

UNIVERSITY OF BAHRAIN
College of Information Technology



Securing Internet of Healthcare Things using a Machine Learning-based Approach for Intrusion Detection

A Thesis Submitted in Partial Fulfilment of the Requirements for the
M.Sc. Degree in Cyber Security

Submitted by
Marwa Ahmed Anas Mahmood Fouda
20105915

Supervised by
Dr. Riadh Bin Mohammad Ksantini
Associate Professor

Dr. Wael Mohamed Elmedany
Associate Professor

University of Bahrain

Kingdom of Bahrain
October 2021



Dissertation Deposit Form

Author: Marwa Ahmed Anas Mahmood Fouda

Thesis Title: Securing Internet of Healthcare Things using a Machine Learning-based Approach for Intrusion Detection

Year: 2021

College: Information Technology

Email: 20105915@stu.uob.edu.bh

Department: Information Systems

Contact No: 36144480/36428821

Author's Declaration: I agree to the following conditions:

The above thesis will be available (in the University of Bahrain library and externally) and reproduced as necessary at the discretion of the University of Bahrain. It may also be digitized by the University of Bahrain and made available via the university's repository or via other parties on the Internet.

This condition shall apply to ALL copies including electronic copies.

The above thesis has been provided on the understanding that it is copyright protected material and that no quotation from this thesis may be published without proper acknowledgement.

Signature of the Author:-----

A handwritten signature in blue ink, appearing to read "Fouda", is written over a dashed line.

Date: 01-10-2021

DECLARATION

I declare that this work is the result of my own investigation and that it has not already been accepted in substance for any degree, nor is it currently submitted for any degree.

Signed:  Date: 01/10/2021
Marwa Ahmed Anas Mahmood Fouda (Candidate)

The Thesis defense committee considers the thesis titled:

Securing Internet of Healthcare Things using a Machine Learning-based Approach for Intrusion Detection

Submitted by Marwa Ahmed Anas Mahmood Fouda to the College of Information Technology is satisfactory and acceptable for the Master of Science in Cyber Security.

The Defense Committee:

 ----- Dr. Riadh Bin Mohammad Ksantini (Supervisor)	Department of Computer Science University of Bahrain
 ----- Dr. Wael Mohammed Elmedany (Co-Supervisor)	Department of Computer Engineering University of Bahrain
 ----- - Dr. Mhd Saeed Sharif (External Examiner)	University of East London (UEL) United Kingdom
 ----- Prof. Faisal Qaseem Alkhateeb (Internal Examiner)	Department of Computer Science University of Bahrain

Abstract

Despite the significant benefits that the IoHT offered to the medical sector, there are concerns and risks regarding these systems which can delay their wide deployment as they handle sensitive, and often life-critical medical information. In addition to the IoHT concerns and risks, there are security constraints which include hardware, software, and network constraints that pose a security challenge to these systems. Therefore, security measures need to be deployed that can overcome the concerns, mitigate the risks, and meet the constraints of the IoHT. For these reasons, a subclasses intrusion detection system for the IoHT is proposed in this research work based on a novel variation of the standard OSVM, namely SDOSVM, which considers subclasses in the target class, i.e. normal class, in order to minimize the data dispersion within and between subclasses thus improving the discriminative power and classification performance of the intrusion detection system.

To the best of knowledge, this research work is the first to explore the usage of subclasses one-class classification in anomaly-based IoHT network intrusion detection in order to incorporate IoHT data subclasses information related to diverse normal behaviors, and improve detection accuracy.

The proposed subclasses intrusion detection system was evaluated using the real-world TON_IoT dataset, and compared to other state-of-the-art ML one-class classifiers. Experimentation results showed that the proposed approach outperformed the other relevant ML one-class classifiers for network intrusion detection in terms of AUC, where the AUC scores achieved by the proposed approach were above 96%. Also, the proposed approach outperformed the one-class SVM-based classifiers in terms of training time, which did not exceed one second, due to its reliance on a low number of SVs thus resulting in lower computational complexity. Finally, the proposed approach achieved the best results than the one-class classifiers for the FPR, which did not exceed 1%, and the TPR in most evaluation subsets of the TON_IoT dataset which was above 98.5%.

Dedication

To my father, mother, and my supportive family for their encouragement, and standing by my side in every aspect of my success.

And,

To my friends and colleagues for boosting my confidence, and their support throughout this journey.

Acknowledgements

I would like to express my special thanks and appreciation to my supervisors Dr. Riadh Bin Mohammad Ksantini and Dr. Wael Mohamed Elmedany for their kind patience, and for helping me with everything possible to develop this research work. I would also like to thank Dr. Khaoula Tbarki for her kind support, and great assistance in developing this research work as well.

Table of contents

Abstract	C
Dedication	D
Acknowledgements	E
Table of Contents	F
List of Tables	H
List of Figures	I
List of Abbreviations	J
Chapter One: Introduction	2
1.1 Overview	2
1.2 Background	2
1.3 Problem Definition	5
1.4 Research Aim and Objectives	6
1.5 Research Significance	7
1.6 Research Limitations	8
1.7 Road Map	8
1.8 Summary	9
Chapter Two: Literature Review	12
2.1 Overview	12
2.2 Related Work	12
2.2.1 Host-based and Network-based Intrusion Detection Systems	16
2.2.2 Intrusion Detection Methods	18
2.2.3 Classifiers	20
2.2.4 Subclasses Intrusion Detection Systems	22
2.3 Discussion	23
2.4 Summary	24
Chapter Three: Methodology	26
3.1 Overview	26
3.2 Proposed SDOSVM Intrusion Detection System	26
3.2.1 Feature Pre-processing	26
3.2.2 Classification using SDOSVM Algorithm	28
3.3 Proposed Algorithms	33

3.4	Experimentation.....	33
3.4.1	Datasets Used	35
3.4.2	Performance Metrics	36
3.4.3	Experimental Protocol	38
3.4.4	Classifiers	42
3.5	Summary	44
Chapter Four: Results and Discussion		46
4.1	Overview.....	46
4.2	Limitations of Proposed Model	47
4.3	Discussion.....	48
4.3.1	ROC Curve and AUC	48
4.3.2	P-value.....	50
4.3.3	FPR and TPR	51
4.3.4	Number of SVs and Training Time.....	51
4.4	Summary	53
Chapter Five: Conclusion and Recommendations		56
5.1	Overview.....	56
5.2	Conclusion	57
5.3	Recommendations and Future Work	57
References.....		59

List of Tables

2.1	Summary of state-of-the-art ML intrusion detection systems for IoT and IoHT	13
3.1	Real-world IoT datasets used for the experimentation	36
3.2	TON_IoT selected best features for the experimentation	40
3.3	Optimal number of clusters obtained per subset for the TON_IoT dataset	41
4.1	Average AUC per subset for the TON_IoT dataset (best method in bold)	49
4.2	Average FPR and TPR per subset for the TON_IoT dataset	52
4.3	Average number of SVs and training time per subset for the TON_IoT dataset	53

List of Figures

2.1	ML intrusion detection taxonomy for the IoHT	16
3.1	The proposed SDOSVM intrusion detection system	26
3.2	Features' scores of the TON_IoT dataset	39
3.3	10-fold cross-validation method	42
4.1	Average ROC curves of one-class classifiers per subset for the TON_IoT dataset	50

List of Abbreviations

ABOD	Angle-Base Outlier Detection
AdaBoost	Adaptive Boosting
AI	Artificial Intelligence
AIDS	Anomaly Intrusion Detection System
ANN	Artificial Neural Network
AUC	Area Under (the Receiver Operating Characteristic) Curve
Bayes Net	Bayesian Network
CNN	Convolutional Neural Network
cuDNNLSTM	cuda Deep Neural Network Long Short Term Memory
DAE-DFFNN	Deep AutoEncoder-Deep Feed Forward Neural Network
DDoS	Distributed Denial of Service
DL	Deep Learning
DNN	Deep Neural Network
DoS	Denial of Service
DT	Decision Tree
EE	Elliptic Envelope
EOS-ELM	Ensemble of Online Sequential-Extreme Learning Machine
FMLE	Fuzzy Maximum Likelihood Estimation
FN	False Negative
FP	False Positive
FPR	False Positive Rate
HADL	Hierarchical Anomaly Detection and Localization
HIDS	Host-based Intrusion Detection System
IF	Isolation Forest
IIoT	Industrial Internet of Things
IoHT	Internet of Healthcare Things
IoT	Internet of Things
IoTDS	Internet of Things Detection System
KNN	K-Nearest Neighbor

LOF	Local Outlier Factor
MiTM	Man in The Middle
ML	Machine Learning
MOPF	Modified Optimum Path Forest
MOSVM	Mahalanobis One-class Support Vector Machine
NB	Naive Bayes
NIDS	Network-based Intrusion Detection System
NN	Neural Network
OC-NN	One-Class-Neural Network
OPFC	Optimum Path Forest Clustering
OSVM	One-class Support Vector Machine
PCA	Principal Component Analysis
p-value	probability-value
RBF	Radial Basis Function
RF	Random Forest
ROC	Receiver Operating Characteristic
SCAD	Soft Computing-based Anomaly Detection
SDOSVM	Subclasses Dispersion One-class Support Vector Machine
SGD	Stochastic Gradient Descent
SIDS	Signature Intrusion Detection System
SOGAAL	Single-Objective Generative Adversarial Active Learning
SOS	Stochastic Outlier Selection
SV	Support Vector
SVDD	Support Vector Data Description
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
TPR	True Positive Rate
WSN	Wireless Sensor Network
XSS	Cross-Site Scripting

Chapter 1

Introduction

- 1.1. Overview
- 1.2. Background
- 1.3. Problem Definition
- 1.4. Research Aim and Objectives
- 1.5. Research Significance
- 1.6. Research Limitations
- 1.7. Road Map
- 1.8. Summary

Chapter One

Introduction

1.1. Overview

Machine Learning (ML) intrusion detection systems for the Internet of Healthcare Things (IoHT) is one of the most common and important research problems in security. Many studies have been conducted on ML intrusion detection systems for the IoHT, but to the best of knowledge none of these studies considered the subclasses data distribution for the normal behavior in the IoHT to develop their proposed intrusion detection systems. Indeed, the concept of subclasses data distribution can be used to build an intrusion detection system for the IoHT, and improve its discriminative power and classification performance through the minimization of data dispersion within and between subclasses in the target class which represents the normal behaviors in the IoHT.

In the literature, the use of subclasses has been considered in developing intrusion detection systems using hierarchical data distribution. However, such systems did not consider the subclasses dispersion information to improve their classification performance.

Therefore, in this research work a novel subclasses intrusion detection system for the IoHT is proposed, and its performance is validated using real-world IoHT datasets as well as compared with state-of-the-art ML one-class classifiers.

1.2. Background

Recent progression in computer technologies, and the fast expansion of the Internet lead to the evolution of the Internet of Things (IoT) (Ghosal

et al., 2018; Kumar et al., 2021). The IoT allows the smooth integration of both physical and digital worlds through a network of physical objects embedded with sensors (Thamilarasu et al., 2020). Such a paradigm gave rise to smart environments like smart cities, smart homes, smart transportation, etc. (Kumar et al., 2021)

The IoHT, a subdomain of the IoT, is an ecosystem of various connected devices which include sensors, wearable devices, medical devices, and clinical systems that enable several applications in the medical sector like remote patient monitoring, fitness programs, and chronic diseases and elderly care. The IoHT devices collect the patient's real-time and sensitive data, transfer the collected data through a gateway for storage in a cloud platform, and redirect the stored data to the required healthcare providers (Kumar et al., 2021; R.M. et al., 2020; Thamilarasu et al., 2020).

The IoHT has benefited the medical sector through remote monitoring of the patient's health status without any visits by him/her to the healthcare facility. This resulted in time and cost savings, enhanced quality of healthcare services, improved quality of patient's health through better support and medication as healthcare providers can effectively and accurately understand critical health conditions, patients becoming well-informed about the improvements in their health status and treatments, and increased patient interest and satisfaction through simple and effective communication with healthcare providers (Ghosal et al., 2018; Kumar et al., 2021; R.M. et al., 2020; Thamilarasu et al., 2020).

Despite the aforementioned benefits of the IoHT, there are security and privacy concerns regarding this subdomain which can delay its wide deployment as such systems handle sensitive, and often life-critical medical information. Cyber-attacks on the IoHT and its components can threaten its confidentiality, integrity, availability, and authentication

which can result in severe physical and life-threatening harm to the patients. For example, hacking a medical insulin pump can result in insulin over dose, and possible death of the patient. Also, hacking a connected cardiac device, for example a pacemaker, can threaten a patient's life (Kumar et al., 2021; Thamilarasu et al., 2020).

