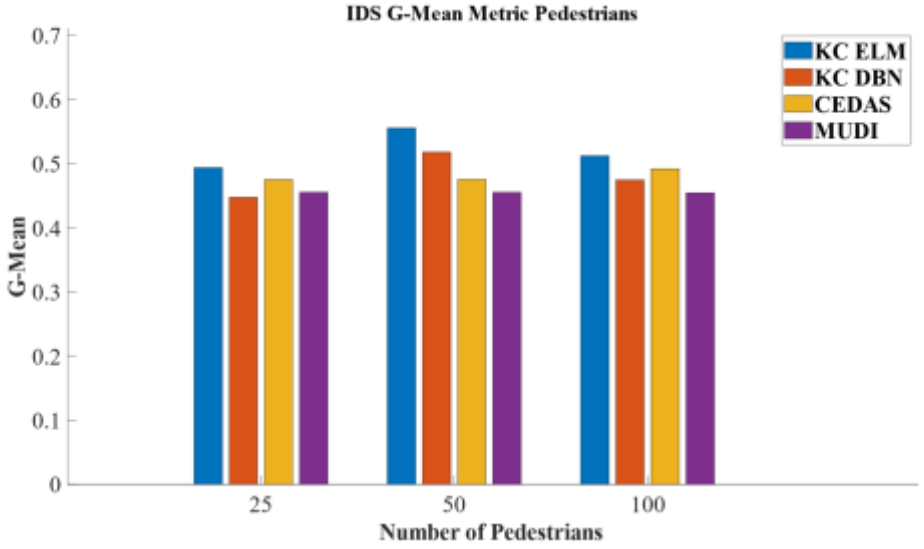

Figure B2.7. IDS G-Mean Metric Pedestrians.

Table B2.7. IDS G-Mean Metric Pedestrians.

KC ELM	KC DBN	MUDI	CEDAS
0.493	0.447	0.474	0.455
0.555	0.517	0.475	0.455
0.512	0.474	0.491	0.454

APPENDIX B- 3 : IDS Metric Speed for 1,2 and 3 m\sec.

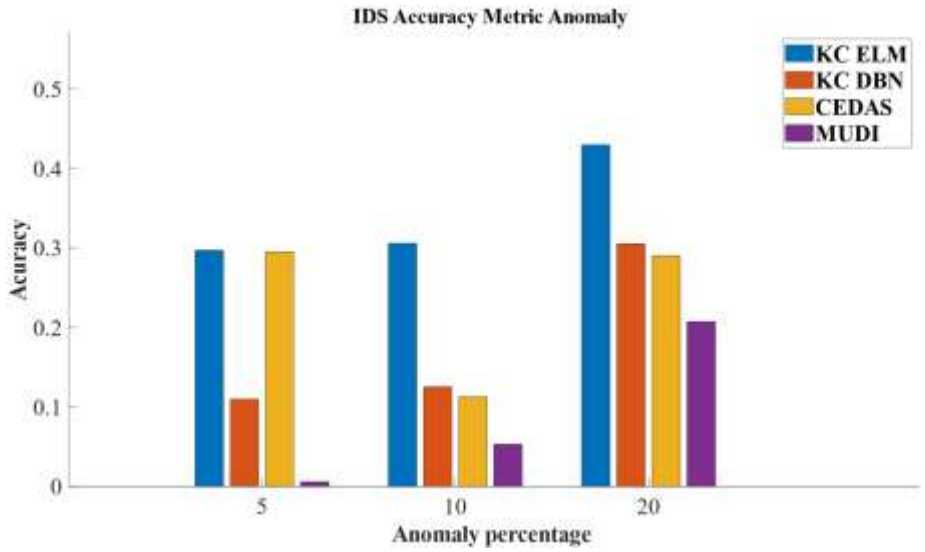

Figure B3.1. IDS Accuracy Metric Speed.

Table B3.1. IDS Accuracy Metric Speed.

KC ELM	KC DBN	MUDI	CEDAS
0.429	0.305	0.289	0.207
0.429	0.305	0.236	0.207
0.429	0.305	0.224	0.207

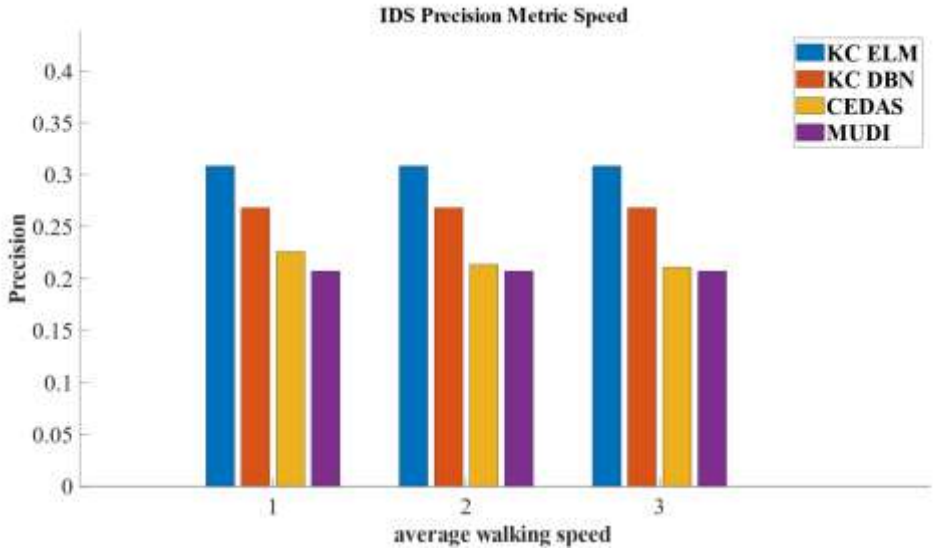

Figure B3.2. IDS Precision Metric Speed.

Table B3.2. IDS Precision Metric Speed.

KC ELM	KC DBN	MUDI	CEDAS
0.308	0.267	0.225	0.207
0.308	0.267	0.213	0.207
0.308	0.267	0.210	0.207

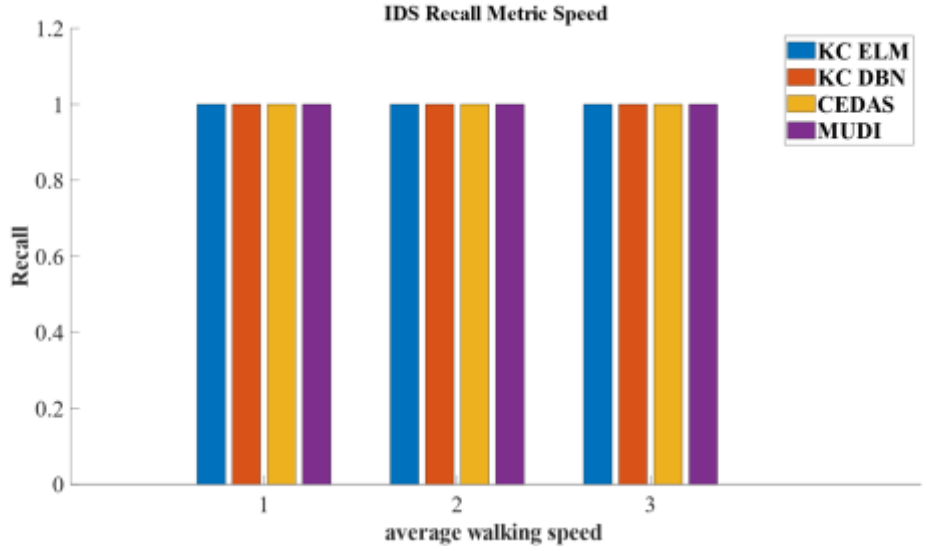

Figure B3.3. IDS Recall Metric Speed.

Table B3.3. IDS Recall Metric Speed.

KC ELM	KC DBN	MUDI	CEDAS
1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000

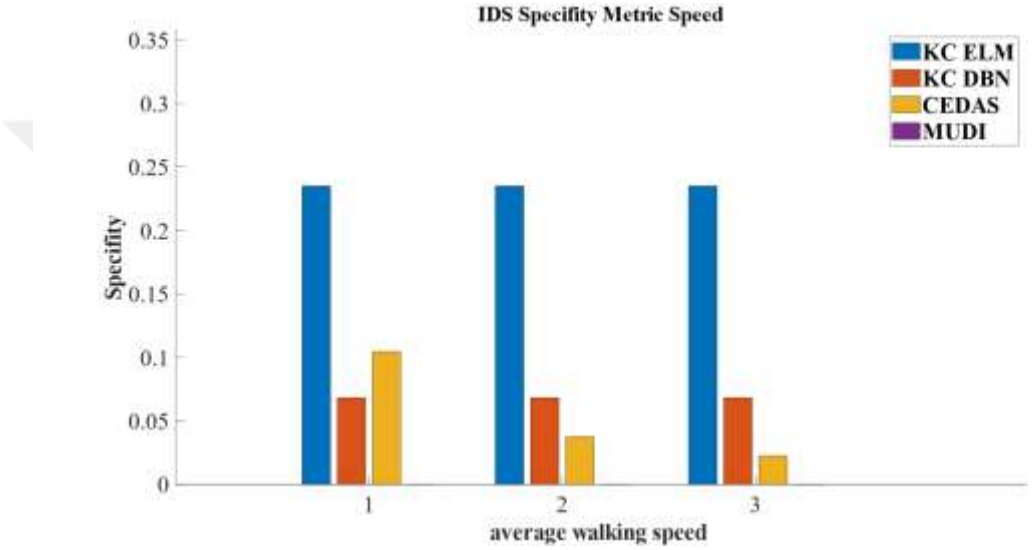

Figure B3.4. IDS Specifity Metric Speed.

Table B3.4. IDS Specifity Metric Speed.