The security of the IoHT is also accompanied with constraints that need to be considered, and are categorized to hardware, software, and network constraints. The hardware constraints include computational, energy, and memory constraints. The software constraints include embedded software constraints, and dynamic security patch. Finally, the network constraints include scalability, diverse IoT devices, diverse communication medium, multi-protocol networking, and dynamic network topology (Yaacoub, et al., 2020).

ML intrusion detection systems became the most promising technique for securing the IoHT. This technique can recognize attacks especially unknown or zero-day attacks that traditional intrusion detection systems are unable to identify through the detection of variations in the characteristics of network traffic. Even though ML may not be a convenient choice for problems requiring a formal descriptive solution, strong results can be achieved by it in areas and issues that can be difficult to formalize. Therefore, ML is proficient in both data clustering and classification fields which are the main blocks in data security applications (Chaabouni et al., 2019; Hady et al., 2020).

Most Internet security models block malicious behaviors using a list of these behaviors. However, attackers continuously improve their techniques making it difficult to detect their adaptive malicious behaviors by these models. Therefore, describing all possible malicious behaviors in a list, and continuously updating that list in these models is considered impractical, and extremely resource-intensive. ML can model normal

behavior such that any incoming event that deviates from the modeled normal behavior is considered an anomaly which is more likely a malicious behavior. This makes ML a more practical security approach. Also, ML intrusion detection systems have low to medium resource consumption, and therefore, resources are used effectively (Chaabouni et al., 2019; Hady et al., 2020).

The growing demands and growth of the IoHT require improvements in security solutions. Therefore, a ML analytical tool is often used to analyze the massive bulk of data, and provide worthy knowledge in the classification, decision analysis, and cyber-attacks detection (Kumar et al., 2021).

ML algorithms use training and learning from experience to improve their performance in fulfilling a task. For a ML intrusion detection system, the task is to classify a system behavior as normal or abnormal. To improve the performance of this system, classification accuracy needs to be improved, and the collection of normal behaviors are the experiences that the ML algorithms learn from (Al-Garadi et al., 2020).

Traditional network intrusion detection systems experience high false detection rates that can be reduced using ML techniques. A ML intrusion detection system uses statistical, genetic, and heuristics or a combination of them to improve the detection rate, and reduce false negatives through dissemination of complex attack patterns (Chaabouni et al., 2019).

1.3. Problem Definition

ML one-class intrusion detection systems have drawn attention in the IoHT to overcome the drawbacks of both binary and multi-class ML intrusion detection systems which include the high cost of data labelling,

and the class imbalance problem. However, ML one-class intrusion detection systems for the IoHT did not consider subclasses data distribution for the normal behavior in the IoHT environment where the definition of normal behavior can vary from one system to another in these environments. Also, the ML binary class and multi-class hierarchy subclasses intrusion detection systems did not consider the subclasses dispersion information.

Therefore, a novel ML subclasses intrusion detection system for the IoHT is proposed that presents a novel variation of the standard One-class Support Vector Machine (OSVM) for classification, namely the Subclasses Dispersion One-class Support Vector Machine (SDOSVM). The proposed SDOSVM approach considers the subclasses in the IoHT normal data class, which is the target class in this research work, in order to minimize the data dispersion within and between subclasses thus improving the discriminative power and classification performance of the intrusion detection system.

1.4. Research Aim and Objectives

The use of subclasses dispersion information in IoHT intrusion detection systems was not considered in order to improve their classification performance. The aim of this research work is to incorporate subclasses dispersion information in IoHT intrusion detection systems to improve their discriminative power and classification performance. A novel ML subclasses intrusion detection system was developed, and its performance was validated for the IoHT through high performance metrics that included Area Under (the Receiver Operating Characteristic (ROC)) Curve (AUC), the number of Support Vectors (SVs), and the training time.

The main objectives of this research work are as follows:

1. Develop a novel ML subclasses intrusion detection system for the IoHT.
2. Validate the developed intrusion detection system for the IoHT through high performance metrics, and publish the validation results.

The guiding research questions for developing the proposed ML subclasses intrusion detection system are as follows:

1. What is the suitable data source (host-based/network-based)?
2. What is the suitable detection method (signature-based/anomaly-based/hybrid)?
3. What is the suitable type of data classifier (one-class/binary class/multi-class)?
4. What are the state-of-the-art subclasses intrusion detection systems available in the literature?

1.5. Research Significance

This research work proposes a novel ML subclasses one-class anomaly-based network intrusion detection system for securing the IoHT that considers subclasses data distribution in the IoHT normal data class to improve the discriminative power and classification performance of the intrusion detection system through the minimization of data dispersion within and between subclasses. To the best of knowledge, this is the first research work to explore the usage of subclasses for the one-class classification problem in the anomaly-based IoHT network intrusion detection.

Hence, this work will be valuable to cyber-security practitioners and researchers in terms of securing the IoHT, and overcoming the challenges

posed by these systems to apply security measures. The results of this work will be shared with the cyber-security community in hopes of giving them information about the effective ML intrusion detection systems that can secure the IoHT, and accelerate its wide deployment in the medical sector.

1.6. Research Limitations

The main limitation of this research work was the benchmark datasets required for experimentation. Most of the benchmark datasets used for validating the IoHT network intrusion detection systems contain homogeneous normal data, i.e. there are no subclasses within the normal data class, and therefore, these datasets were excluded from the experimentation of the proposed ML subclasses intrusion detection system. There was only one benchmark dataset with heterogeneous normal data found, namely the TON_IoT, and this was the dataset used for experimentation.

1.7. Road Map

This research work is divided into 5 main chapters, each chapter describes a part of the research work in detail as follows:

1. **Chapter 1:** This is the introductory chapter of this research work, and gives a brief background about the IoHT, its benefits, its security concerns and constraints, and the motivation behind the use of ML intrusion detection systems for the IoHT. The problem definition, research aim and objectives, research significance, and research limitations are also described in this chapter.
2. **Chapter 2:** This chapter, which is entitled Literature Review, provides a review of state-of-the-art ML intrusion detection

systems for both the IoT and IoHT. State-of-the-art ML subclasses intrusion detection solutions are also presented followed by the contribution of this research work.

3. **Chapter 3:** This chapter, which is entitled Methodology, explains in detail the proposed ML subclasses intrusion detection system as well as the experimentation datasets, performance metrics used to validate the proposed system, the relevant ML one-class classifiers compared to the proposed approach, and the experimental protocol.
4. **Chapter 4:** This chapter, which is entitled Results and Discussion, presents the experimentation results obtained when the proposed approach was applied on the benchmark datasets. The comparative analysis of the proposed approach with other state-of-the-art ML one-class classifiers is also presented. The strength and limitations of the proposed approach are also discussed.
5. **Chapter 5:** This final chapter, which is entitled Conclusion and Recommendation, summarizes the work done in this research work, and the recommendations and future work.

1.8. Summary

This introductory chapter gave a brief background about the IoHT, its benefits to the medical sector, its security concerns and constraints, and the motivation behind the use of ML intrusion detection systems for securing the IoHT. The problem definition section described the drawback of state-of-the-art ML one-class and subclasses intrusion detection systems which is not considering the normal subclasses dispersion information to improve the classification performance of the

intrusion detection system, and the need for a ML subclasses intrusion detection system to overcome such a drawback. The research aim and objectives, research significance, and research limitations are also described in this chapter along with the road map of this research work.

Chapter 2

Literature Review

2.1. Overview

2.2. Related Work

2.3. Discussion

2.4. Summary

Chapter Two

Literature Review

2.1. Overview

ML intrusion detection systems for the IoT and IoHT is one of the most common and important research problems in security. Many studies have been conducted on ML intrusion detection systems for the IoT and IoHT, and these studies are reviewed in this chapter.

The purpose of conducting this literature review, is to gain knowledge of progression done so far in ML intrusion detection systems for the IoT and IoHT in order to develop a novel ML intrusion detection system that yields better results for securing the IoHT. This review is provided in detail in section 2.2, and the contribution of this research work based on the conducted review is provided in section 2.3.

2.2. Related Work

In this section, a review of state-of-the-art ML intrusion detection systems for the IoT and IoHT is provided. A summary of the reviewed ML intrusion detection systems is illustrated in Table 2.1.

Table (2.1): Summary of state-of-the-art ML intrusion detection systems for IoT and IoHT

Reference	Data Source	Classifier	Intrusion Detection Method	Algorithm
AL-Hawawreh et al. (2018)	Network	Binary class	Anomaly-based	Deep AutoEncoder-Deep Feed Forward Neural Network (DAE-DFNN)
Alrashdi et al. (2019)	Network	Binary class	-	Ensemble of Online Sequential-Extreme Learning Machine (EOS-ELM)
Bagaa et al. (2020)	Network	Multi-class for signature-based detection One-class for anomaly-based detection	Hybrid	For signature-based detection, three approaches were used: Approach (1): Supervised algorithms (J48, Bayesian Network (Bayes Net), Random Forest (RF), and Hoeffding tree) Approach (2): Multi-layer Neural Network (NN) using backpropagation learning Approach (3): Distributed classification using Adaptive Boosting (AdaBoost) algorithm For anomaly-based detection: OSVM
Bakhtiar et al. (2019)	Network	Multi-class	-	J48 Decision Tree (DT)
Bezerra et al. (2019)	Host	One-class	Anomaly-based	Elliptic Envelope (EE), Isolation Forest (IF), Local Outlier Factor (LOF), OSVM

Reference	Data Source	Classifier	Intrusion Detection Method	Algorithm
Ciklabakkal et al. (2019)	Network	Multi-class One-class	Anomaly-based	Autoencoder, IF, OSVM, RF, Single-Objective Generative Adversarial Active Learning (SOGAAL), K-means clustering
Efstathopoulos et al. (2019)	Operational data in smart grids	One-class	Anomaly-based	OSVM, IF, Angle-Base Outlier Detection (ABOD), Stochastic Outlier Selection (SOS), Principal Component Analysis (PCA), deep fully connected autoencoders
Eskandari et al. (2020)	Network	One-class	Anomaly-based	IF, LOF
Ghosal et al. (2018)	Network	Binary class	-	-
Hady et al. (2020)	Network and patient's biometric	Binary class	Anomaly-based	Artificial Neural Network (ANN), K-Nearest Neighbor (KNN), Linear SVM, RF
Kamble and Gawade (2019)	Network	Multi-class	Pre-defined threshold	Q-learning
Khraisat et al. (2019)	Network	One-class	Hybrid	For signature-based detection: C5.0 DT For anomaly-based detection: OSVM
Kumar et al. (2021)	Network	Binary class	Anomaly-based	Ensemble learning with two learning levels: First level uses three learning algorithms: DT, Naive Bayes (NB), RF Second level uses XGBoost

Reference	Data Source	Classifier	Intrusion Detection Method	Algorithm
Liaqat et al. (2020)	Network	Binary class	-	Hybrid Deep Learning (DL): cuda Deep Neural Network Long Short Term Memory-Convolutional Neural Network (cuDNNLSTM-CNN)
R.M. et al. (2020)	Network	Binary class	Anomaly-based	Deep Neural Network (DNN)
Said et al. (2020)	Network	One-class	Anomaly-based	OSVM
Sheikhan and Bostani (2016)	Network	Multi-class	Hybrid	For signature-based detection: Modified Optimum Path Forest (MOPF) For anomaly-based detection: Optimum Path Forest Clustering (OPFC)
Shorman et al. (2020)	Network	One-class	Anomaly-based	OSVM
Thamilarasu et al. (2020)	Network and device	Multi-class and binary for network level detection One-class for device level detection	Anomaly-based	DT, KNN, NB, RF, SVM for network level detection Polynomial Regression for device level detection
Yahyaoui et al. (2019)	Network	Binary class for gateway level detection One-class for Wireless Sensor Network (WSN) level detection	Anomaly-based	For WSN detection level: OSVM For gateway detection level: Deep learning

The thematic taxonomy of the ML intrusion detection systems for the IoHT shown in Figure 2.1 is discussed in the following subsections. The blue boxes in the figure demonstrate the classification of the proposed approach in each level.