KC ELM	KC DBN	MUDI	CEDAS
0.234	0.068	0.104	0
0.234	0.068	0.037	0
0.234	0.068	0.022	0

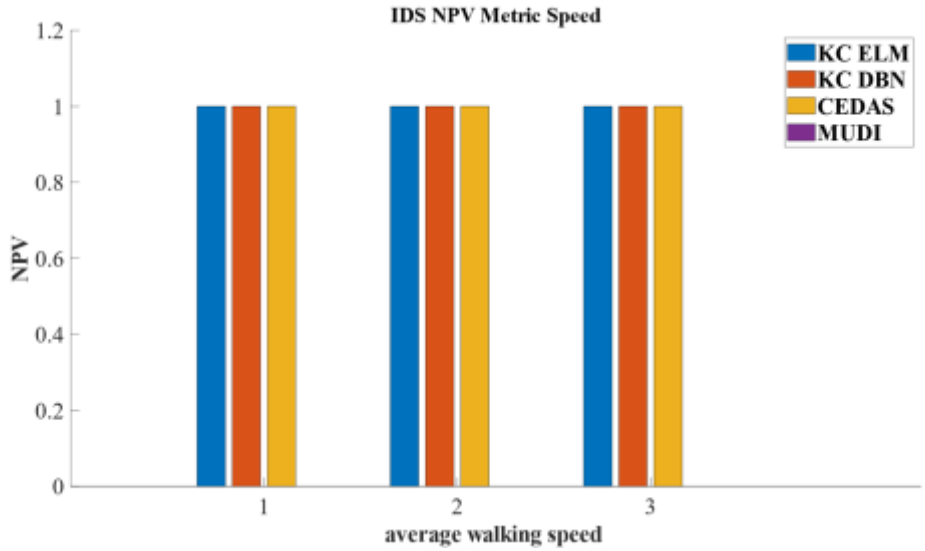

Figure B3.5. IDS NPVs Metric Speed.

Table B3.5. IDS NPVs Metric Speed.

KC ELM	KC DBN	MUDI	CEDAS
1.000	1.000	1.000	-
1.000	1.000	1.000	-
1.000	1.000	1.000	-

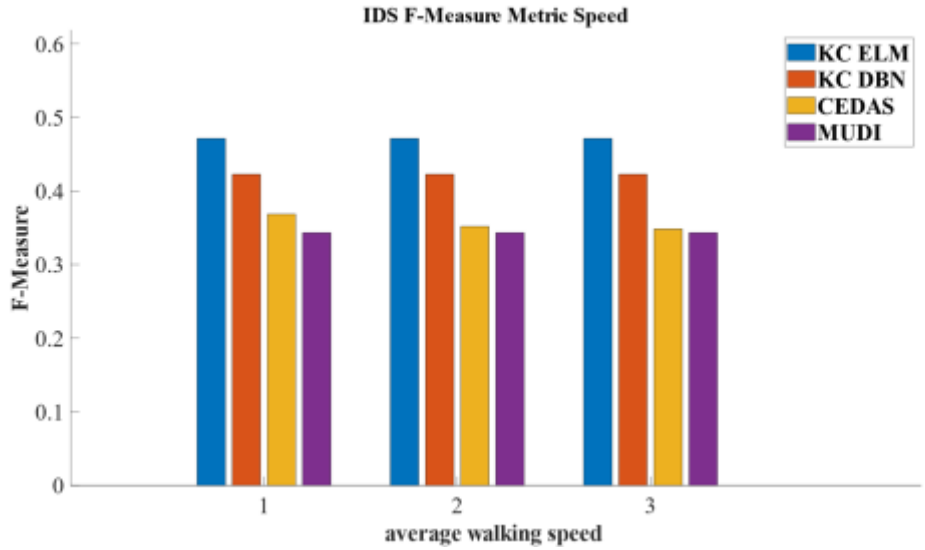

Figure B3.6. IDS F-Measure Metric Speed.

Table B3.6. IDS F-Measure Metric Speed.

KC ELM	KC DBN	MUDI	CEDAS
0.471	0.422	0.368	0.343
0.471	0.422	0.351	0.343
0.471	0.422	0.348	0.343

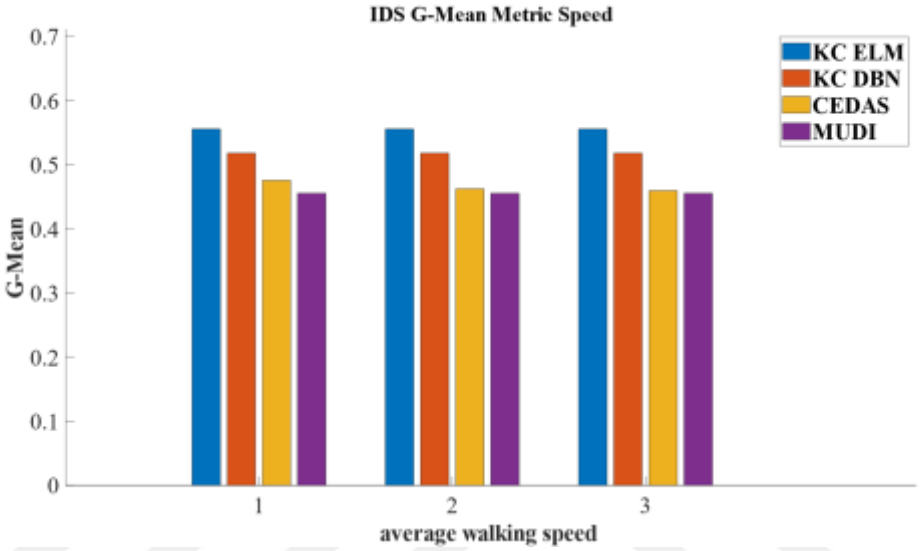

Figure B3.7. IDS G-Mean Metric Speed.

Table B3.7. IDS G-Mean Metric Speed.

KC ELM	KC DBN	MUDI	CEDAS
0.555	0.517	0.4751	0.455
0.555	0.517	0.461	0.455
0.555	0.517	0.459	0.455

APPENDIX C : E2E Delay & PDR for Anomaly Detection

APPENDIX C- 1 : E2E Delay & PDR Metric Pedestrians for 25, 50 and 100 pedesterians.

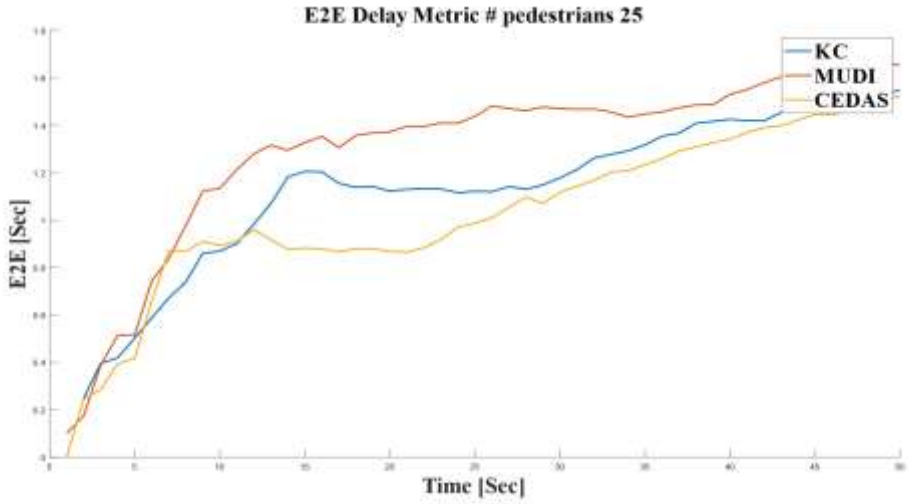

Figure C1.1. E2E Delay Metric Pedestrians 25 Plot.

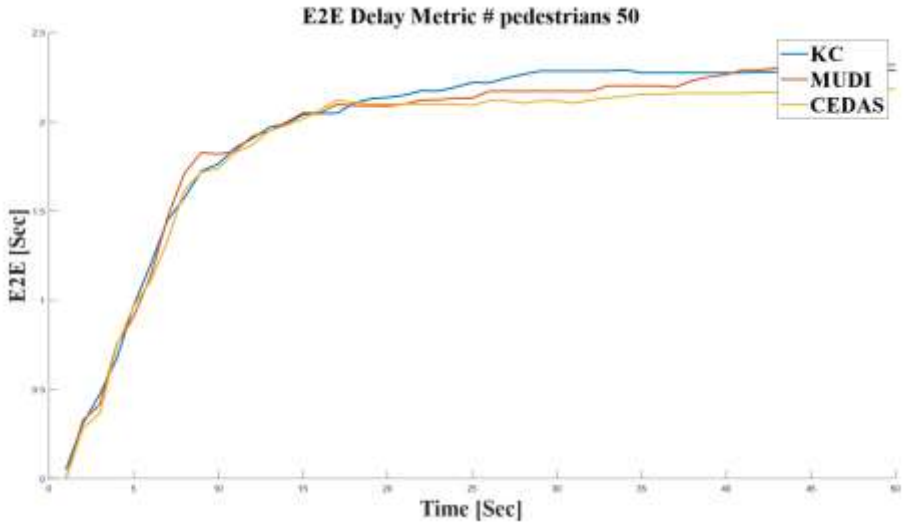

Figure C1.2. E2E Delay Metric Pedestrians 50 Plot.