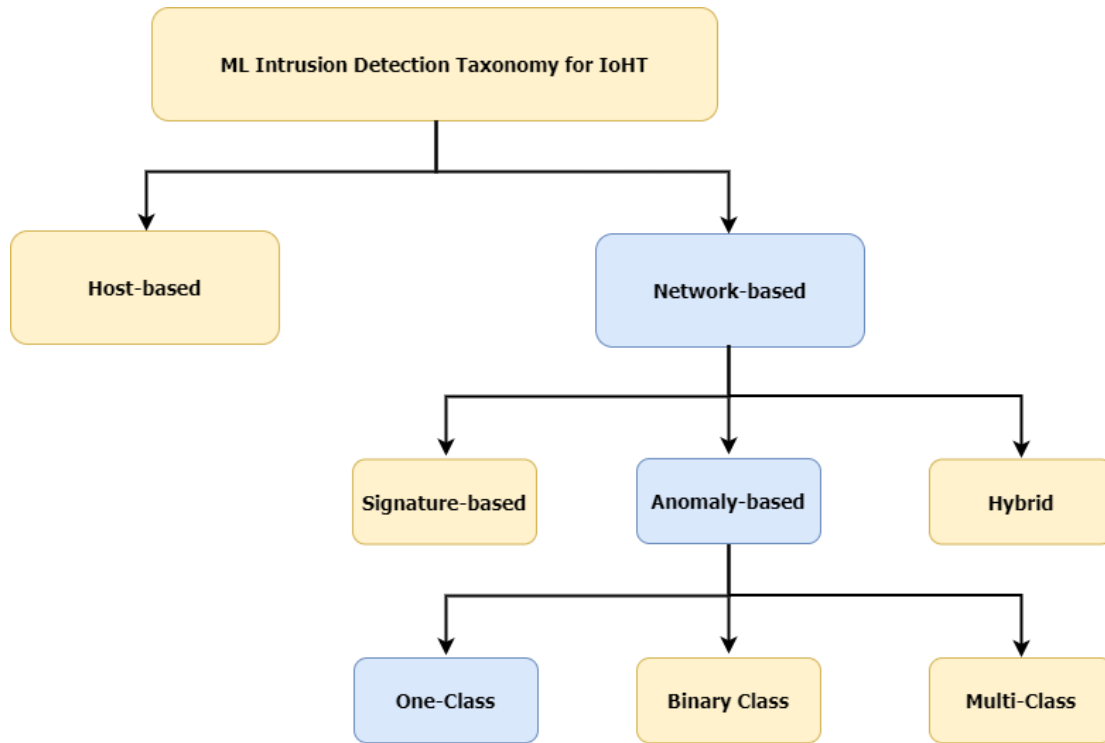


Figure (2.1): ML intrusion detection taxonomy for the IoHT

2.2.1. Host-based and Network-based Intrusion Detection Systems

The IoT and IoHT intrusion detection systems can be categorized based on the data source to Host-based Intrusion Detection System (HIDS) and Network-based Intrusion Detection System (NIDS).

According to Chaabouni et al. (2019), the HIDS is suitable for inside intrusion detection. Also, Bezerra et al. (2019) used the HIDS approach to overcome the drawback of the NIDS where the end-to-end encryption is used in most network communications, thus making the acquisition of useful information from the network traffic more difficult.

Bezerra et al. (2019) proposed a host-based Internet of Things Detection System (IoTDS) which depends on the host device resource consumption data due to its sensitivity to the presence of bot malware, and therefore, it was able to detect the malware present within the host device accurately. Also, because of its dependency on the host device

resource consumption data, the IoTDS malware detection capability was not obstructed by the encryption of malware traffic. The authors considered the impact of IoTDS on the host device's performance where the experiments showed that the host device remained functional despite the increase in resource consumption when the intrusion detection system was run on the host device. The authors stated that pre-installation of IoTDS can be done in IoT devices, thus providing out of the box protection to these devices. The authors also recommended the combination of other types of data, such as network-based data, with host-based data for a more secure system that includes both the hosts and the network.

However, Chaabouni et al. (2019) did not recommend the HIDS for the heterogeneous and large IoT systems due to the per-device security approach of the HIDS where each device is analyzed separately. Also, Shorman et al. (2020) did not recommend the use of the HIDS for several reasons. First, the IoT manufacturers cannot be trusted to install the required HIDSs on the IoT devices. Second, the limited access to some of the IoT devices, such as wearables, is another drawback of the HIDS as it is not possible to compel their installation on these devices. Finally, the HIDS consumes resources when deployed in the resource-constraint IoT devices, and harms their functionality. Therefore, these systems need to be lightweight with respect to the resource-constraint IoT devices, and preserve their functionality.

On the other hand, the NIDS is suitable for external intrusion detection. Also, these systems offer more degree of freedom than the HIDS for developing security solutions as they are suitable for the resource-constraint IoT systems. The NIDS can be used as a single system or multiple systems that can monitor the entire IoT network

making them suitable for large and heterogeneous IoT systems (Chaabouni et al., 2019).

2.2.2. Intrusion Detection Methods

The intrusion detection methods for the IoT and IoHT can be categorized to signature-based, anomaly-based, and hybrid.

In the signature-based detection, also known as abuse detection or rule-based detection or misuse-detection, predefined attack patterns are modeled, and preserved in the attacks' signature database. If the pattern of an incoming traffic matches any of the attacks' signatures in the database, then the administrator is alerted (Kumar et al., 2021; Sheikhan and Bostani, 2016).

Signature-based detection can detect known attacks, and thus has high detection rates for these attacks. However, it suffers from poor performance in detecting unknown attacks. Also, another drawback of signature-based detection is that it is not suitable for real-time intrusion detection because any increase in the number of newly recognized attacks results in an increase in the number of signatures, and therefore, more comparisons will be required between the stored signatures and incoming events. This can cause an overload for the intrusion detection system, and thus affects its response. Therefore, these systems neglect the examination of incoming events at specific rates based on the available processing resources. Furthermore, signature-based detection requires human experts to develop signatures for new attacks, and this can consume time in order to study the characteristics of the new attack and generate its signature (Eskandari et al., 2020; Shorman et al., 2020).

Because of the aforementioned drawbacks of the signature-based detection method, anomaly-based and hybrid detection methods emerged as an alternative for ML intrusion detection systems in the IoT and IoHT.

In the anomaly-based detection, a typical normal network behavior is modeled, and any deviations between incoming network traffic and normal network behaviors or system activities is considered an anomaly which is most likely an intrusion (Kumar et al., 2021; Sheikhan and Bostani, 2016).

Anomaly-based detection can detect both known and unknown attacks because it assumes that the difference between normal and abnormal behavior can be detected (Shorman et al., 2020).

The anomaly-based detection was used by Kumar et al. (2021) because IoHT environments are heterogeneous and diverse which practically limits the usage of a signature-based detection method in the protection of these environments from cyber-attacks. Therefore, this method is preferable for designing a strong intrusion detection system.

R.M. et al. (2020) preferred the anomaly-based detection than the signature-based detection for their intrusion detection system because it provides a scalable solution to handle the dynamic and random behaviors of malicious attacks in an IoHT environment, thus making their system effective and efficient in classifying and detecting unexpected attacks. A lightweight anomaly-based approach was used by Eskandari et al. (2020) in their Passban intrusion detection system for the same reason because of its capability of detecting new attacks without any hardware upgrades making it an effective approach for resource-constraint IoT devices.

However, a major challenge for anomaly-based detection is the difficulty of generating a model for a singular normal behavior of a system which consists of diverse behaviors created by various data sources. This results in high False Positive Rate (FPR) because of the minimal likeness between incoming events and modeled normal behavior (Eskandari et al., 2020).

Khraisat et al. (2019) used both signature-based and anomaly-based detection methods for their two stage novel hybrid intrusion detection system that takes advantage of both methods, and overcomes their limitations. In the first stage of the hybrid intrusion detection, the signature-based detection was used to identify known attacks. In the second stage, the anomaly-based detection was used to identify unknown or zero-day attacks without having a negative impact on the energy consumption of the IoT node due to the low processing costs of this method. This lead to improvement in the True Positive Rate (TPR), and reduction of the FPR compared to the TPR and FPR performance for each approach implemented individually. Also, for the Signature Intrusion Detection System (SIDS) to have an updated history of known attacks, the attacks recognized by the Anomaly Intrusion Detection System (AIDS) were saved in the SIDS database for protection from similar attacks in the future.

2.2.3. Classifiers

The classifiers used in the IoT and IoHT can be categorized to one-class, binary class, and multi-class classifiers.

One-class classifiers, a special type of ML technique, model a single pattern, i.e. all data samples in the training dataset belong to a single class of interest usually the normal class for the intrusion detection system, and then that model is used to recognize if a new data sample belongs to that pattern or class. On the other hand, both binary class and multi-class classifiers recognize a new data sample as belonging to one of two or more pre-defined patterns, respectively (Bezerra et al., 2019; Eskandari et al., 2020).

Binary class and multi-class classifiers for the IoT and IoHT intrusion detection systems require labelled data, where each label

represents a class of interest, when training a model that has the ability of detecting each class (Bezerra et al., 2019).

The above requirement is considered a drawback for these classifiers for the following reasons (Bezerra et al., 2019; Khraisat et al., 2019):

1. **Cost of data labelling can be too high:** For the case of malware detection, testbeds or honeypots that are designed specially for this purpose are used to collect the labelled data which requires vulnerable devices, and a special kind of access to malware to be done manually.
2. **Class imbalance problem:** Normal class data is available, and in larger scale unlike the malicious class data which are rare or do not even exist.

Because of the aforementioned drawbacks of both binary class and multi-class classifiers, one-class classifiers emerged as an alternative for the IoT and IoHT intrusion detection systems where a new model is trained using the normal data only as the target class, that is easily collected than the malicious data which is unavailable or poorly sampled or not well defined, and that model recognizes a new instance as normal if it fits within the boundaries of the model or malicious if it doesn't (Bezerra et al., 2019; Khraisat et al., 2019).

As a result (Bezerra et al., 2019; Eskandari et al., 2020):

1. Labelling costs as well as the necessity for malicious data samples in the training phase will be reduced significantly.
2. Easy adapting models, as periodical updates of the model using new normal data requires less effort than updates that use labelled normal and malicious data.
3. Effective models for resource-constraint IoT devices.

4. Suitable models for instant detection of anomalies in real-time network traffic monitoring due to the low computational cost of the model inference.

Bezerra et al. (2019) used the one-class classifier for their IoTDS to address the issues of both the binary class and multi-class classifiers which is the dependency on algorithms that need labelled data to train the model, and this can be a problem as it is difficult to obtain labelled data for botnets attacks as well as this data can become quickly outdated by new botnets and attacks.

Eskandari et al. (2020) lightweight Passban one-class anomaly-based intrusion detection system provides protection near the IoT network edge. One-class classification algorithms were used as they proved to be efficient for resource-constraint IoT devices.

Said et al. (2020) used the OSVM for their detection model for several reasons. First, its low energy consumption for the WSN when compared to other ML algorithms. Second, it was considered a convenient method to detect selective forward and jamming attacks. Finally, it works well with structured data as a table of values when compared to other ML algorithms.

Yahyaoui et al. (2019) proposed Hierarchical Anomaly Detection and Localization (HADL) in which a lightweight OSVM was used for detecting WSN routing attacks. It is executed from parent to child nodes on-demand in the WSN, and hence continuous network monitors are not required which reduces energy consumption and processing time.

2.2.4. Subclasses Intrusion Detection Solutions

One relevant study was found in the literature regarding state-of-the-art subclasses intrusion detection systems. This study is for Singh et al. (2019) who proposed an autonomous two stage intrusion detection system

based on hierarchical subclasses data distribution named Soft Computing-based Anomaly Detection (SCAD). The authors proved through experimentation the effectiveness of their system for both binary and multi-class classification. However, to reduce the training time the authors recommended the use of GPU and parallel processing.

2.3. Discussion

The IoHT is a growing and heterogeneous environment, and consists of resource-constraint devices. Therefore, the proposed intrusion detection system in this research work is recommended to be a NIDS to meet the IoHT security constraints mentioned in section 1.2.

The benefits of the one-class classifier over binary and multi-class classifiers discussed previously in section 2.2.3, made the one-class classifier recommended for the proposed intrusion detection system to meet the IoHT security constraints mentioned in section 1.2.

Since the one-class classifier models normal network behaviors, and identifies any event that deviates from that normal behavior as an outlier which is most likely an intrusion, the anomaly-based detection is used for the classification problem in this research work. Also, to overcome the challenge of this detection method mentioned previously in section 2.2.2, and the drawback of one-class intrusion detection systems which did not consider subclasses data distribution for the normal behaviors of the IoHT environments where the definition of normal behavior can vary from one system to another in these environments, a subclasses approach is proposed in this research work in order to incorporate the IoHT subclasses information related to diverse normal behaviors, and improve the FPR.

Although, state-of-the-art subclasses intrusion detection solutions were discussed in the literature review through the work of Singh et al.

(2019), and this solution proved its effectiveness through experimentation, they did not consider the subclasses dispersion information in order to improve the classification performance of their intrusion detection system.

Therefore, based on the conducted literature review a novel ML subclasses one-class anomaly-based network intrusion detection system for the IoHT is proposed in this research work that presents a novel variation of the standard OSVM for classification, namely the SDOSVM. The proposed SDOSVM approach considers the subclasses in the IoHT normal data class, which is considered the target class in this research work, in order to minimize the data dispersion within and between subclasses thus improving the discriminative power and classification performance of the intrusion detection system.

2.4. Summary

This chapter gave a review of the state-of-the-art ML intrusion detection systems for the IoT and IoHT. Also, a thematic taxonomy of ML intrusion detection systems for the IoHT was illustrated, and discussed in detail. The results of the conducted literature review were discussed followed by the contribution of this research work.

Chapter 3

Methodology

3.1. Overview

3.2. Proposed SDOSVM Intrusion Detection System

3.3. Proposed Algorithms

3.4. Experimentation

3.5. Summary

Chapter Three

Methodology

3.1. Overview

The proposed SDOSVM intrusion detection system consists of two steps: feature pre-processing, and classification using the proposed SDOSVM algorithm as illustrated in Figure 3.1. These steps are described in detail in the next section. The experimentation methodology is also described in detail within this chapter in section 3.4.

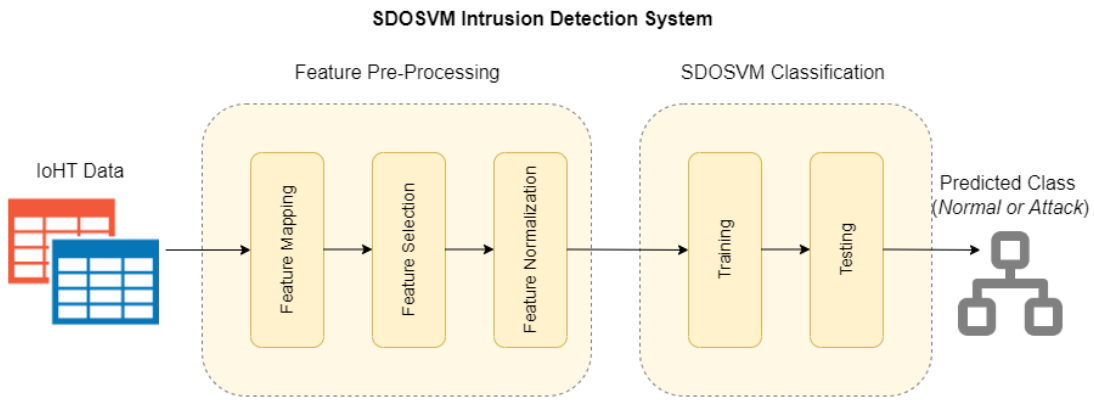


Figure (3.1): The proposed SDOSVM intrusion detection system

3.2. Proposed SDOSVM Intrusion Detection System

In this section, a detailed description of the proposed SDOSVM intrusion detection system is provided.

3.2.1. Feature Pre-processing

This subsection presents in detail the steps involved in feature pre-processing. The IoHT network traffic generated by various sensors has different types of features that include numerical and categorical values, and pre-processing these features as described below is important so that

the proposed intrusion detection system can achieve accurate and efficient results (Kumar et al., 2021).

Feature Mapping

The IoHT network traffic consists of numerical and categorical features. The values of categorical features need to be transformed to numeric ones because the proposed intrusion detection system can handle efficiently only numeric values (Kumar et al., 2021).

Feature Selection

This method involves the identification of proper features, and dismissing the irrelevant ones in order to obtain a features subset that best represent the classification problem, and with minimum performance degradation in the IoHT environment. The advantages of such a method include improvement in the performance of the intrusion detection system, and reduction in processing time (Kumar et al., 2021).

Feature Normalisation

The features of the IoHT traffic have dissimilar ranges of values. Therefore, these features need to be normalized, i.e. convert every value in a particular feature to a value in the data range [0,1]. The advantages of this method include speeding up the gathering and removal of bias from the raw data, ease pattern analysis, and support convergence and training time (Alrashdi et al., 2019; Kumar et al., 2021; Thamilarasu et al., 2020).

The following equation was used to normalize the features of the IoHT data:

$$F_{Normalized} = a + \frac{F - F_{min}}{F_{max} - F_{min}} (b - a), \quad (3.1)$$

where,

$F_{Normalized}$ are the new values for a feature F in X after normalization,

a is the minimum value for the normalized data range, i.e. 0,

b is the maximum value for the normalized data range, i.e. 1,

F is the feature to be normalized in X ,

F_{min} is the minimum value for feature F ,

F_{max} is the maximum value for feature F ,

$X = \{x_i\}_{i=1}^N$ is the training dataset of N samples.

3.2.2. Classification using SDOSVM Algorithm

This subsection presents in detail the mathematical formulation of the proposed SDOSVM algorithm, using linear and nonlinear kernel functions, in order to handle linear and nonlinear decision boundaries between data.

Linear SDOSVM

The proposed SDOSVM algorithm is a novel variation of the standard OSVM that considers subclasses in the target class, i.e. the normal class, in order to minimize the data dispersion within and between subclasses thus improving the discriminative power and classification performance. To achieve this, the within and between subclasses matrix of Eq. (3.2) will be added to the standard OSVM primal optimization problem of Eq. (3.5).

The within and between subclasses matrix M_{sub} can be formulated as:

$$M_{sub} = M_W^{sub} + M_B^{sub}, \quad (3.2)$$

where,

M_W^{sub} represents the within subclass matrix,

M_B^{sub} represents the between subclasses matrix.

M_W^{sub} and M_B^{sub} can be formulated as:

$$M_W^{sub} = \sum_{i=1}^{N_{sub}} \sum_{j=1}^{N_i} p_i (x_j - \mu_i)(x_j - \mu_i)^T, \quad (3.3)$$

where,

N_{sub} is the number of subclasses,

N_i is the number of samples that belong to the i^{th} subclass,

$p_i = \frac{N_i}{N}$ is the prior probability of the i^{th} subclass,

μ_i is the mean vector of the i^{th} subclass,

x_j is the training sample of the i^{th} subclass.

$$M_B^{sub} = \sum_{i=1}^{N_{sub}} (\mu_i - \mu)(\mu_i - \mu)^T, \quad (3.4)$$

where,

μ is the mean vector of the entire target class.

The primal optimization problem of the standard OSVM can be defined as (Bezerra et al., 2019; Shorman et al., 2020):

$$\min_{w, \xi, \rho} \frac{1}{2} w^T w + \frac{1}{vN} \sum_{i=1}^N \xi_i - \rho, \quad (3.5)$$

$$\text{s.t } w^T x_i \geq \rho - \xi_i, \xi_i \geq 0 \forall i = 1, \dots, N,$$

where,

w is the weight vector of the separating hyperplane,

$v \in [0,1]$ is a key parameter that acts as an upper bound for the fraction of outliers, and a lower bound for the fraction of SVs within the training dataset,

ξ_i is a slack variable,

ρ is a threshold or bias term.

After adding the within and between subclasses matrix M_{sub} of Eq. (3.2) to the primal optimization problem of Eq. (3.5), the SDOSVM primal optimization problem can be defined as:

$$\min_{w, \xi, \rho} \frac{1}{2} w^T M_{sub} w + \frac{1}{vN} \sum_{i=1}^N \xi_i - \rho, \quad (3.6)$$

$$\text{s.t } w^T M_{sub} w > 0, w^T x_i \geq \rho - \xi_i, \xi_i \geq 0 \forall i = 1, \dots, N.$$

The Langrangian dual function can be used to solve the SDOSVM primal optimization problem. First, the SDOSVM primal optimization problem is reformulated to the Langrangian dual function as follows:

$$L = \frac{1}{2} w^T M_{sub} w + \frac{1}{vN} \sum_{i=1}^N \xi_i - \rho - \sum_{i=1}^N \alpha_i (w^T x_i + \xi_i - \rho) - \sum_{i=1}^N \beta_i \xi_i, \quad (3.7)$$

where,

L is the Langrangian dual function,

α_i and β_i are the Langrangian multipliers.

Second, the Langrangian dual function is solved by deriving it relatively to the variables w , ξ , and ρ of the primal optimization problem, and the following is obtained:

$$w = M_{sub}^{-1} \sum_{i=1}^N \alpha_i x_i. \quad (3.8)$$

$$\alpha_i = \frac{1}{vN} - \beta_i. \quad (3.9)$$

$$\sum_{i=1}^N \alpha_i = 1. \quad (3.10)$$

Third, Eq. (3.8), Eq. (3.9), and Eq. (3.10) are substituted in Eq. (3.7), and the SDOSVM dual optimization problem is obtained:

$$\min_{\alpha} W(\alpha, \rho) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j K_{sub}(x_i, x_j) + \rho(1 - \sum_{i=1}^N \alpha_i), \quad (3.11)$$

$$\text{s.t } 0 \leq \alpha_i \leq \frac{1}{vN}, \text{ and } \sum_{i=1}^N \alpha_i = 1 \forall i = 1, \dots, N,$$

where,

$K_{sub} = X^T M_{sub}^{-1} X$ is the resulting kernel matrix,

$X = [x_1, \dots, x_N]$ is the initial training matrix.

Nonlinear Expansion of SDOSVM

In order to distinguish between both the target and outlier classes using a nonlinear decision boundary, we need to project the classification from the input space R^I to R^h which is a higher dimension space such that $h > I$.

This can be achieved through the kernel trick where the inner product in Eq. (3.6) is substituted by the mapping kernel function ϕ such that each data point is mapped from the input space to the higher dimension space without altering any relations used in the input space.

Using the kernel trick, the SDOSVM primal optimization problem in Eq. (3.6) can be re-formulated as:

$$\min_{w, \xi, \rho} \frac{1}{2} w^T M_{sub}^\phi w + \frac{1}{vN} \sum_{i=1}^N \xi_i - \rho, \quad (3.12)$$

$$\text{s.t } w^T M_{sub}^\phi w > 0, w^T \phi(x_i) \geq \rho - \xi_i, \xi_i \geq 0 \forall i = 1, \dots, N,$$

where,

$\phi(x_i)$ is the higher dimension class to which the target class is mapped using the mapping kernel function ϕ .

M_W^{sub} and M_B^{sub} can also be re-formulated as:

$$M_W^{sub\phi} = \sum_{i=1}^{N_{sub}} \sum_{j=1}^{N_i} p_i (\phi(x_j) - \mu_i^\phi) (\phi(x_j) - \mu_i^\phi)^T, \quad (3.13)$$

$$M_B^{sub\phi} = \sum_{i=1}^{N_{sub}} (\mu_i^\phi - \mu^\phi) (\mu_i^\phi - \mu^\phi)^T, \quad (3.14)$$

where,

μ_i^ϕ is the mean vector of the i^{th} subclass in the higher dimension space,

μ^ϕ is the mean vector of the entire target class in the higher dimension space.

Since M_{sub}^ϕ will be used for the nonlinear classification problem, and in order to compute it easily, the M_{sub} will be decomposed using the singular value decomposition as:

$$M_{sub} = V^T D V, \quad (3.15)$$

where,

V is the orthonormal matrix that contains the eigenvectors of M_{sub} ,

D is the diagonal matrix that contains the computed eigenvalues of M_{sub} .