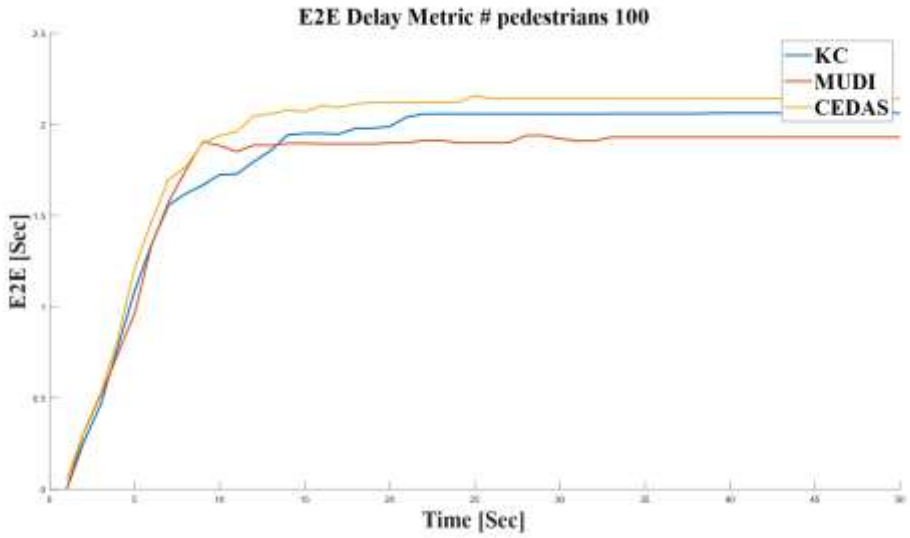

Figure C1.3. E2E Delay Metric Pedestrians 100 Plot.

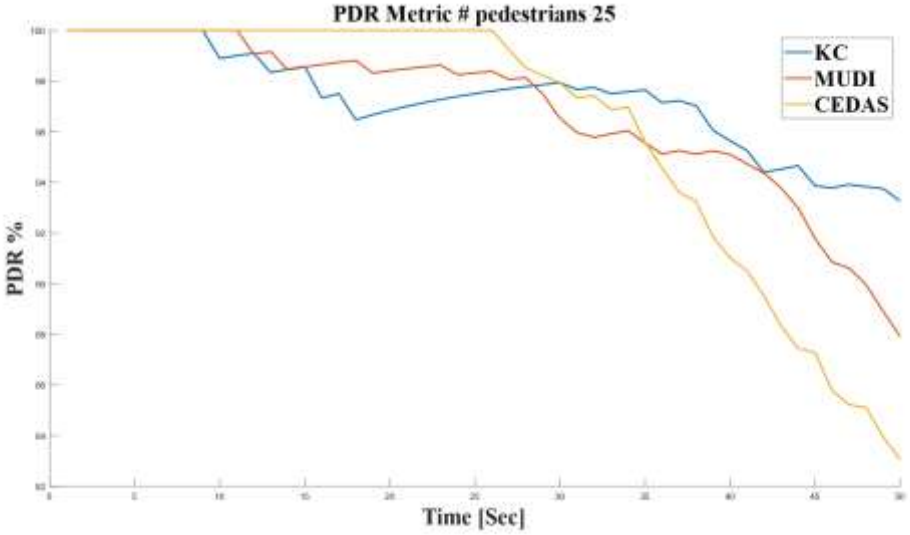

Figure C1.4. PDR Metric Pedestrians 25 Plot.

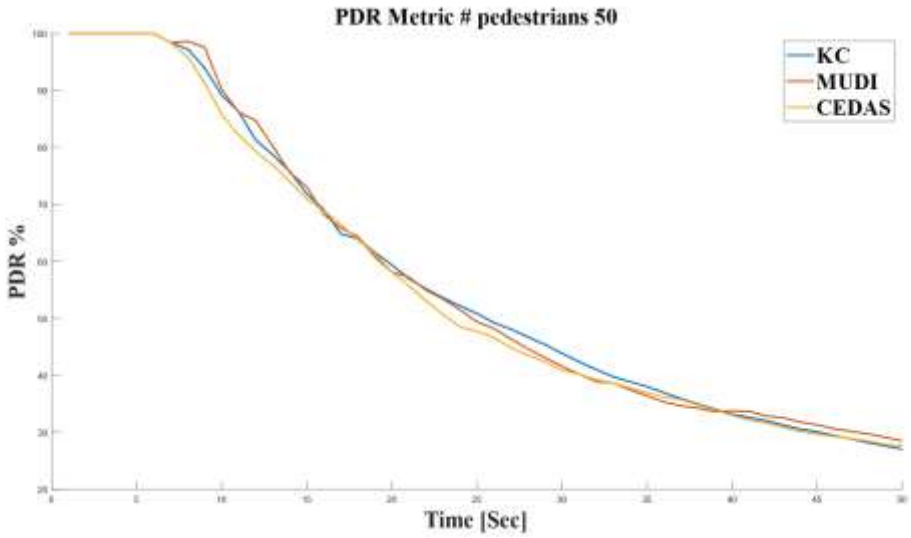

Figure C1.5. PDR Metric Pedestrians 50 Plot.

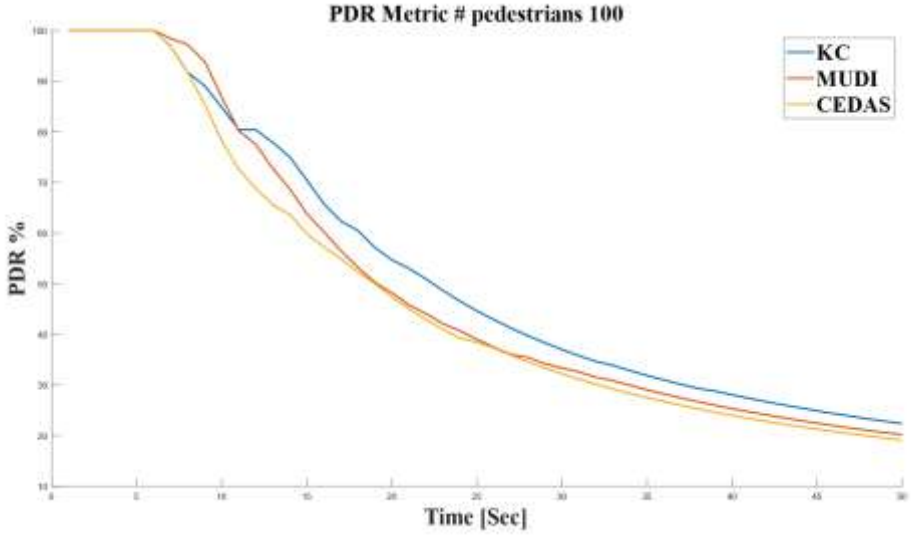

Figure C1.6. PDR Metric Pedestrians 100 Plot.

APPENDIX C- 2 : E2E Delay & PDR Metric Anomaly for 5%, 10% and 20%.

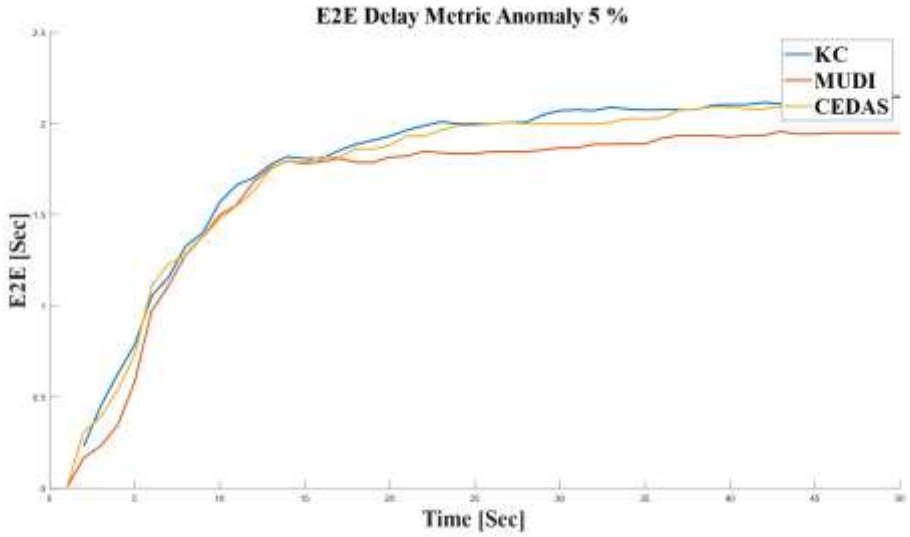

Figure C2.1. E2E Delay Metric Anomaly 5 % Plot.

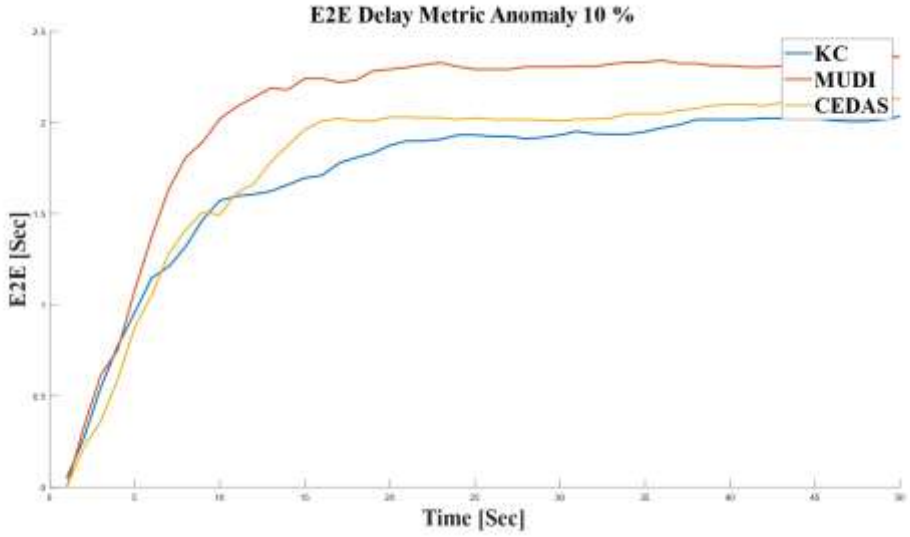

Figure C2.2. E2E Delay Metric Anomaly 10 % Plot.