After decomposing M_{sub} , the resulting matrices can be used for the following transformation:

$$Q = D^{-\frac{1}{2}} V. \quad (3.16)$$

Using the transformation in Eq. (3.16), each sample x_i can be transformed to $x_i^Q = Q x_i$. As a result the following formula is generated:

$$x_i^{Q^T} x_j^Q = x_i^T Q^T Q x_j = x_i^T M_{sub}^{-1} x_j. \quad (3.17)$$

Eq. (3.17) is applied to the entire training dataset X , and the result is:

$$X^Q = [x_1^Q, \dots, x_N^Q] = [Q x_1, \dots, Q x_N] = Q X.$$

Using the above formula, the following is calculated:

$$X^Q X^Q = X^T Q^T Q X = X^T M_{sub}^{-1} X. \quad (3.18)$$

As a result, the Radial Basis Function (RBF) kernel function of the SDOSVM can be formulated as:

$$K_{M_{sub}-RBF}(x_i, x_j) = e^{\left(\frac{-g(x_i, x_j)}{2\sigma^2}\right)}, \quad (3.19)$$

where,

$$g(x_i, x_j) = (x_i - x_j)^T M_{sub}^{-1} (x_i - x_j).$$

After the SDOSVM classifier is constructed, the test phase is performed to classify x , an unseen data sample, through the following classification function:

$$f(x) = \sum_{j=1}^{N_{sv}} \alpha_j K_{M_{sub}-RBF}(x_j, x) + \rho, \quad (3.20)$$

where,

N_{sv} is the number of SVs, x_j , acquired during the training step.

3.3. Proposed Algorithms

In this section, the proposed SDOSVM and the SDOSVM intrusion detection algorithms are summarized in Algorithm 1 and Algorithm 2, respectively.

3.4. Experimentation

In this section, the experimentation methodology of the proposed SDOSVM intrusion detection system is presented. First, a description of the benchmark datasets used for experimentation is provided. Second, the performance metrics used for validating the proposed system are described briefly. Third, the experimental

protocol is described in detail. Finally, a brief description of relevant ML one-class classifiers that are compared with the proposed approach is provided.

Algorithm 1 The SDOSVM Algorithm

Inputs: $X = \{x_i\}_{i=1}^N$ and v

Training Step:

1. Compute the within subclass matrix (M_W^{sub}) using Eq. (3.3).
2. Compute the between subclasses matrix (M_B^{sub}) using Eq. (3.4).
3. Compute the RBF kernel matrix ($K_{M_{sub}-RBF}$) using Eq. (3.19).
4. Solve the SDOSVM optimization problem of Eq. (3.11) to find $\alpha_i, i \in [1, \dots, s]$ (Eq. (3.9) and Eq. (3.10)), and ρ .

Outputs: $\alpha_i, i \in [1, \dots, s]$, and ρ .

Testing Step:

Compute the classification function using Eq. (3.20).

Output: Predicted class label vector

Algorithm 2 The SDOSVM Intrusion Detection Algorithm

1. **Feature Mapping:** Transform values of categorical features to numeric ones.
 2. **Feature Selection:** Select best features for the classification problem.
 3. **Clustering:** Find the optimal number of clusters for the target class.
 4. **Feature Normalization:** Normalization of features using Eq. (3.1).
 5. **Subclass Matrix Calculation:** Compute subclass matrix M_{sub} using Eq. (3.2), Eq. (3.3), and Eq. (3.4).
 6. **Train the Model:** Train the SDOSVM model with RBF kernel using Eq. (3.19) on the target class by solving the primal optimization problem in Eq. (3.12).
 7. **Testing the Model:** Test the SDOSVM model using the classification function in Eq. (3.20) to distinguish between
-

normal and attack data.

8. **Assessing Performance:** Compute the AUC, the number of SVs, and the training time.
-

3.4.1. Datasets Used

In this subsection, we describe in detail the datasets used for the experimentations as illustrated in Table 3.1.

The TON_IoT dataset includes four diverse data sources that were collected from Telemetry datasets of IoT and Industrial Internet of Things (IIoT) sensors, Operating systems datasets of Windows 7 and 10 as well as Ubuntu 14.04 and 18.04 TLS, and Network traffic datasets. The dataset was generated using a pragmatic and large-scale network designed in the Cyber Range and IoT Labs of UNSW Canberra. More details about the dataset can be found in (Booij et al., 2021; Moustafa, 2019a, 2021), and the directories of the dataset can be found in (Moustafa, 2019b).

For easy handling of the TON_IoT dataset, the author created *Train_Test_datasets* that consists of four datasets for evaluating the performance of cyber security application-based Artificial Intelligence (AI) and ML algorithms. For the experiments the *Train_Test_Network_dataset* is used that consists of 461,043 data samples divided to 300,000 normal and 161,043 attack data samples. The attacks included in the network dataset are Backdoor, Distributed Denial of Service (DDoS), Denial of Service (DoS), Injection, Man in The Middle (MiTM), Password Cracking, Ransomware, Scanning, and Cross-Site Scripting (XSS) (Moustafa, 2021).

In the experiments five subsets are used for each of the target and outlier classes. Both target class and outlier class subsets were created using a random selection of 2500 normal and attack data samples,

respectively, for each of the five subsets from the *Train_Test_Network_dataset*.

Table (3.1): Real-world IoT datasets used for the experimentation

Reference	Dataset	Number of Target Class Instances per Subset	Number of Outlier Class Instances per Subset	Number of Best Features per Subset
Moustafa (2021)	TON_IoT (5 subsets)	2500	2500	15

3.4.2. Performance Metrics

In this subsection, the performance metrics used for validating the proposed SDOSVM intrusion detection system are described briefly. The response of each performance metrics to the proposed SDOSVM intrusion detection system is described in detail in section 4.3.

ROC Curve and AUC

The ROC curve is a visual evaluation metric for the performance of classifiers that measures the trade-off between FPR and TPR (Hady et al., 2020).

The ROC curve is a powerful metric due to its dependency on the FPR and TPR without taking into account the number of training or testing data samples (Besrour and Ksantini, 2019).

The AUC metric corresponds to the classifier's performance. It is generated by the ROC curve, and is summarized as a float number in the range [0,1] (Hady et al., 2020; Kumar et al., 2021).

P-value

Paired-sample t-test probability-values (p-values) were computed per subset between the proposed SDOSVM approach and each of the relevant ML one-class classifiers using the AUC values.

The p-values quantify the probability of whether the paired AUC values for the same subset are the same or not in order to elicit the improvement in classification performance by the proposed SDOSVM approach over the relevant ML one-class classifiers.

FPR and TPR

The FPR is the ratio of the number of outlier class samples misclassified as target class samples to the total number of outlier class samples in the testing dataset:

$$FPR = \frac{FP}{FP+TN}, \quad (3.21)$$

where,

FP is the number of outlier class samples misclassified as target class samples,

TN is the number of outlier class samples classified correctly.

The TPR is the ratio of the number of target class samples classified correctly to the total number of target class samples in the testing dataset:

$$TPR = \frac{TP}{TP+FN}, \quad (3.22)$$

where,

TP is the number of target class samples classified correctly,

FN is the number of target class samples misclassified as outlier class samples.

Number of SVs and Training Time

Both the number of SVs and training time have been computed for the proposed SDOSVM approach and relevant ML one-class classifiers to demonstrate that the proposed SDOSVM approach outperforms these classifiers in terms of computational complexity.

3.4.3. Experimental Protocol

This subsection presents the experimental protocol used to assess the performance of the proposed SDOSVM intrusion detection system in order to show the advantage of the SDOSVM approach over the standard OSVM and other relevant ML one-class classifiers.

First, feature mapping was done to transform values in the categorical features of the experimental datasets to numeric ones. The feature mapping was performed by the author on the TON_IoT dataset using the *unclass* function of the R programming language (Moustafa, 2020).

Second, feature selection was performed to select the best features for the classification problem. For the TON_IoT dataset, the *SelectKBest* function was used from the *sklearn.feature_selection* library of the python programming language to select the best features from the total 43 features of the dataset that gave best results for the proposed intrusion detection system. As shown in Figure 3.2, the TON_IoT features are sorted from those with the highest scores to those with the lowest scores. Based on the generated scores, the top 15 best features were selected for experimentation, and are illustrated in Table 3.2. The

description of these features can be found in (Booij et al., 2021; Moustafa, 2021).

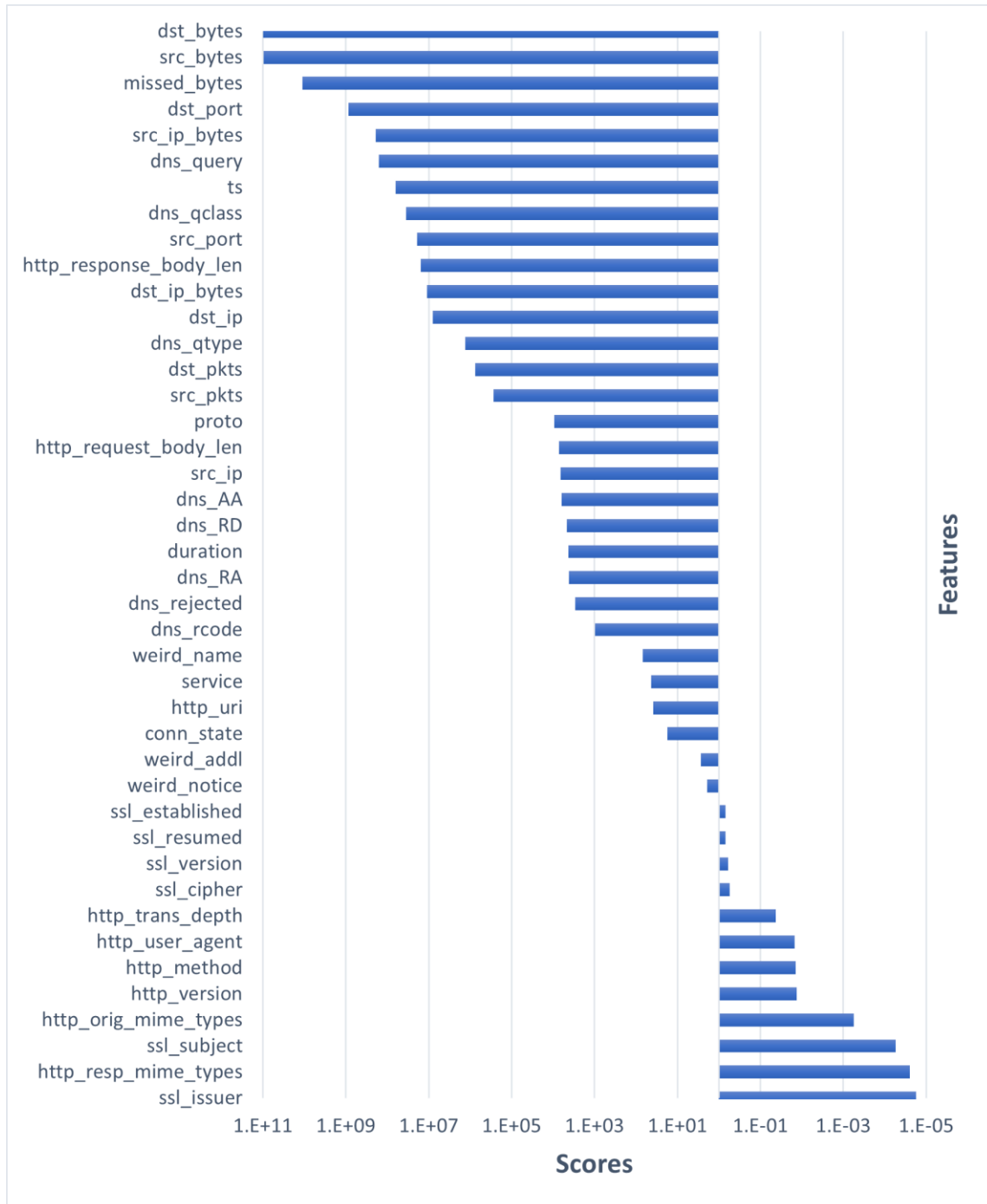


Figure (3.2): Features' Scores of the TON_IoT dataset

Table (3.2): TON_IoT selected best features for the experimentation

Feature	Type of Feature
dst_bytes	Connection
src_bytes	Connection
missed_bytes	Connection
dst_port	Connection
src_ip_bytes	Statistical
dns_query	Domain Name System
ts	Connection
dns_qclass	Domain Name System
src_port	Connection
http_response_body_len	HyperText Transfer Protocol
dst_ip_bytes	Statistical
dst_ip	Connection
dns_qtype	Domain Name System
dst_pkts	Statistical
src_pkts	Statistical

Third, in order to find the optimal number of clusters for the experimental datasets, the Fuzzy Maximum Likelihood Estimation (FMLE) algorithm and the validity index V_{sc} proposed by Bouguessa et al. (2006) were used.