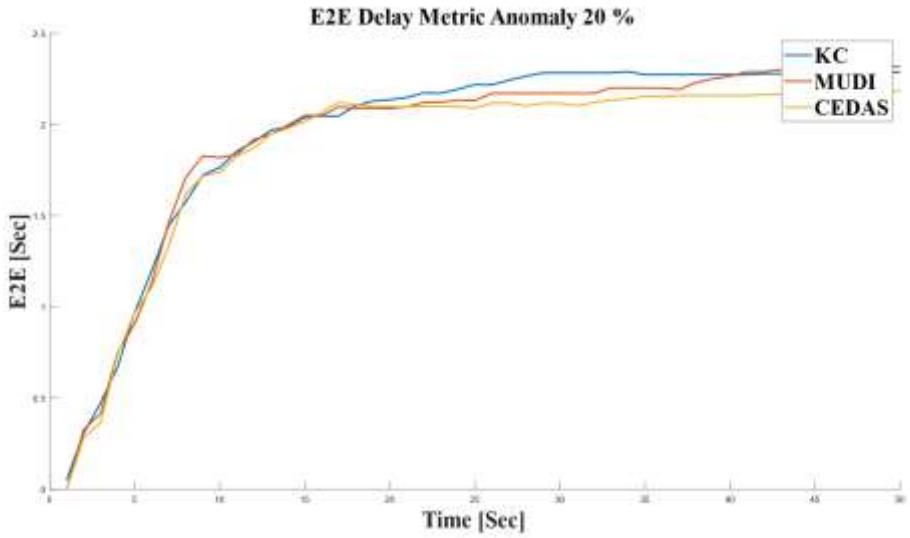

Figure C2.3. E2E Delay Metric Anomaly 20 % Plot.

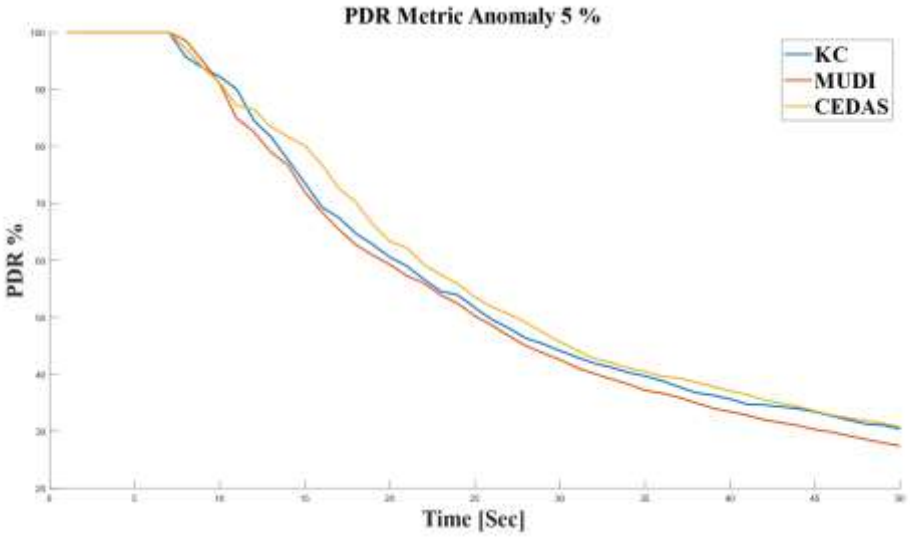

Figure C2.4. PDR Metric Anomaly 5 % Plot.

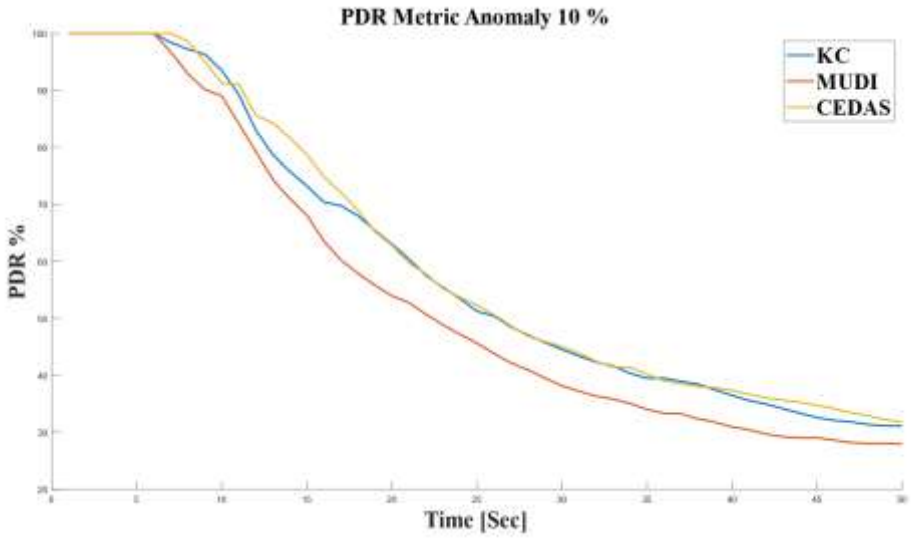

Figure C2.5. PDR Metric Anomaly 10 % Plot.

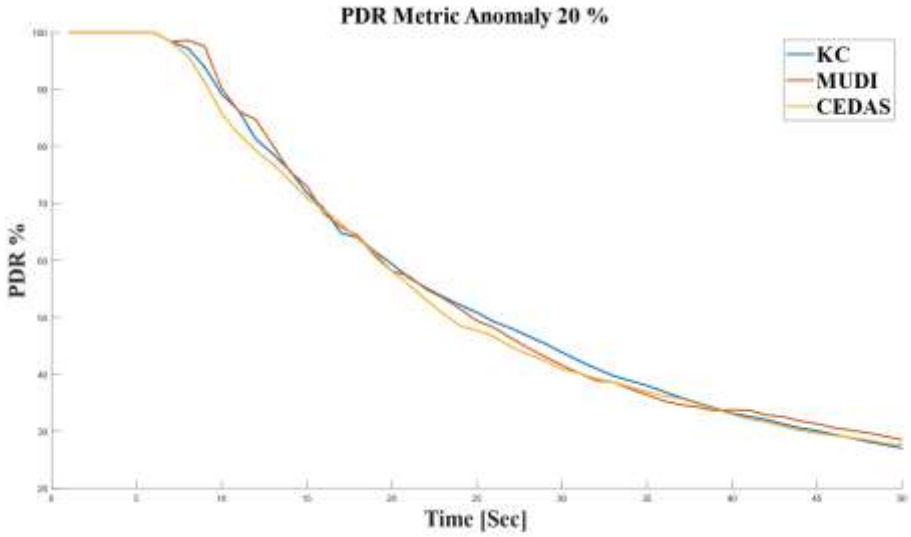

Figure C2.6. PDR Metric Anomaly 20 % Plot.

APPENDIX C- 3 : E2E Delay & PDR Metric speed for 1, 2 and 3m\ces.

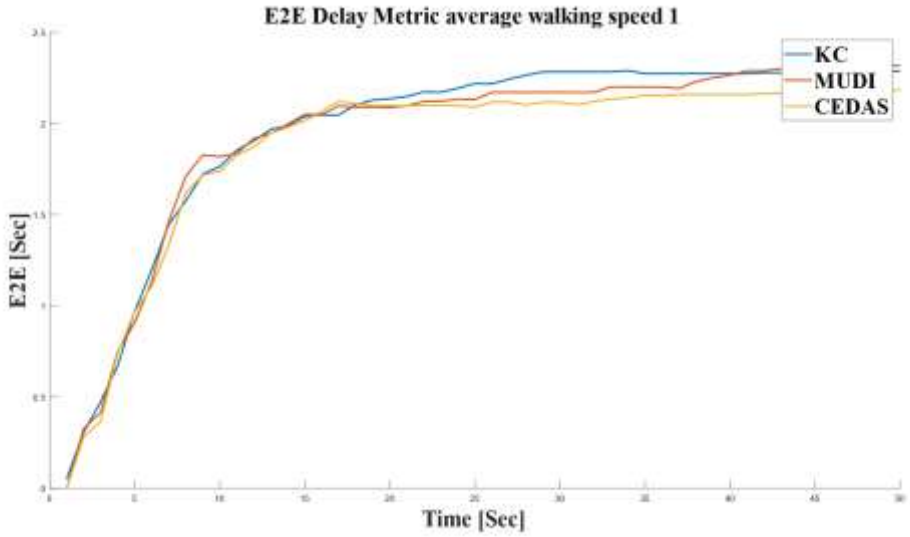

Figure C3.1. E2E Delay Metric average walking speed 1 Plot.

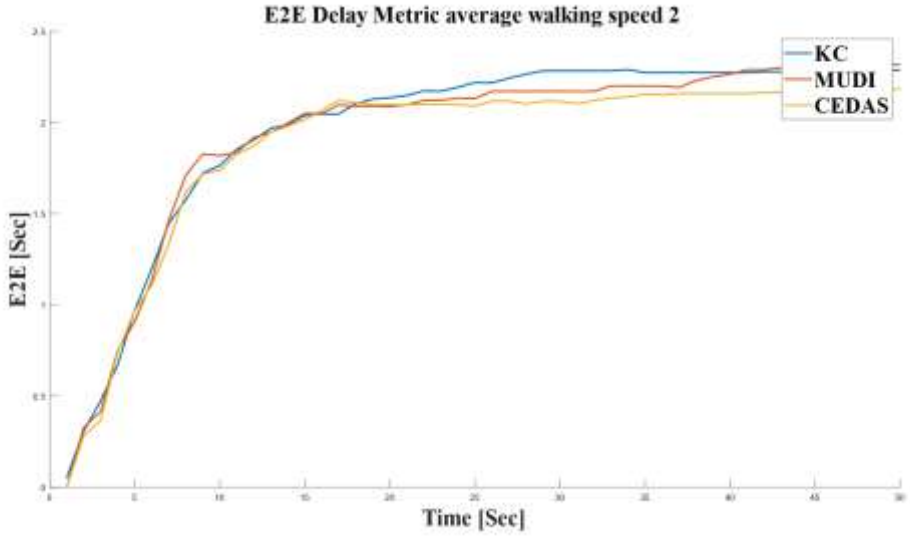

Figure C3.2. E2E Delay Metric average walking speed 2 Plot.

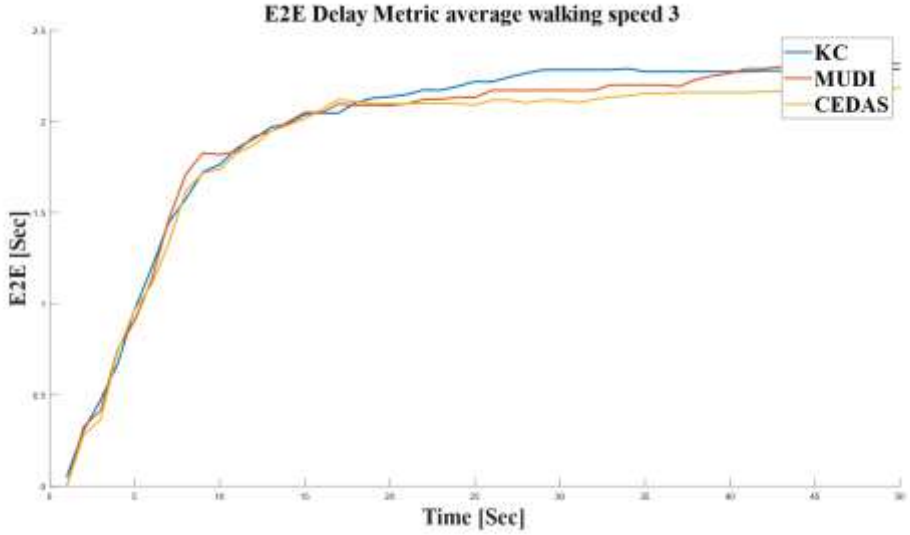

Figure C3.3. E2E Delay Metric average walking speed 3 Plot.

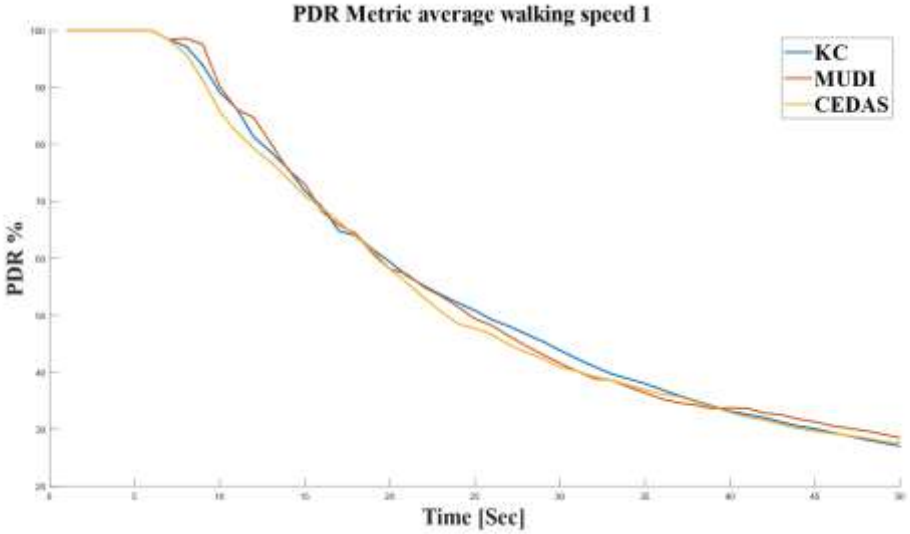

Figure C3.4. PDR Metric average walking speed 1 Plot.

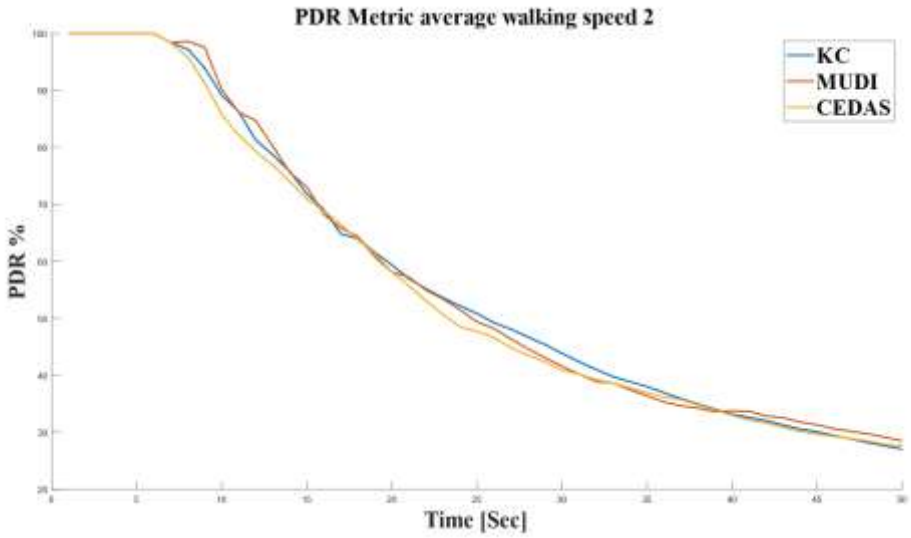

Figure C3.5. PDR Metric average walking speed 2 Plot.

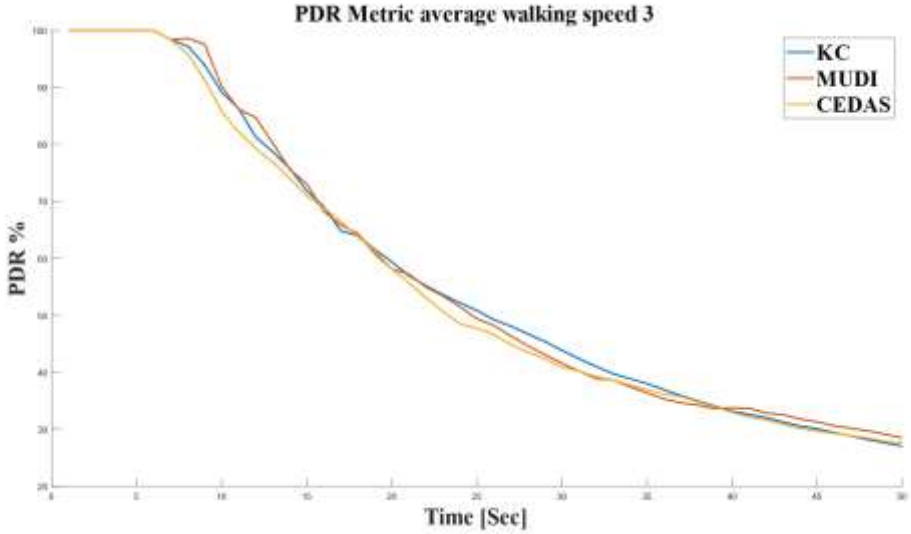

Figure C3.6. PDR Metric average walking speed 3 Plot.

APPENDIX D : E2E Delay & PDR for IDS

APPENDIX D- 1 : IDS E2E Delay & IDS PDR Metric pedestrians for 25,50 and 100 pedesterians.

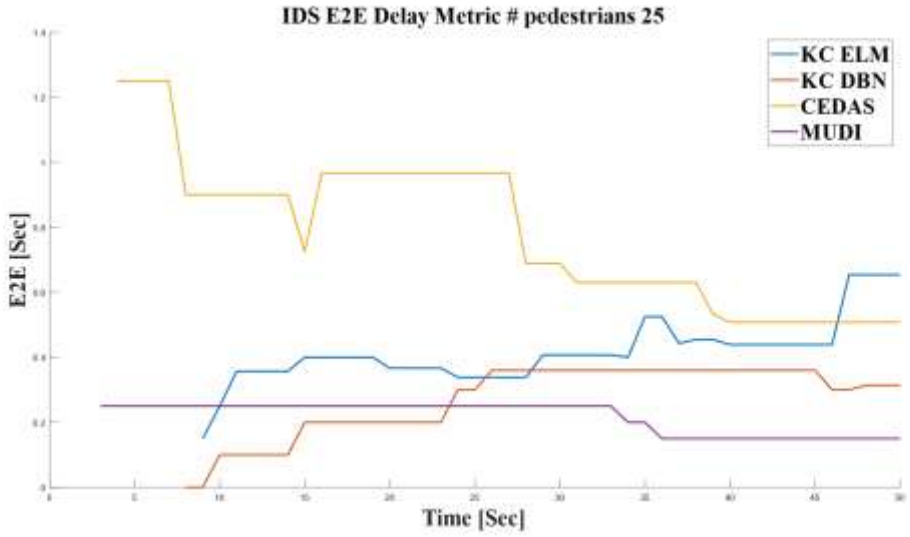

Figure D1.1. IDS E2E Delay Metric pedestrians 25 Plot.

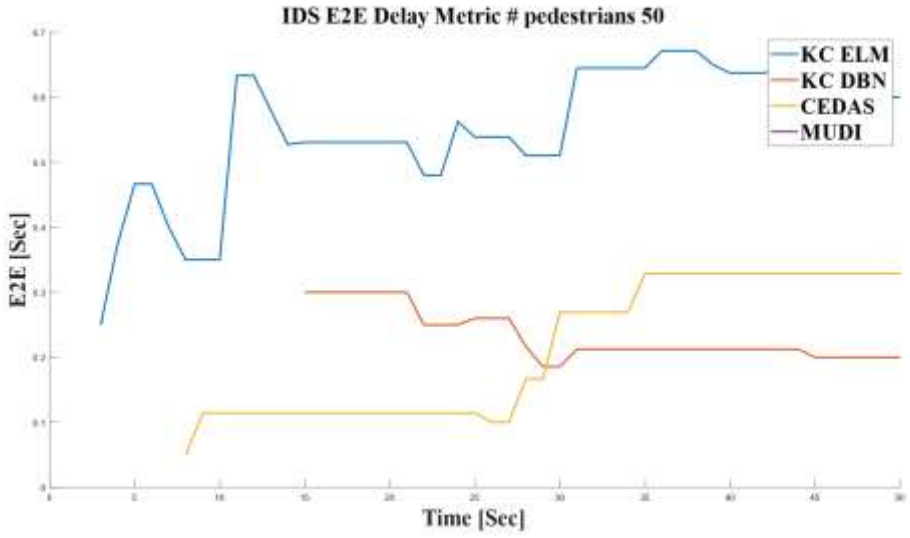

Figure D1.2. IDS E2E Delay Metric Pedestrians 50 Plot.