The FMLE algorithm was used due to its efficiency in handling variations in clusters' shapes and densities. The V_{sc} validity index was used in order to optimize the combined function of compactness and separation as well as its efficiency in handling variations in clusters' overlap, shapes, and densities (Bouguessa et al., 2006).

The validity index V_{sc} was calculated for the experimental datasets using ten different c -partitions where the minimum value of c -partitions (C_{min}), and the maximum value of c -partitions (C_{max}) were predefined as the values 2 and 10, respectively. An optimal number of clusters is obtained when V_{sc} is maximized (Bouguessa et al., 2006).

For the TON_IoT dataset, the optimal number of clusters obtained for each subset is shown in Table 3.3.

Table (3.3): Optimal number of clusters obtained per subset for the TON_IoT dataset

Subset	Optimal Number of Clusters
1	4
2	5
3	4
4	4
5	4

The K-means clustering algorithm was used to generate the subclasses for the target class of the experimental datasets based on the optimal number of clusters obtained.

Fourth, the experimental target class datasets and their subclasses were normalized in order to compute the subclasses matrix M_{sub} .

Fifth, a 10-fold cross-validation method was used for the experimental normalized target class datasets where each dataset was randomly divided into 10 subsets such that each time 9 subsets were used for training to optimize the proposed model, and one subset was added to an experimental normalized outlier class dataset for testing to validate the proposed model as shown in Figure 3.3. For each fold the performance metrics, i.e. the AUC, the number of SVs generated by the model, and the training time were computed.

The 10-fold cross-validation method was used to show explicitly unbiased and reliable results for the proposed SDOSVM approach, and select the optimal values for the SDOSVM hyperparameters ν of Eq. (3.12) and σ of Eq. (3.19) over the dataset.

Finally, the average AUC, the average number of SVs, and the average training time of the 10-fold cross-validation method were

computed for the SDOSVM to assess its performance against the standard OSVM, and the other relevant ML one-class classifiers.

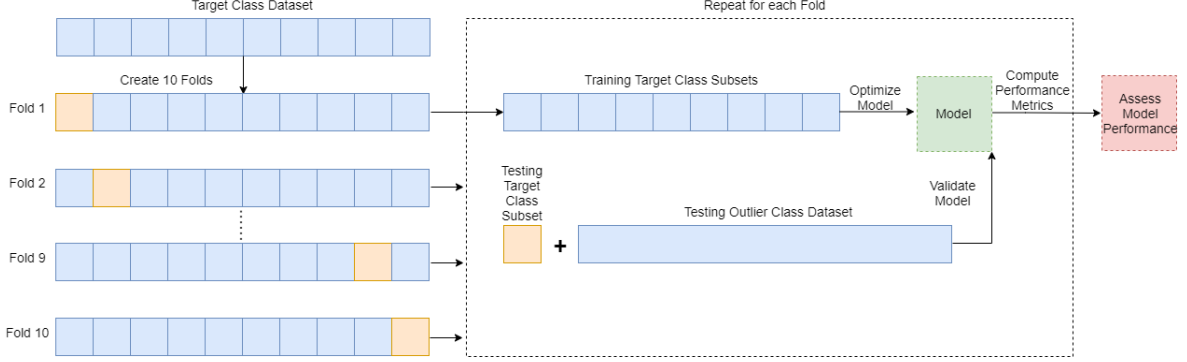


Figure (3.3): 10-fold cross-validation method

For implementation, Google Colaboratory, MATLAB R2021a, R i386 3.6.3, and Spyder 5 were used. The Clustering Toolbox (Balasko et al., 2014; Abonyi, 2021) was used during implementation for finding the optimal number of clusters for the target class. The SVM and Kernel Methods Matlab Toolbox (Canu et al., 2005) and the Data Description Toolbox (Tax, 2013) were also used for implementing the one-class classifiers. The github code created by Chalapathy (2018) was used for implementing the One-Class-Neural Network (OC-NN) classifier.

3.4.4. Classifiers

In this subsection, we describe briefly the ML one-class classifiers used for the comparative analysis of the SDOSVM approach.

The proposed SDOSVM approach is compared against the following five state-of-the-art ML one-class classifiers:

1. **Gaussian:** The target class is modelled in this classifier by utilizing a single gaussian density. The mean and covariance matrix are estimated using maximum likelihood. In case of high

dimensional data, a regularization term is added to the covariance matrix. There is no need to optimize parameters for this classifier (Khan et al., 2014).

2. **Support Vector Data Description (SVDD):** This classifier is inspired by the SVM. It fits a spherical boundary which has the minimal volume (or radius) around the target class, and this boundary is constructed using a set of SVs describing it. The implementation of this classifier used in experimentation has the RBF kernel hard-coded in order to ease implementation and evaluation (Tax and Duin, 1999).
3. **OSVM:** The SDOSVM was built upon the OSVM as described in section 3.2.2. The OSVM differs from the SVDD as the SVDD fits the hypersphere with minimal radius around the target class, where as the OSVM tries to find the hyperplane with maximum margin to separate the target class from the origin (Khan et al., 2014).
4. **Mahalanobis One-class Support Vector Machine (MOSVM):** This classifier is an extension of the OSVM. The Mahalanobis distance is used instead of the Euclidean distance as the distance-metric in order to utilize the data covariance as a matrix (Tsang et al., 2006).
5. **OC-NN:** This classifier can be considered as a neural architecture utilizing the OSVM equivalent loss function. It utilizes a simple feed forward network with a single hidden layer having linear or sigmoid activation, and one output node. The linear activation is used in the experimentation (Chalapathy et al., 2018).

3.5. Summary

This chapter gave a detailed description of the steps involved in the proposed SDOSVM intrusion detection system which include feature pre-processing, and classification using the proposed SDOSVM algorithm. The datasets and performance metrics used for validating the proposed system were described as well as the experimental protocol, and the implementation tools. The relevant ML one-class classifiers to which the proposed SDOSVM approach is compared were also described briefly.

Chapter 4

Results and Discussion

4.1. Overview

4.2. Limitations of Proposed Model

4.3. Discussion

4.4. Summary

Chapter Four

Results and Discussion

4.1. Overview

The proposed SDOSVM intrusion detection system was evaluated using the real-world benchmark IoT dataset, TON_IoT. The results obtained were compared to other relevant ML one-class classifiers that used the same evaluation benchmark dataset as the proposed algorithm.

The proposed SDOSVM algorithm was programmed and evaluated using MATLAB R2021a installed on a personal laptop with the following specifications:

1. **CPU:** Intel Core i7-8550U CPU 2.00 GHz
2. **RAM:** 16 GB
3. **OS:** Windows 10 Home

The Gaussian and SVDD classifiers from the Data Description Toolbox were run using MATLAB R2021a. The OSVM and MOSVM classifiers from the SVM and Kernel Methods Matlab Toolbox were also run using MATLAB R2021a. The OC-NN classifier was run using the github code generated by Chalapathy (2018) on Spyder 5. The results of running these classifiers were obtained for comparison with the performance of the proposed SDOSVM.

The average results of the ten folds obtained from the conducted experiments on the five TON_IoT subsets for the proposed SDOSVM and the relevant ML one-class classifiers are summarised in Tables 4.1, 4.2, 4.3, and Figure 4.1. These results are discussed in section 4.3.

4.2. Limitations of the Proposed Model

This section presents the limitations of the proposed SDOSVM intrusion detection system. First, the proposed SDOSVM algorithm is a batch model. The batch learning model is built using data at rest, i.e. an entire dataset available at a specific period in time. When the model is evaluated using a test set, and the results reveal that it performs well, the model is then shipped for production, and the learning process ends. The model is re-trained from scratch using both previous and new data samples when new data emerges, and the model is required to be updated. This is considered a drawback for batch learning models because it will not be suitable for training a model with new data that continuously emerges from a source as in the IoHT environments (Bisong, 2019).

To overcome such a drawback for batch learning models, the incremental or online learning is a preferable choice. An incremental learning model is built using flows of data streams which can be either individual samples in the dataset or small batch sizes. The incremental learning is considered ideal for training a model with new data that continuously emerges from a source, and where there is a need to build a prediction model using real-time data samples (Bisong, 2019). Therefore, the incremental learning model is suitable for the IoHT environments.

Second, the proposed SDOSVM intrusion detection algorithm uses the K-means clustering algorithm to generate the subclasses for the target class of the experimental datasets based on the optimal number of clusters obtained. Classical clustering algorithms, such as K-means, grasp similarities between data points using distance-based metrics, and therefore, are considered shallow models. Such models fail to achieve reliable discriminative power, i.e. using distance-based metrics is not adequate for the discovery of representative similarities between data

points. Moreover, the quadratic complexity of some of the shallow models is considered a restriction for processing large-scale datasets similar to those generated by the IoHT environments. And since data clustering is done in raw data space, the shallow models cannot cope with the curse of dimensionality (Mrabah et al., 2020).

Therefore, deep clustering models are recommended for subclasses generation because of several reasons. First, large-scale datasets like the IoHT datasets can be handled using the multi-layer architecture of these models because of the mini-batch Stochastic Gradient Descent (SGD) that gives the neural networks the capacity to evade both time and memory bottlenecks. Second, the ability of neural networks to project data from high-dimensional space to low-dimensional space. Finally, the neural networks can discover representative similarities between data points, and therefore, achieve a better discriminative power than shallow models (Mrabah et al., 2020).

4.3. Discussion

This section presents a detailed discussion of the obtained experimentation results. The improvement in performance by the proposed SDOSVM approach over the relevant ML one-class classifiers is validated using the performance metrics described in subsection 3.4.2.

4.3.1. ROC Curve and AUC

As we can see from the obtained average AUC results in Table 4.1, the SDOSVM provides the best AUC results for all five subsets than the other one-class SVM-based classifiers which are the MOSVM, the Standard OSVM, and the SVDD. This can be explained due to the fact that the SDOSVM considers subclasses in the target class thus

minimizing the data dispersion within and between subclasses, and improving the discriminative power and classification performance.

Table (4.1): Average AUC per subset for the TON_IoT dataset (best method in bold)

Subsets	1		2		3		4		5	
	AUC	1-p-value	AUC	1-p-value	AUC	1-p-value	AUC	1-p-value	AUC	1-p-value
Gaussian	75.96	100	79.13	100	79.86	100	76.14	100	84.36	100
SVDD	94.86	100	93.90	100	93.59	100	94.20	100	94.12	100
OSVM	92.86	100	92.43	100	92.14	100	92.98	100	92.95	100
MOSVM	92.92	100	92.73	100	92.79	100	93.10	100	92.74	100
OC-NN	75.95	100	76.19	100	78.11	100	77.15	100	76.08	100
SDOSVM	97.14	-	99.49	-	99.92	-	99.25	-	96.91	-

Also, the Gaussian classifier is outperformed by the one-class SVM-based classifiers (MOSVM, OSVM, SVDD, and SDOSVM). This can be explained by the limitation of the Gaussian classifier as it assumes that the underlying distribution of the training data is Gaussian, and this case is not always true for real-world datasets. The one-class SVM-based classifiers are free from such an assumption, and hence these classifiers generated better AUC results in all five subsets (Khan et al., 2014).

The SDOSVM also outperforms the OC-NN in terms of AUC values for all five subsets. This can be explained due to the fact that the objective function for the OC-NN is non-convex, and hence, deducing the model parameters will not reach to a global optima (Chalapathy et al., 2018).

Moreover, the average ROC curves for the SDOSVM and the aforementioned one-class classifiers per subset are illustrated in Figure 4.1. As observed, the ROC curves of the SDOSVM are the best because it achieved the best trade-off between the average FPR and TPR which resulted in the best AUC for all five subsets, and therefore, it has the best classification performance over the other one-class classifiers.