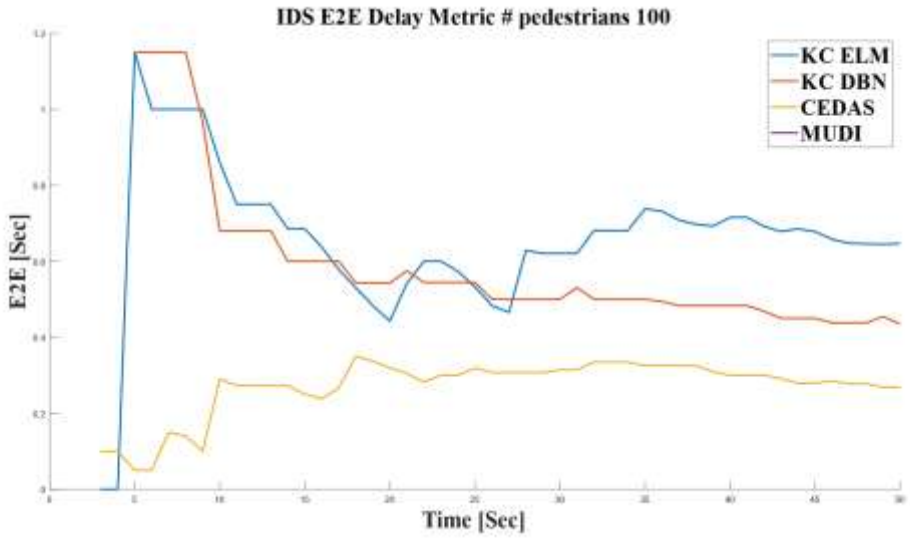

Figure D1.3. IDS E2E Delay Metric Pedestrians 100 Plot.

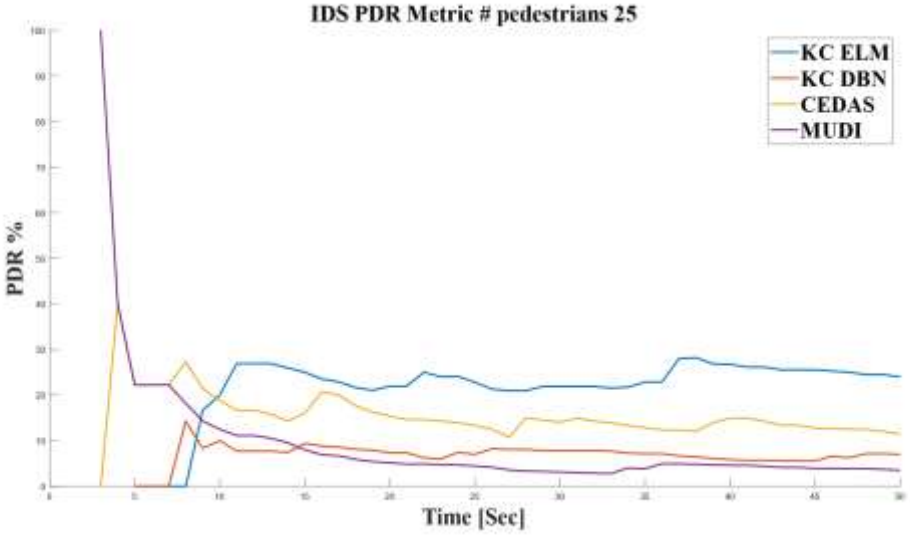

Figure D1.4. IDS PDR Metric Pedestrians 25 Plot.

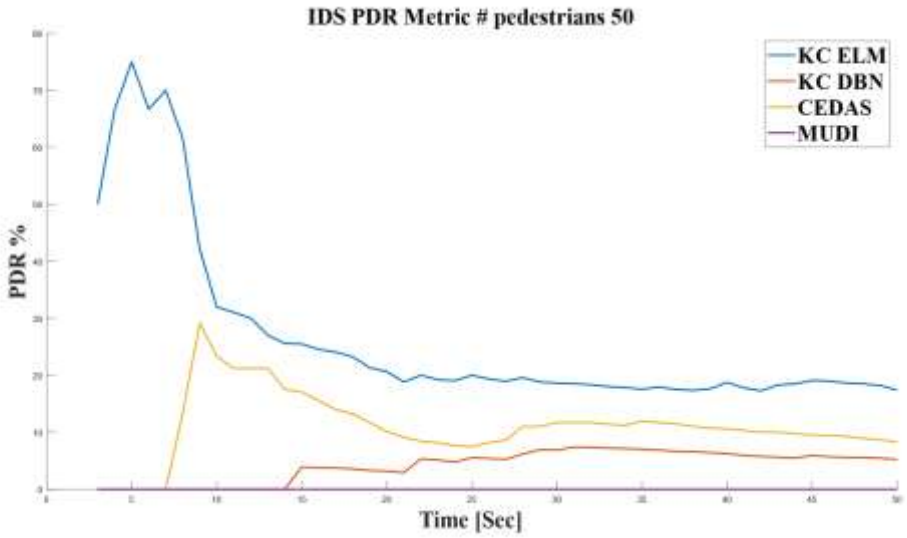

Figure D1.5. IDS PDR Metric Pedestrians 50 Plot.

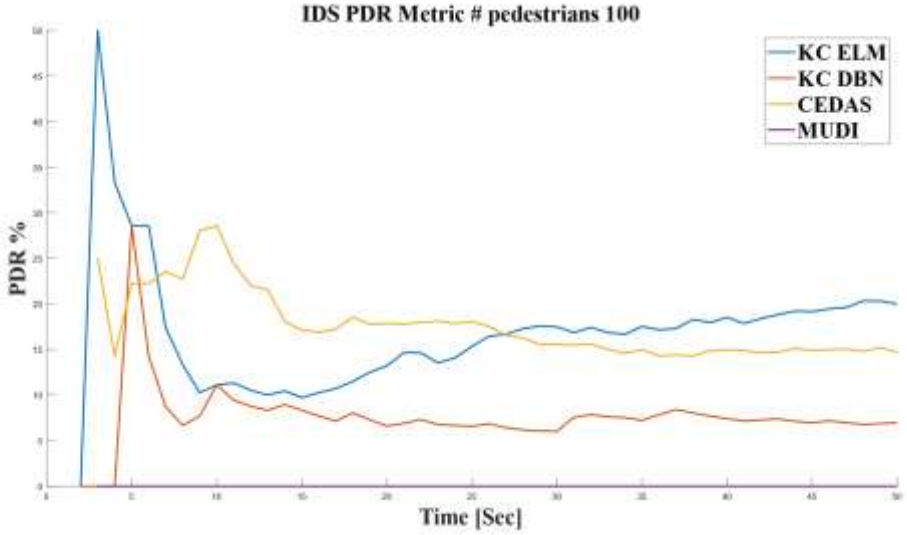

Figure D1.6. IDS PDR Metric Pedestrians 100 Plot.

APPENDIX D- 2 : IDS E2E Delay &IDS PDR Metric Anomaly for 5%, 10% and 20%.

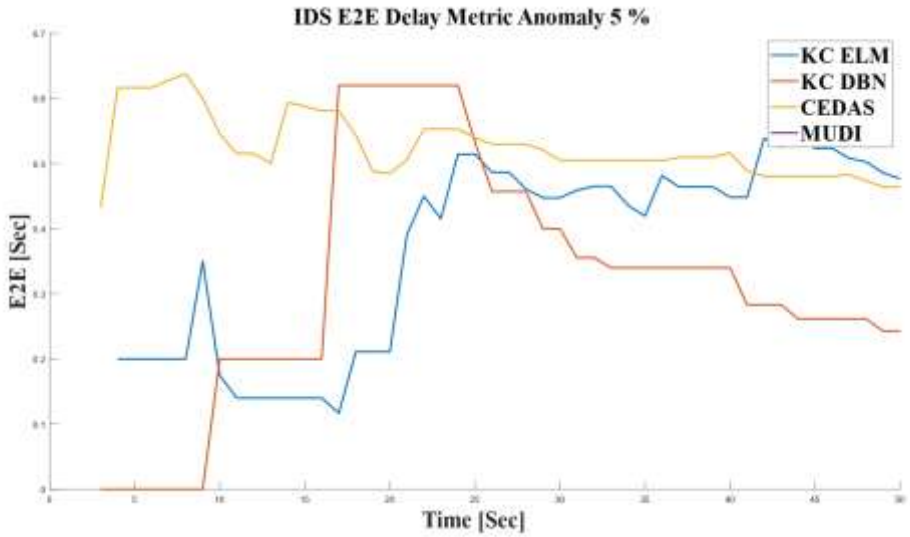

Figure D2.1. IDS E2E Delay Metric Anomaly 5 % Plot.

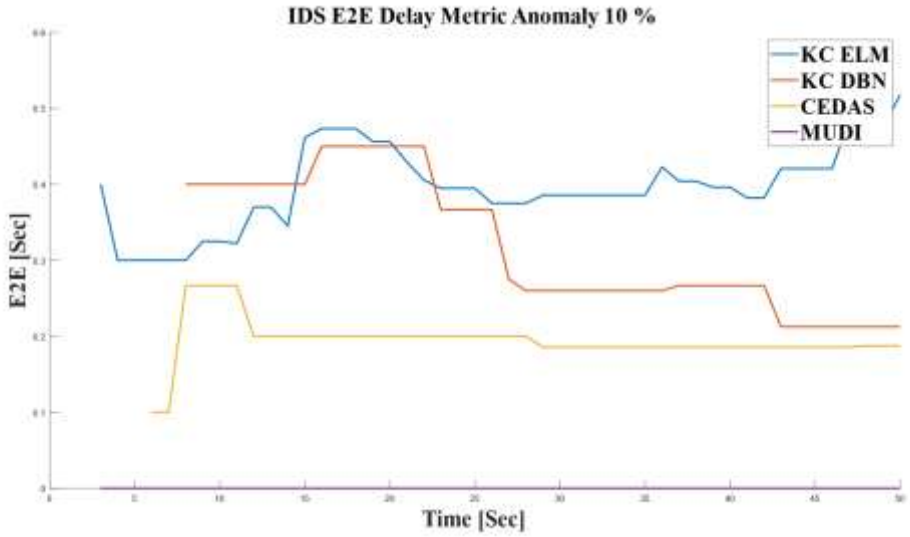

Figure D2.2. IDS E2E Delay Metric Anomaly 10 % Plot.