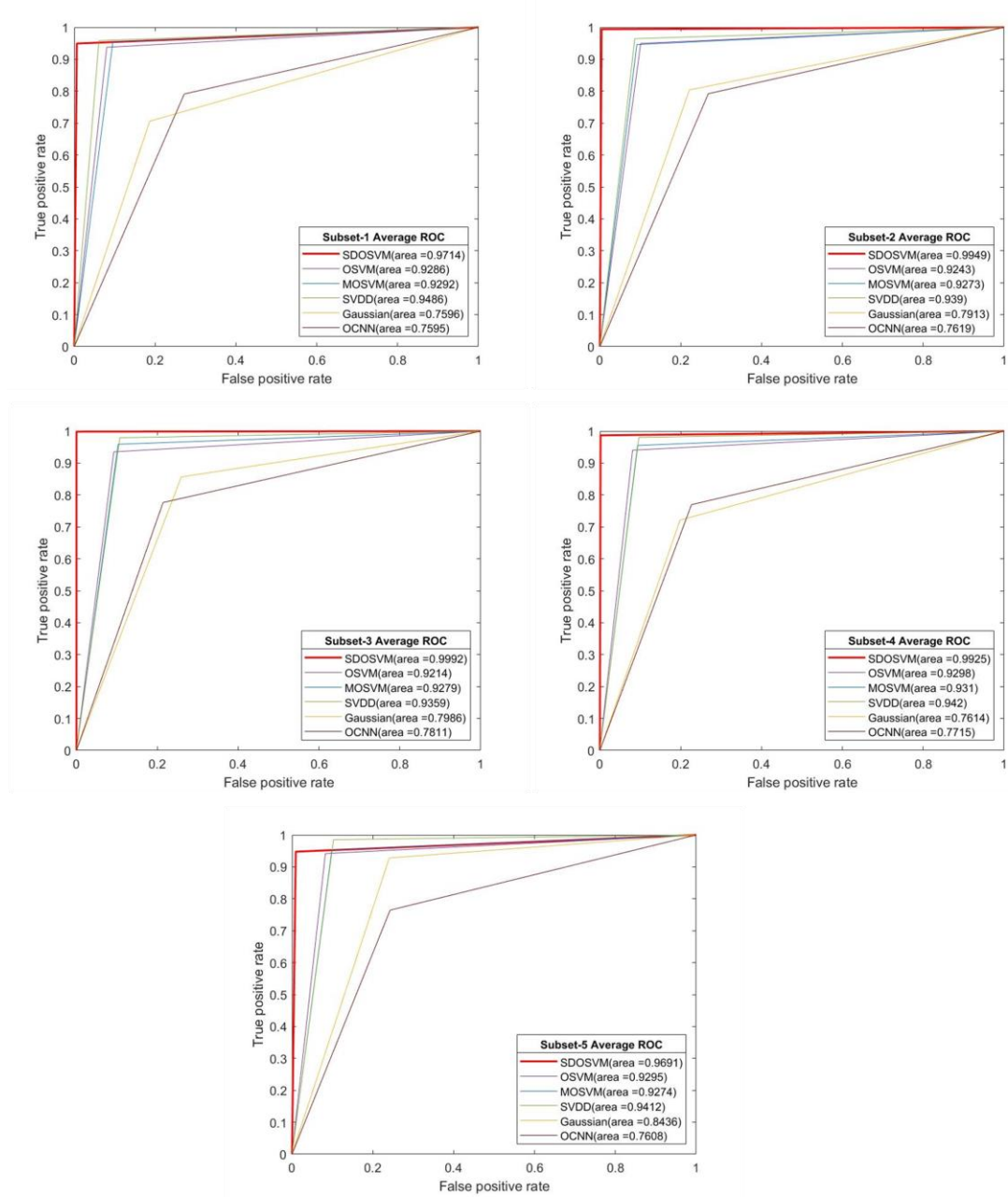


Figure (4.1): Average ROC curves of one-class classifiers per subset for the TON_IoT dataset

4.3.2. P-value

As observed from Table 4.1, all the (1-p-value) are approximately 100% which indicates a high probability that the paired AUC values of the SDOSVM and each of the one-class classifiers are not the same for each of the five subsets, and therefore, the SDOSVM provides statistically

considerable improvement in performance over the other one-class classifiers.

4.3.3. FPR and TPR

In Table 4.2, the average FPR and TPR for the SDOSVM and the other one-class classifiers are presented per subset. As can be seen, the SDOSVM provides the lowest value of FPR for all subsets, and the highest TPR for most subsets.

The SVDD provides the highest TPR for both Subset-1 and Subset-5 but is considerably outperformed by the SDOSVM in terms of FPR. This can be explained due to the fact that the SDOSVM considers subclasses in the target class in order to improve the discriminative power and classification performance.

4.3.4. Number of SVs and Training Time

All the compared SVM-based classifiers including the proposed SDOSVM scale with a complexity of $O(N^2)$, where N represents the number of training samples. However, as seen from the obtained average SVs and training time results in Table 4.3, the SDOSVM has the best average SVs and training time compared to the other SVM-based classifiers. This can be explained due to the fact that the SDOSVM minimizes the data dispersion while projecting data which reduces the number of SVs, and hence, results in lower computational complexity as indicated by the obtained average training time.

In particular the MOSVM provides the highest training time, since it incorporates the inversion of both kernel matrix and covariance matrix, whereas for the SDOSVM the inversion of the dispersion matrix is performed once in the input space (refer to the equations from 3.15 to 3.19).

Table (4.2): Average FPR and TPR per subset for the TON_IoT dataset

Subsets		Gaussian	SVDD	OSVM	MOSVM	OC-NN	SDOSVM
1	Average FPR	18.67	6.11	8.04	9.56	27.22	0.64
	Average TPR	70.60	95.84	93.76	95.40	79.12	94.92
2	Average FPR	22.14	8.63	10.10	9.13	26.82	0.42
	Average TPR	80.40	96.44	94.96	94.60	79.20	99.40
3	Average FPR	25.93	10.74	9.24	10.30	21.42	0.00
	Average TPR	85.64	97.92	93.52	95.88	77.64	99.84
4	Average FPR	19.84	9.64	8.09	9.32	22.66	0.17
	Average TPR	72.12	98.04	94.04	95.52	76.96	98.68
5	Average FPR	24.08	10.28	8.26	9.88	24.27	0.94
	Average TPR	92.80	98.52	94.16	94.36	76.44	94.76

The SDOSVM also outperforms the OC-NN classifier as seen in Table 4.3 because of the high number of epochs, which is 100 for the experimentation, despite the use of backward propagation instead of the forward propagation.

Finally, as seen from Table 4.3, the Gaussian classifier achieves better training time than the SDOSVM and the other one-class classifiers which is approximately 0.01 seconds for all five subsets. This can be explained due to the fact that the Gaussian is a non-iterative algorithm unlike the SDOSVM and the other one-class classifiers that use the gradient descent algorithm, and therefore, are iterative algorithms.

Table (4.3): Average number of SVs and training time per subset for the TON_IoT dataset

Subsets		Gaussian	SVDD	OSVM	MOSVM	OC-NN	SDOSVM
1	Average SVs	-	2250	171	938	-	121
	Average Training Time (s)	0.01	17.06	1.40	21.31	3.45	0.98
2	Average SVs	-	2250	124	974	-	19
	Average Training Time (s)	0.01	16.89	0.99	18.99	3.89	0.35
3	Average SVs	-	2250	149	969	-	5
	Average Training Time (s)	0.01	18.41	1.18	22.47	3.28	0.30
4	Average SVs	-	2250	151	901	-	38
	Average Training Time (s)	0.01	18.82	1.05	20.83	3.68	0.42
5	Average SVs	-	2250	137	968	-	116
	Average Training Time (s)	0.01	17.57	0.94	23.61	3.66	0.75

4.4. Summary

This chapter presented a detailed discussion of the experimentation results obtained when the proposed SDOSVM algorithm was evaluated using real-world IoT benchmark datasets, and compared with other relevant ML one-class classifiers. The results showed that the proposed SDOSVM classifier outperformed the other one-class classifiers in terms of AUC, number of SVs, and training time. The limitations of the proposed SDOSVM intrusion detection system were also discussed which includes the restriction of handling dynamic, high-dimensional, and large-scale data similar to that generated by the IoHT environment because the

SDOSVM is a batch learning model, and uses the K-means algorithm for clustering data to subclasses.

Chapter 5

Conclusion and Recommendations

5.1. Overview

5.2. Conclusion

5.3. Recommendations and Future Work

Chapter Five

Conclusion and Recommendations

5.1. Overview

Several state-of-the-art ML intrusion detection systems for the IoT and IoHT were discussed in the literature review based on a thematic taxonomy which included data sources (host-based and network-based), intrusion detection methods (signature-based, anomaly-based, and hybrid), and classifiers (one-class, binary class, and multi-class) along with state-of-the-art subclasses intrusion detection solutions. Based on the conducted literature review, the contribution of this research work was presented as a proposed subclasses one-class anomaly-based network intrusion detection system for the IoHT, namely SDOSVM, which is a novel variation of the standard OSVM that considers the subclasses in the IoHT normal data class in order to minimize data dispersion within and between subclasses thus improving the discriminative power and classification performance.

The proposed SDOSVM intrusion detection system was evaluated using a real-world IoT benchmark dataset, TON_IoT, that includes data from heterogeneous data sources, and compared with state-of-the-art ML one-class classifiers which included MOSVM, OSVM, SVDD, Gaussian, and OC-NN. The main findings of this research work will be discussed in the next section.

5.2. Conclusion

Experimentation results based on the real-world IoT dataset, TON_IoT, showed that the proposed SDOSVM approach outperformed the standard OSVM and other relevant ML one-class classifiers in terms of AUC, where the AUC scores achieved by the proposed approach are above 96%.

Also, the proposed approach outperformed the one-class SVM-based classifiers which are the OSVM, MOSVM, and SVDD in terms of training time, which did not exceed one second, due to its reliance on a low number of SVs thus resulting in lower computational complexity.

Finally, the proposed approach achieved the best results than the one-class classifiers for the FPR, which did not exceed 1%, and the TPR in most evaluation subsets of the TON_IoT dataset which was above 98.5%.

Based on the obtained results, the proposed SDOSVM approach can be considered suitable for intrusion detection in the IoHT environments as these environments are heterogeneous, i.e. the definition of normal behavior can differ from one system to another, and resource-constraint.

5.3. Recommendations and Future Work

The proposed SDOSVM algorithm used the subclasses dispersion information to improve the classification performance of the IoHT one-class anomaly-based network intrusion detection system, which was not tackled before in the literature to the best of knowledge.

Having considered the subclasses dispersion information for developing an intrusion detection system for the IoHT, the proposed SDOSVM intrusion detection system still suffers limitations of being a batch learning model, and using classical clustering algorithms, namely

K-means, for generating subclasses thus hindering the proposed system to handle dynamic, high-dimensional, and large-scale data similar to that generated by the IoHT.

Therefore, for future work it is recommended to modify the proposed SDOSVM intrusion detection system to an incremental intrusion detection system for the IoHT. The incremental version should be able to handle sequential obtained data, and large-scale datasets without any degradation in performance, and manage the learning process at the beginning with few available data samples.

Also, for future work it is recommended to use deep clustering models to generate subclasses from the high-dimensional and large-scale data of the IoHT.

References

- Abonyi, J. (2021). Clustering Toolbox. *Clustering Toolbox*. Retrieved from <https://www.mathworks.com/matlabcentral/fileexchange/7486-clustering-toolbox>
- Al-Garadi, M. A., Mohamed, A., Al-Ali, A. K., Du, X., Ali, I., & Guizani, M. (2020). A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security. *IEEE Communications Surveys Tutorials*, 22, 1646-1685.
- AL-Hawawreh, M., Moustafa, N., & Sitnikova, E. (2018). Identification of malicious activities in industrial internet of things based on deep learning models. *Journal of Information Security and Applications*, 41, 1-11.
- Alrashdi, I., Alqazzaz, A., Alharthi, R., Aloufi, E., Zohdy, M. A., & Ming, H. (2019). FBAD: Fog-based Attack Detection for IoT Healthcare in Smart Cities. *2019 IEEE 10th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, (pp. 515-522).
- Bagaa, M., Taleb, T., Bernabe, J. B., & Skarmeta, A. (2020). A Machine Learning Security Framework for Iot Systems. *IEEE Access*, 8, 114066-114077.
- Bakhtiar, F. A., Pramukantoro, E. S., & Nihri, H. (2019). A Lightweight IDS Based on J48 Algorithm for Detecting DoS Attacks on IoT Middleware. *2019 IEEE 1st Global Conference on Life Sciences and Technologies (LifeTech)*, (pp. 41-42).
- Balasko, B., Abonyi, J., & Feil, B. (2014, July). Fuzzy Clustering and Data Analysis Toolbox for Use With Matlab.
- Besrou, A., & Ksantini, R. (2019). Incremental Subclass Support Vector Machine. *International Journal on Artificial Intelligence Tools*, 28, 1950020.
- Bezerra, V. H., da Costa, V. G., Barbon Junior, S., Miani, R. S., & Zarpelão, B. B. (2019). IoTDS: A One-Class Classification Approach to Detect Botnets in Internet of Things Devices. *Sensors*, 19.
- Bisong, E. (2019). Batch vs. Online Learning. In S. McDermott, L. Berendson, & R. Fernando (Eds.), *Building Machine Learning and Deep Learning Models on Google Cloud Platform* (pp. 199-201). Berkeley, CA: Apress.
- Booij, T. M., Chiscop, I., Meeuwissen, E., Moustafa, N., & den Hartog, F. T. (2021). ToN_IoT: The Role of Heterogeneity and the Need for Standardization of Features and Attack Types in IoT Network Intrusion Datasets. *IEEE Internet of Things Journal*, 1-1.
- Bouguessa, M., Wang, S., & Sun, H. (2006). An objective approach to cluster validation. *Pattern Recognition Letters*, 27, 1419-1430.