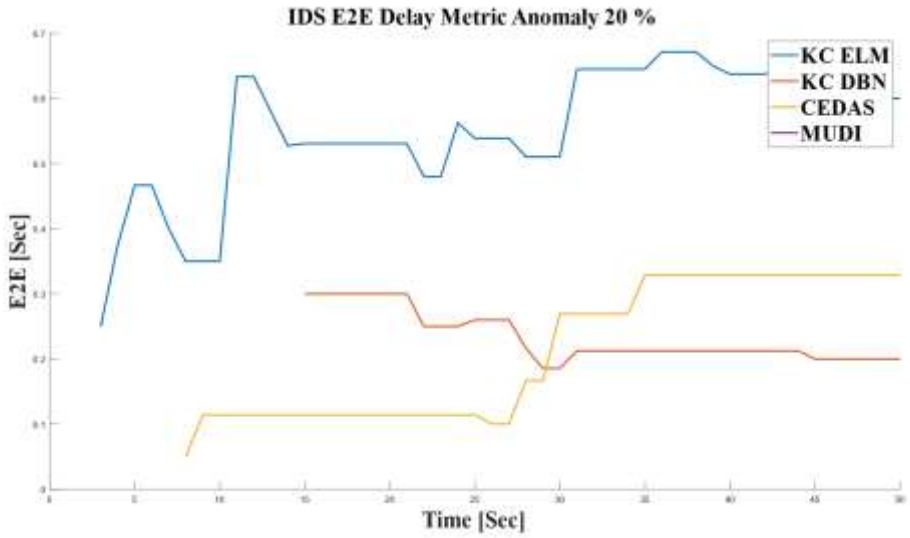

Figure D2.3. IDS E2E Delay Metric Anomaly 20 % Plot.

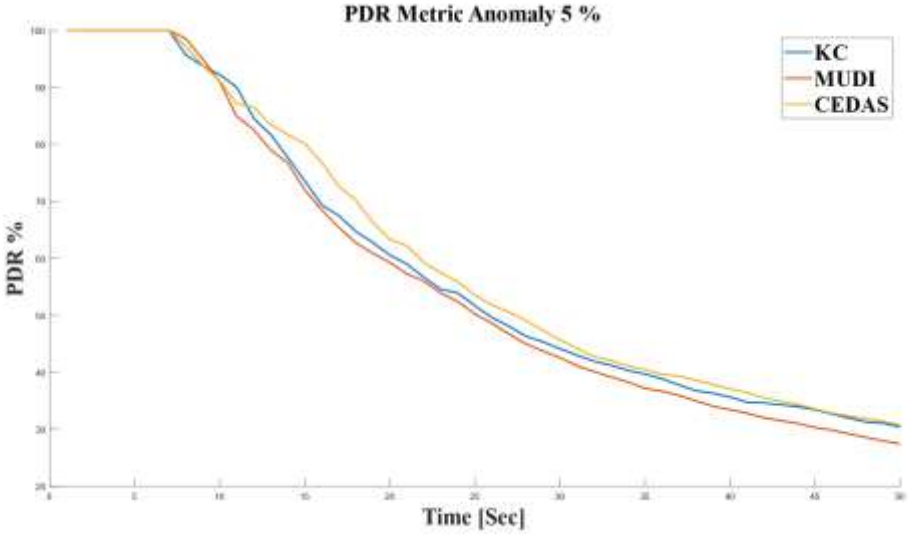

Figure D2.4. PDR Anomaly 5%.

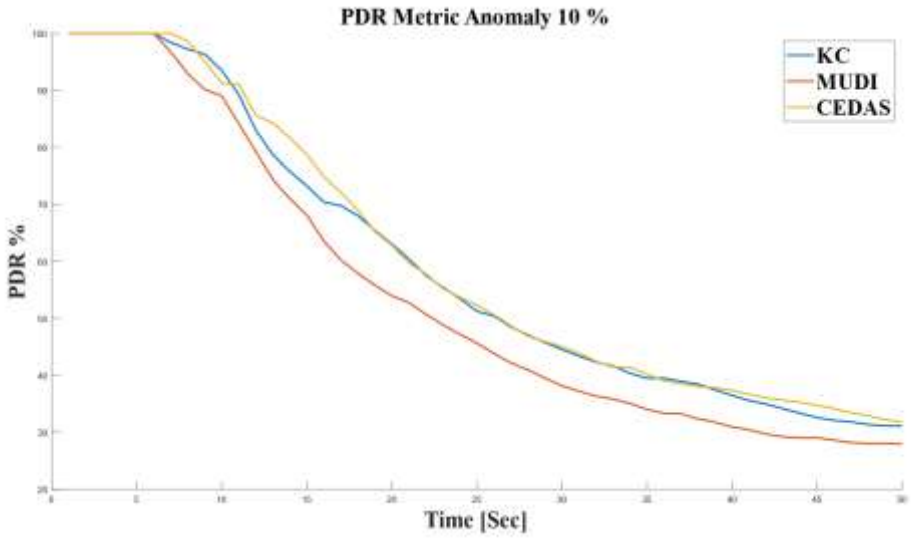

Figure D2.5. PDR Anomaly 10%.

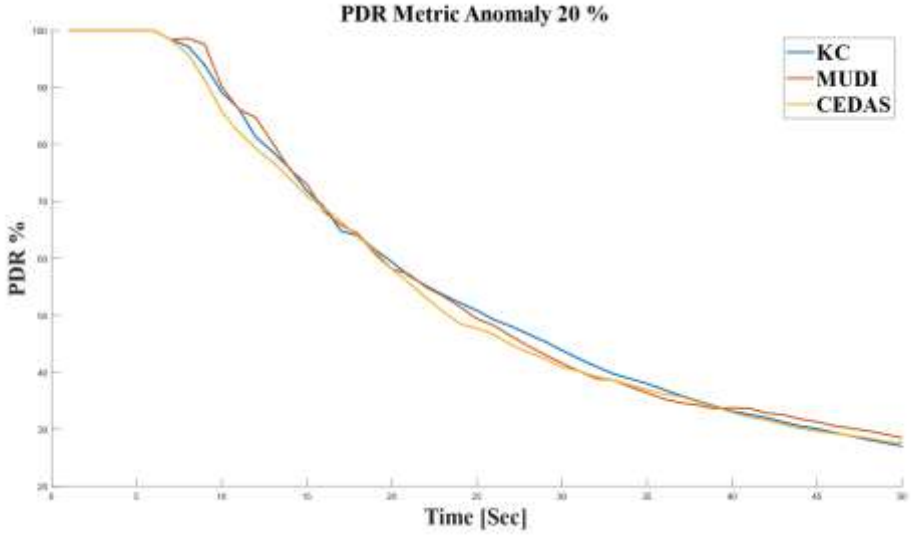

Figure D2.6. PDR Anomaly 20%.

APPENDIX D- 3 : IDS E2E & IDS PDR Metric speed for 1, 2 and 3 m\sec.

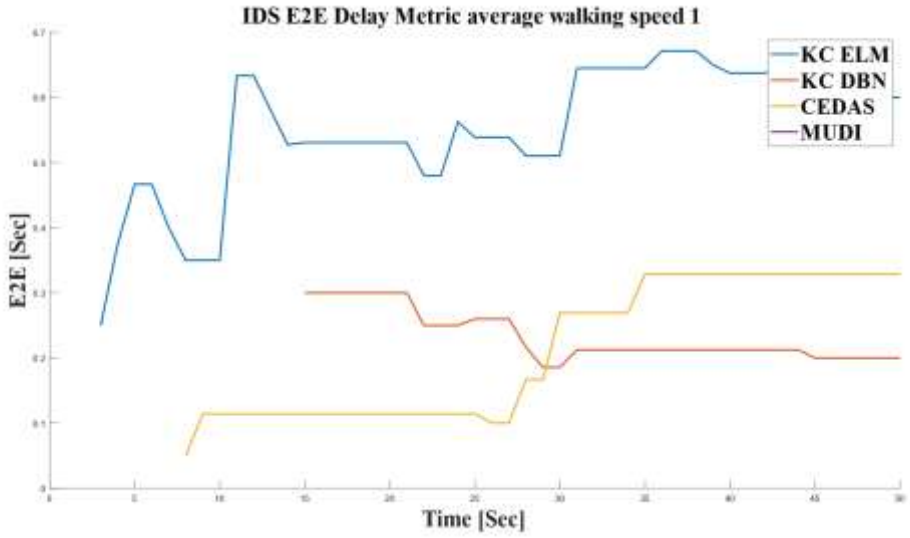

Figure D3.1. IDS E2E Delay Metric average walking speed 1 Plot.