- Canu, S., Grandvalet, Y., Guigue, V., & Rakotomamonjy, A. (2005). SVM and Kernel Methods Matlab Toolbox. *SVM and Kernel Methods Matlab Toolbox*. Retrieved from <http://asi.insa-rouen.fr/enseignants/arakoto/toolbox/>
- Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., & Faruki, P. (2019). Network Intrusion Detection for IoT Security Based on Learning Techniques. *IEEE Communications Surveys Tutorials*, 21, 2671-2701.
- Chalapathy, R. (2018). oc-nn. *oc-nn*. Retrieved from <https://github.com/wzmsltw/oc-nn>
- Chalapathy, R., Menon, A. K., & Chawla, S. (2018). Anomaly Detection using One-Class Neural Networks. *CoRR*, abs/1802.06360. Retrieved from <http://arxiv.org/abs/1802.06360>
- Ciklabakkal, E., Donmez, A., Erdemir, M., Suren, E., Yilmaz, M. K., & Angin, P. (2019). ARTEMIS: An Intrusion Detection System for MQTT Attacks in Internet of Things. *2019 38th Symposium on Reliable Distributed Systems (SRDS)*, (pp. 369-3692).
- Efstathopoulos, G., Grammatikis, P. R., Sarigiannidis, P., Argyriou, V., Sarigiannidis, A., Stamatakis, K., . . . Athanasopoulos, S. K. (2019). Operational Data Based Intrusion Detection System for Smart Grid. *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, (pp. 1-6).
- Eskandari, M., Janjua, Z. H., Vecchio, M., & Antonelli, F. (2020). Passban IDS: An Intelligent Anomaly-Based Intrusion Detection System for IoT Edge Devices. *IEEE Internet of Things Journal*, 7, 6882-6897.
- Ghosal, P., Das, D., & Das, I. (2018). Extensive Survey on Cloud-based IoT-Healthcare and Security using Machine Learning. *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, (pp. 1-5).
- Hady, A. A., Ghubaish, A., Salman, T., Unal, D., & Jain, R. (2020). Intrusion Detection System for Healthcare Systems Using Medical and Network Data: A Comparison Study. *IEEE Access*, 8, 106576-106584.
- Kamble, P., & Gawade, A. (2019). Digitalization of Healthcare with IoT and Cryptographic Encryption against DOS Attacks. *2019 International Conference on contemporary Computing and Informatics (IC3I)*, (pp. 69-73).
- Khan, N. M., Ksantini, R., Ahmad, I. S., & Guan, L. (2014). Covariance-guided One-Class Support Vector Machine. *Pattern Recognition*, 47, 2165-2177.
- Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., & Alazab, A. (2019). A Novel Ensemble of Hybrid Intrusion Detection System for Detecting Internet of Things Attacks. *Electronics*, 8.

- Kumar, P., Gupta, G. P., & Tripathi, R. (2021). An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks. *Computer Communications*, 166, 110-124.
- Liaqat, S., Akhunzada, A., Shaikh, F. S., Giannetsos, A., & Jan, M. A. (2020). SDN orchestration to combat evolving cyber threats in Internet of Medical Things (IoMT). *Computer Communications*, 160, 697-705.
- Moustafa, N. (2019a). New Generations of Internet of Things Datasets for Cybersecurity Applications based Machine Learning: TON_IoT Datasets. *Proceedings of the eResearch Australasia Conference*.
- Moustafa, N. (2019b). The TON_IoT Datasets. *The TON_IoT Datasets*. Retrieved from <https://cloudstor.aarnet.edu.au/plus/s/ds5zW91vdgjEj9i>
- Moustafa, N. (2020). TON_IoT Network dataset. *TON_IoT Network dataset*. Retrieved from [https://github.com/Nour-Moustafa/TON_IoT Network dataset](https://github.com/Nour-Moustafa/TON_IoT_Network_dataset)
- Moustafa, N. (2021). A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. *Sustainable Cities and Society*, 72, 102994.
- Mrabah, N., Khan, N. M., Ksantini, R., & Lachiri, Z. (2020). Deep clustering with a Dynamic Autoencoder: From reconstruction towards centroids construction. *Neural Networks*, 130, 206-228.
- R.M., S. P., Maddikunta, P. K., Parimala, M., Koppu, S., Gadekallu, T. R., Chowdhary, C. L., & Alazab, M. (2020). An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture. *Computer Communications*, 160, 139-149.
- Said, A. M., Yahyaoui, A., Yaakoubi, F., & Abdellatif, T. (2020). Machine Learning Based Rank Attack Detection for Smart Hospital Infrastructure. In M. Jmaiel, M. Mokhtari, B. Abdulrazak, H. Aloulou, & S. Kallel (Ed.), *The Impact of Digital Technologies on Public Health in Developed and Developing Countries*, (pp. 28–40).
- Sheikhan, M., & Bostani, H. (2016). A hybrid intrusion detection architecture for Internet of things. *2016 8th International Symposium on Telecommunications (IST)*, (pp. 601-606).
- Shorman, A. A., Faris, H., & Aljarah, I. (2020). Unsupervised intelligent system based on one class support vector machine and Grey Wolf optimization for IoT botnet detection. *Journal of Ambient Intelligence and Humanized Computing*, 11, 2809–2825.
- Singh, P., Krishnamoorthy, S., Nayyar, A., Luhach, A. K., & Kaur, A. (2019). Soft-computing-based false alarm reduction for hierarchical data of intrusion detection system. *International Journal of Distributed Sensor Networks*, 15, 1550147719883132.

- Tax, D. (2013). Data Description Toolbox. *Data Description Toolbox*. Retrieved from <http://prtools.tudelft.nl/Guide/37Pages/software.html>
- Tax, D. M., & Duin, R. P. (1999). Support vector domain description. *Pattern Recognition Letters*, 20, 1191-1199.
- Thamilarasu, G., Odesile, A., & Hoang, A. (2020). An Intrusion Detection System for Internet of Medical Things. *IEEE Access*, 8, 181560-181576.
- Tsang, I. W., Kwok, J. T., & Li, S. (2006). Learning the Kernel in Mahalanobis One-Class Support Vector Machines. *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, (pp. 1169-1175).
- Yaacoub, J.-P. A., Noura, M., Noura, H. N., Salman, O., Yaacoub, E., Couturier, R., & Chehab, A. (2020). Securing internet of medical things systems: Limitations, issues and recommendations. *Future Generation Computer Systems*, 105, 581-606.
- Yahyaoui, A., Abdellatif, T., & Attia, R. (2019). Hierarchical anomaly based intrusion detection and localization in IoT. *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)*, (pp. 108-113).

المخلص

على الرغم من الفوائد الكبيرة التي قدمها إنترنت أشياء الرعاية الصحية للقطاع الطبي، إلا أن هناك مخاوف ومخاطر تتعلق بهذه الأنظمة والتي يمكن أن تؤخر نشرها على نطاق واسع لأنها تتعامل مع المعلومات الطبية الحساسة، والتي غالبًا ما تكون ضرورية لحياة المريض. بالإضافة إلى مخاوف ومخاطر إنترنت أشياء الرعاية الصحية، هناك قيود أمنية تشمل القيود الخاصة بالأجهزة، البرامج، والشبكات والتي تشكل تحديًا آمنًا لهذه الأنظمة. لذلك، يجب نشر الإجراءات الأمنية التي يمكنها التغلب على المخاوف، تخفيف المخاطر، وتلبية القيود الأمنية لإنترنت أشياء الرعاية الصحية. لهذه الأسباب، تم اقتراح نظام لكشف التسلل إلى شبكة إنترنت أشياء الرعاية الصحية مبني على الفئات الفرعية للبيانات في هذا العمل البحثي وهو عبارة عن نسخة معدلة من خوارزمية OSVM القياسية، والذي أطلق عليه اسم SDOSVM، والذي يأخذ في الاعتبار الفئات الفرعية في الفئة المستهدفة من البيانات، وهي فئة البيانات العادية، من أجل تقليل تشتت البيانات داخل وبين الفئات الفرعية وبالتالي تحسين القوة التمييزية وأداء تصنيف البيانات لنظام كشف التسلل.

بقدر ما هو معروف، فإن هذا العمل البحثي هو الأول من نوعه لاستكشاف استخدام خاصية الفئات الفرعية من البيانات في اكتشاف اختراق شبكة إنترنت أشياء الرعاية الصحية من خلال دمج معلومات الفئات الفرعية لبيانات إنترنت أشياء الرعاية الصحية المتعلقة بالسلوكيات العادية المتنوعة لتلك الشبكة، وتحسين دقة الكشف عن أي تسلل إليها.

تم تقييم نظام الكشف عن التسلل المبني على الفئات الفرعية المقترح باستخدام مجموعة بيانات TON_IoT لإنترنت الأشياء من العالم الحقيقي، ومقارنتها بأحدث خوارزميات تعلم الآلة ذات الفئة الواحدة للبيانات. أظهرت نتائج التجربة أن النهج المقترح تفوق في الأداء على خوارزميات تعلم الآلة ذات الفئة الواحدة للكشف عن اختراق الشبكة من ناحية مقياس AUC، حيث كانت نتائج AUC التي تم تحقيقها من خلال النهج المقترح أعلى من 96%. كذلك، تفوق النهج المقترح على الخوارزميات المستندة على SVM ذات الفئة الواحدة من ناحية مقياس الوقت المنقضي لتدريب النموذج، والذي لم يتجاوز الثانية الواحدة، نظرًا لاعتماده على عدد منخفض من SVs مما أدى إلى تعقيد حسابي أقل. وأخيرًا، حقق النهج المقترح أفضل النتائج من حيث مقياس FPR، والتي لم تتجاوز 1%، ومقياس ال TPR في معظم مجموعات التقييم الفرعية لمجموعة بيانات TON_IoT التي كانت أعلى من 98.5%.



(استمارة إيداع البحث)

اسم الباحث: مروه أحمد أنس محمود فوده

عنوان البحث: تأمين إنترنت أشياء الرعاية الصحية باستخدام نهج قائم على تعلم الآلة لكشف التسلل

سنة النشر: 2021م

الكلية: تقنية المعلومات البريد الإلكتروني: 20105915@stu.uob.edu.bh

القسم: نظم المعلومات رقم الاتصال: 36428821/36144480

إقرار الباحث بالموافقة على الشروط التالية:

يحق لجامعة البحرين إتاحة وتوفير الرسالة أو الأطروحة المذكورة أعلاه في مكتبة الجامعة أو من خلال موقع الحاوية العلمية للجامعة أو على نطاق أوسع في الخارج؛ كما ويمكن إعادة طباعة الرسالة إذا اقتضت الحاجة لذلك أو رقمتها مع إتاحتها على شبكة الأنترنت وذلك حسب تقدير إدارة الجامعة.

تنطبق هذه الشروط على جميع النسخ بما في ذلك النسخ الإلكترونية

تعتبر هذه الرسالة أو الأطروحة من المواد المحمية والتي تخضع إلى حقوق النشر والتأليف، ولا يجوز نشر أي اقتباس من الرسالة بدون توثيق المصدر.

التاريخ: 2021/10/01م

توقيع الباحث: 

جامعة البحرين
كلية تقنية المعلومات



تأمين إنترنت أشياء الرعاية الصحية باستخدام نهج قائم على تعلم الآلة لكشف التسلل

أطروحة مقدمة كجزء من متطلبات الحصول على
درجة الماجستير في الأمن الإلكتروني

اعداد

مروه أحمد أنس محمود فوده

20105915

اشراف

د. رياض بن محمد قسطيني

(أستاذ مشارك)

د. وائل محمد المدني

(أستاذ مشارك)

جامعة البحرين

مملكة البحرين

أكتوبر 2021