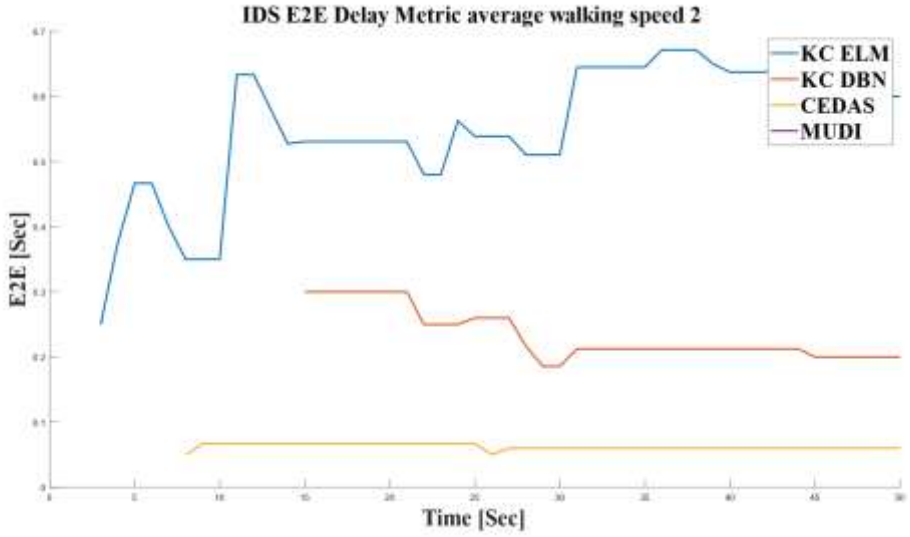

Figure D3.2. IDS E2E Delay Metric average walking speed 2 Plot.

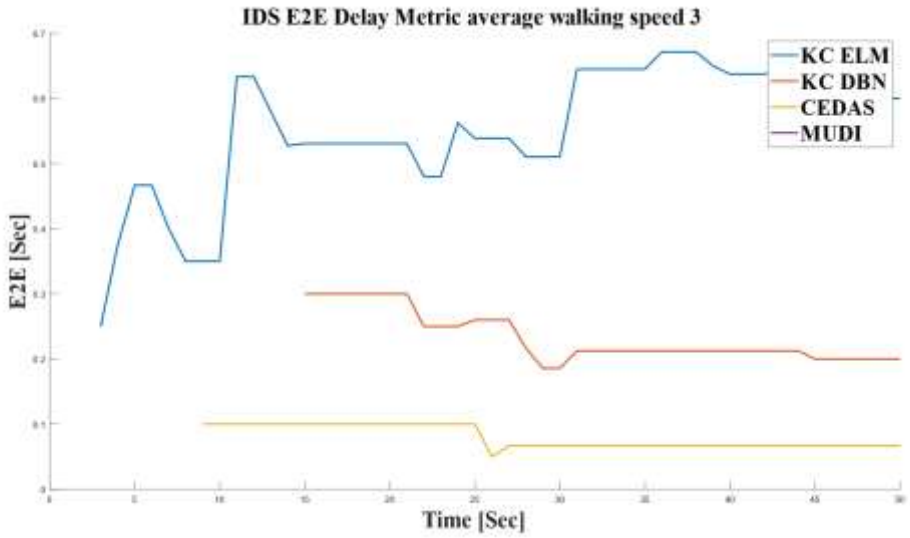

Figure D3.3. IDS E2E Delay Metric average walking speed 3 Plot.

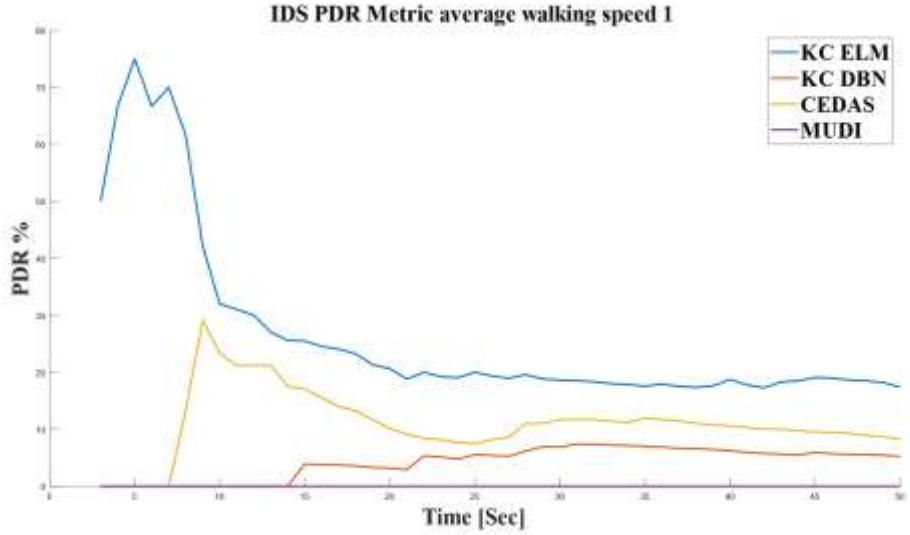

Figure D3.4. IDS PDR Metric average walking speed 1 Plot.

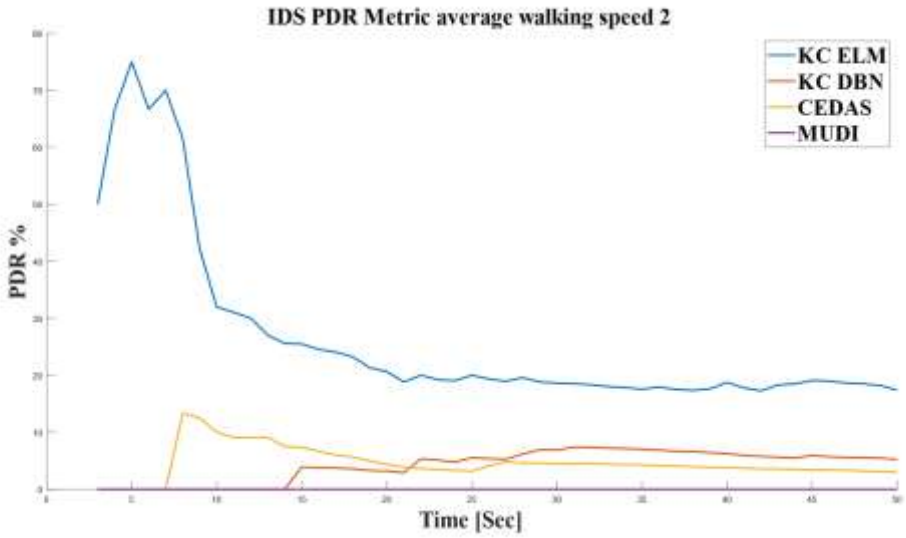

Figure D3.5. IDS PDR Metric average walking speed 2 Plot.

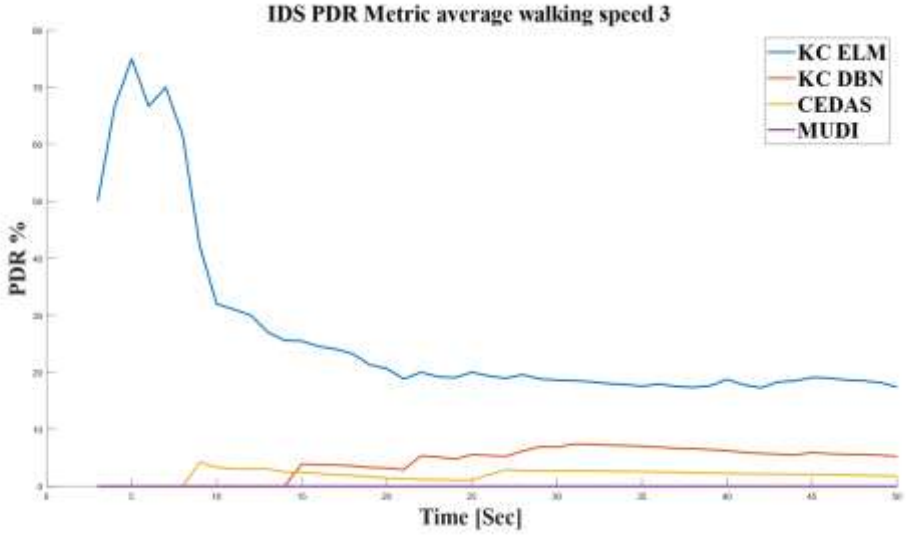

Figure D3.6. IDS PDR Metric average walking speed 3 Plot.

CURRICULUM VITAE

Name and Surname : Tamara Saad Mohamed AL-JANABI

Address : Baghdad, IRAQ

EDUCATION INFORMATION

Undergraduate : Almustansiriah University \ College Of Engineering \Computer And Software Engineering 2002-2006 \IRAQ.

Master : SRM University \ College Of Engineering \ Information Technology Engineering Department 2010-2012 \ INDIA.

Ph.D. : Aksaray University, 2017-2022, TURKEY.

PROFESSIONAL EXPERIENCE AND AWARDS:

Current and previous posts, with dates :

1. (2012-2015) work as Asst.Lecturer At Cihan University
2. Nominated to teach communication system course in Greenwich university/UK after interviewing done by their committee in order to open a new department(Business Information Technology) as a collaboration between Cihan university and Greenwich university.
3. (2016-2017) work as Asst.Lecturer At Al-Turath University.
4. (2017 – up to now) work as senior lecturer at College Of Baghdad Of Economic Sciences University \computer sciences department .
5. Editorial board member of Qubahan Academic Journal , http://journal.qubahan.com/index.php/qaj/about/editorialTeam
6. Reviewer in Qubahan Academic Journal , http://journal.qubahan.com/index.php/qaj/listofreviewers
7. International reviewer in i*nternational Research Journal of Science, Technology, Education, and Management* (IRJSTEM)
8. Editorial board member in many international journals in VIT press and member of Scope Database-International Advisory Board.

PUBLICAIONS, PRESENTATIONS AND PATENTS PRODUCED FROM THESIS

1. Mohamed,T and Aydin,S 2021. IoT-Based Intrusion Detection Systems: A Review,August2021,Smart journal ,DOI:10.1080/23080477.2021.1972914, publisher: Taylor and Francis.
2. Mohamed,T., Aydin,S.,Alkhayat,A.,and Malik,R 2022. Kalman and Cauchy clustering for Anomaly Detection based Authentication of IoMTs using extreme learningmachine,IET Communications journal,publisher:Wiley,DOI: :https://doi.org/10.1049/cmu2.12467.